



CIMPLICITY 11

Tracker User Guide



Proprietary Notice

The information contained in this publication is believed to be accurate and reliable. However, General Electric Company assumes no responsibilities for any errors, omissions or inaccuracies. Information contained in the publication is subject to change without notice.

No part of this publication may be reproduced in any form, or stored in a database or retrieval system, or transmitted or distributed in any form by any means, electronic, mechanical photocopying, recording or otherwise, without the prior written permission of General Electric Company. Information contained herein is subject to change without notice.

© 2020, General Electric Company. All rights reserved.

Trademark Notices

GE, the GE Monogram, and Predix are either registered trademarks or trademarks of General Electric Company.

Microsoft® is a registered trademark of Microsoft Corporation, in the United States and/or other countries.

All other trademarks are the property of their respective owners.

We want to hear from you. If you have any comments, questions, or suggestions about our documentation, send them to the following email address:

doc@ge.com

Tracker

Chapter 1. About Proficy HMI/SCADA - CIMPLICITY Tracker.....	7
Chapter 2. Tracker Tutorial.....	8
About the CIMPLICITY Tracker Tutorial.....	8
Tracker Overview.....	10
Tracker Overview.....	10
Production Tracking (PRT).....	11
Routing and Control Objects (RCO).....	13
Tracker Structure.....	14
1. Set up for the Tracker Tutorial.....	16
1. Set up for the Tracker Tutorial.....	16
Exercise 1.1. Plan the Tracker Factory Layout.....	16
Answer 1.1. Check your Rough Sketch of the Factory Floor.....	17
1.2. Create a CIMPTRACK Project with a TRACKING Resource.....	17
1.3. Review the Workbench Components for Tracker.....	18
1.4. Open the Tracker Configuration Window.....	19
2. Configure Production Tracking (PRT).....	22
2. Configure Production Tracking (PRT).....	22
2.1. Configure PRT Regions and Groups.....	24
Exercise 2.1. Add Additional Regions to CIMPTRACK.....	29
2.2. Configure PRT Item Types and Classes.....	30
Exercise 2.2. Add Additional Item Types to CIMPTRACK.....	31
2.3. Configure PRT Routes.....	32
Exercise 2.3. Add Additional Routes to CIMPTRACK.....	36
2.4. Review Default Status Codes.....	37
2.5. Configure a PRT Graphic Display.....	38
2.6. Define Attributes for the Serialized Items.....	42
2.7. Enter System Definitions.....	43
2.8. Validate the CIMPTrack PRT Configuration.....	44
3. Use the Production Tracking User Interface (PRT_UI).....	47
3. Use the Production Tracking User Interface (PRT_UI).....	47
3.1 Add Blocks to the SCHEDULE Region.....	48
Exercise 3.1. Add Additional Blocks to the SCHEDULE Region.....	51

3.2. Add a Carrier to the LOAD_CONV Region.....	52
Exercise 3.2. Add Additional Carriers to the LOAD_CONV Region.....	53
3.3. Simulate the Assembly Production Line Manually.....	54
Exercise 3.3. Continue to Move Items throughout the Tracking Model.....	58
4. Export and Import Production Tracking Data.....	59
5. Create a Simple CimView Screen.....	61
5. Create a Simple CimView Screen.....	61
5.1. Configure a Quantity Text Object for the SCHEDULE Region.....	62
Exercise 5.1. Configure/Display the Quantity for all Regions.....	63
5.2. Configure a Toggle Button for the LOAD to ROBOT_CONV Route.....	64
Exercise 5.2. Create the Remaining Toggle Buttons.....	66
5.3. Configure a Button for the SCHEDULE to LOAD Route.....	66
Exercise 5.3 Create a Variable Setpoint Button for the UNLOAD to PALLET Route.....	68
6. Use BCE Extensions to Modify Tracking Information.....	69
6. Use Basic Control Engine Extensions to Modify Tracking Information.....	69
6.1. Create a Basic Control Engine Script for Blocks.....	70
Exercise 6.1. Create a Basic Control Engine Script for Carriers.....	73
Script 6.1. Check the Script to Add Carriers Automatically.....	73
Exercise 6.2. Create a Script to add Blocks Automatically.....	74
Script 6.2. Check the Script to add Blocks Automatically.....	74
6.3. Create a Script to add Attributes to Block Items.....	74
Exercise 6.3. Add Attributes for Carrier Items to the Script.....	78
Script 6.3. Check the Script for Adding Carrier Attributes.....	79
7. Use BCE Extensions to Control the ROBOT_CONV to MACHINE Process.....	79
8. Create a Routing Control Object to Modify and Route Items.....	82
8. Create a Routing Control Object to Modify and Route Items.....	82
8.1. Create a Screen to Update Points that will Hold your Attributes.....	83
Exercise 8.1. Configure Additional Buttons for the attrentry.cim Screen.....	88
8.2. Set up an Attribute Maintenance List for your RCO Configuration.....	89
8.3 Configure an AddParts Routing Control Site.....	90
8.4. Apply the ADDPARTS_DONE Point to the attrentry.cim Screen.....	97
8.5. Test the AddParts RCO.....	99

8.6. Add Another Function Block to the AddParts RCO.....	100
8.7. Implement the RCO Move Block/Carrier Configuration.....	102
9. Create a Split Routing RCO Site.....	104
9. Create a Split Routing RCO Site.....	104
9.1. Create a SPLIT RCO Site.....	105
9.2. Create a RCONV_MACH0 Decision for the SPLIT Site.....	106
Exercise 9.2. Create a RCONV_MACH1 Decision for the SPLIT Site.....	107
9.3. Create a Routing Logic Module.....	108
9.4. Revise the MACHINE Button Configuration on the CIMPTRACK Screen.....	108
9.5. Create an RLM to Move Blocks by Color.....	109
Exercise 9.5. Complete the Decision_by_Attribute RLM Configuration.....	110
Answer 9.5. Review the Decision_by_Attribute RLM Configuration.....	111
Exercise 9.6. Create a Block by Attribute RLM.....	112
Answer 9.6. Review the Block by Attribute Function Configuration.....	113
10. Use Logging with Tracker.....	114
10. Use Logging with Tracker.....	114
10.1 Configure a PRT_GRP Table.....	115
Exercise. 10.1. Create PRT_ITEM and PRT_REG Tables.....	116
10.2. Select the Data to Log.....	116
10.3. Use the Global Parameter Synchronization Capability.....	119
11. Configure Alarming for Tracker.....	120
Conclusion.....	121
Appendix.....	122
Script Keys.....	122
Tracker Training Script Key 1.....	122
Tracker Training Script Key 2.....	123
Tracker Training Script Key 3.....	125
Chapter 3. Tracker Guide.....	126
About Tracker (Base).....	126
Production Tracking: PRT.....	126
About Production Tracking.....	126

Production Tracking System Definitions and Guidelines.....	127
Production Tracking Model Layout Planning.....	156
Production Tracking Configuration.....	164
Production Tracking User Interface (PRT_UI).....	205
PRT Technical Reference.....	248
Routing Control Objects: RCO.....	324
About the Routing Control Objects Module.....	324
Routing Control Object Definitions.....	324
Routing Control Object Configuration.....	329
Routing Control Object User Interface (RCO_UI).....	352
Relocate RCO and TADB Databases.....	373
RCO Function Blocks.....	376
Tracker Data Logging.....	589
Tracker Data Logging.....	589
Step 1. Edit the Datalog.idt.....	591
Step 2. Edit the Data_field.idt.....	593
RCO Site Logging Summary Overview.....	599
Chapter 4. Tracker Appendix.....	603
About the Tracker Appendix.....	603
Production Tracking Application Interface.....	603
Production Tracking Application Interface.....	603
Application Subroutines.....	604
Process Resynchronization.....	689
PRT Macros.....	690
Event Codes.....	691
PRT Interface to External Software.....	695
PRT Objects for use in the Basic Control Engine.....	696
PRT Objects for use in the Basic Control Engine.....	696
PRT Object Model Member List.....	696

RCO External Decision.....	754
RCO External Decision API Functions.....	754
ExtDec_api_init.....	755
ExtDec_api_get_msg.....	755
ExtDec_api_init_comm.....	756
ExtDec_api_send_decision.....	757
ExtDec_api_term.....	757
RCOExtDec API Calls in relation to PDC_API Calls.....	758
External Decision Simulator.....	758
Simulator Configuration Requirements.....	761
RcoExtDecision.....	765
Chapter 5. Order Execution Management.....	772
Order Execution Management.....	772
About Order Execution Management.....	772
1. External Scheduling.....	773
2. XMLT.....	773
3. DIR_WATCHER.bcl.....	796
4. Product Order Management System.....	799
5. POMS CimView Order Entry.....	810
6. Tracker Attribute Database.....	817
7. Range Source Architecture.....	859
8. Tracker Query Engine.....	893
9. Order Execution Management Broadcast.....	934

Chapter 1. About Proficy HMI/SCADA - CIMPPLICITY Tracker

Overview

Tracker is a state-of-the-art tracking software for use with CIMPPLICITY that provides you with two option levels.

Level	Description
1 (page 126)	Tracker core option, the premier software for tracking and making decisions for building products, beginning with orders and following the process through the warehouse for the completed product.
2 (page 172)	Order Execution Management provides a comprehensive addition to Tracker that enables you to track, store, categorize and sequence your customers' orders based your configured criteria.

Tracker and the Workbench


If Tracker and Order Execution Management are enabled, the Workbench provides the following icons to access Tracker and Order Execution Management features.

rect 32, 230, 165, 247 [About Tracker \(Base \) \(page 126\)](#)

rect 31, 245, 177, 263 [About the Tracker Query Engine \(page 893\)](#)

rect 38, 397, 191, 415 [Production Tracking User Interface \(PRT UI\) \(page 205\)](#)

rect 39, 412, 193, 435 [Routing Control Object User Interface \(RCO UI\) \(page 352\)](#)

Icon	Opens the
Tracker Configuration (page 126)	Tracker Configuration User Interface.  Note: This interface provides the features for both Tracker and Order Execution Management configuration.
Tracker Query Engine (page 893)	(Query) Expression Browser.
PRT TADB Diff (page 848)	PRT vs. TADB Validation window.
Production Tracking UI (page 205)	Runtime Production Tracking User interface
Routing Control Objects UI (page 352)	Runtime Routing Control Objects User interface

Chapter 2. Tracker Tutorial


About the CIMPLICITY Tracker Tutorial

This tutorial provides you with a snapshot of the options and procedures available for tracking a production process. It will be most beneficial if you are acquainted with many CIMPLICITY features including point configuration, CimEdit configuration, Basic Control Engine scripting and alarming.

There will be links throughout this tutorial to more detailed documentation (that is not in the tutorial) about the topic you are reviewing or working on.

You will find that:

- Some topics in the tutorial provide you with detailed configuration descriptions, particularly when you might be unfamiliar with the configuration.
- Other topics are exercises that suggest that you do the configuration yourself.


 **Note:** When answers are not easily found in a previous topic, an answer topic will be available for you to check your work.

- Each exercise builds on what you learned in previous exercises. Therefore, it is important that you follow them in order.

Tracker Tutorial: Subject/Process Details

In this Tracker tutorial, you are in charge of assembling and shipping an electronic product. This includes configuring the details required to bring three blocks through a factory process.

1. There is an area where the carriers reside until they are needed.
2. The various parts of the electronic product are verified and placed onto a carrier.
3. The product/carrier is then moved to the automatic assembly area where, depending on the decision criteria, it is automatically routed to one of two assembly machines.
4. After automatic assembly (regardless of which machine was used), the product is moved to the pack and ship area.

 **Note:** There is only one area for pack and ship.

5. The assembled product is removed from the carrier.
6. Documentation is added; the product is boxed and placed onto a pallet
7. (After four boxes have been placed onto a pallet) the pallet is moved to the warehouse for storage.

Electronic Product Attributes are:

Attribute Name	Description
CUSTOMER ORDER #	Alphanumeric value
COLOR	Red, Green, or Blue
CPU SER #	Holds the CPU Serial Number
BASE PLATFORM	Checklist item
DRIVE HOUSING	Checklist item
SUB_ASSEMBLY A	Checklist item
SUB_ASSEMBLY B	Checklist item
ASSEMBLY HOUSING	Checklist item
POWER SUPPLY	Checklist item
PROD START TIME	Time/date of product start
AUTOCELL MACHINE	Machine Number
AUTOCELL TEMP	Temperature
AUTOCELL PRESS	Pressure
AUTOCELL TIME	Time/date of assembly
DVD	Checklist item
MANUALS	Checklist item
WARRANTY	Checklist item
PACK TIME	time/date when packed

Tracker Tutorial: Main Sections

The Tracker Tutorial will bring you through the basic procedures and exercises to set up the factory process.

You will find detailed descriptions about how to use the various procedures and then, you will be given exercises so you can make sure you understand what was completed. The exercises are integral with the factory process configuration.

Step	Description
1 (page 16)	Set up for the Tracker Tutorial
2 (page 22)	Configure Production Tracking (PRT)
3 (page 47)	Use the Production Tracking User Interface (PRT_UI)
4 (page 59)	Export and Import Production Tracking Data
5 (page 61)	Create a Simple CimView Screen
6 (page 69)	Use BCE Extensions to Modify Tracking Information
7 (page 79)	Use BCE Extensions to Control the ROBOT_CONV to MACHINE Process
8 (page 82)	Create a Routing Control Object to Modify and Route Items
9 (page 104)	Create a Split Routing RCO Site
10 (page 114)	Use Logging with Tracker
11 (page 120)	Configure Alarming for Tracker

Tracker Overview

Tracker Overview

Tracker is an option designed for production tracking and decision control.

[Tracker \(page 14\)](#) uses and interfaces with CIMPLICITY's robust features.

Tracker can be divided into two basic components:

PRT (page 11)	Production Tracking (PRT)
RCO (page 13)	Routing and Control Objects (RCO)
Tracker (page 14)	Structure

Production Tracking (PRT)

Production Tracking (PRT) is a [specialized database \(page 128\)](#) designed for the purpose of tracking products as they progress through a factory.

- PRT:Administration Interaction
- PRT: Uses
- PRT: Assignments
- PRT: Associations
- PRT: Decisions and Control Tools

PRT: Administration Interaction

Production Tracking has three points, which are:

- The PRT User Interface.
- Configuration files that you use to define your application
- Optional application subroutines that you can use to customize your application

PRT: Uses

You can use Production Tracking to:

- Receive information on items as they move through the production facility
- Request information on the contents of Regions through which Items are being tracked
- Add new Items to be tracked
- Delete or modify Items currently being tracked; Locate specific Items being tracked
- Place or remove external holds on tracked Items
- Move Items between Regions.

PRT: Assignments

- Each product in Production Tracking is assigned a [tracking item \(page 129\)](#) that represents its location and associated attributes (e.g. Color, Part number, Entry time) within the PRT database.
- Physical areas of the factory are assigned [Regions \(page 138\)](#) in the PRT database.

A product's physical location corresponds to its tracking item's location, defined within PRT by its PRT Region and PRT Region Location.

PRT: Associations

Products and areas are often [associated \(page 134\)](#) in some physical or logical fashion on the factory floor, a concept represented within PRT by a hierarchical grouping scheme.

	Feature	Description	Note
1	One or more items	Are single entities.	Items can be associated with one another to represent the progress of a group of products through the factory.
2	Each location	<ul style="list-style-type: none"> • Contains one or more items. • Is a physical space within a region. 	
3	Each Region	<ul style="list-style-type: none"> • Contains one or more locations. • Must be the member of at least one tracking region group. 	Example Regions can represent bins, conveyors and workstations on the factory floor that are in physical or logical proximity to one another.
4	A PRT Group	<ul style="list-style-type: none"> • Contains one or more regions. • All regions in the group must be assigned to the same PRT process service ID. 	Groups: <ul style="list-style-type: none"> • Organize regions by physical location or by function. • Allow a user to search for items, or groups of items, based on several criteria.
5	A PRT Service	Supports an independent list of one or more groups.	
6	PRT Services	Have one or more (multiple) copies of the PRT process running simultaneously.	Services are at the highest level of the hierarchy with a unique object name,

PRT: Decisions and Control Tools

Within the PRT database via several subsystems, items can be

- Added
- Deleted

- Moved or
- Modified

Subsystems include the following.

	Subsystem	Can do the following:
	Transition Points (CIMPLICITY points)	Move items from region to region based upon location or identification.
	Basic Control Engine (BCE)	Perform all functions available in PRT.
	Application Programming Interface (API)	Perform all functions available in PRT.
	Graphical User Interface (PRT_UI)	Perform all functions available in PRT.

Routing and Control Objects (RCO)

A Routing Control Object monitors one or more areas of the Production Tracking Model and makes routing decisions for that area based on logic modules or custom Basic Control Engine (BCE) logic and actions configured by the user.

The RCO system allows production routing decisions to be made automatically via logic configured in a graphical user interface.

- RCO: Components
- RCO: Process
- RCO: Optional Configurations

RCO: Components

RCO consists of two components.

1. RCO Configuration in the TrackerCfg_UI and
2. RCO Runtime Environment (RCOUI).

Multiple routing control objects can be configured to manage different areas of the manufacturing environment.

RCO: Process

3. The routing control object (RCO) monitors triggers that are either manually initiated via RCO_UI or predefined sequences of CIMPLICITY points.
4. When it finds that a trigger has been set, the RCO executes a routing logic module to determine what decisions, if any, to make based upon that new information.

5. RCO executes:
 - a. The decision-specific output logic module associated with the selected decision.
 - b. A common output module (if one has been configured) for the control site.
6. The output logic modules are used to communicate the RCO's decision to a PLC on the factory floor.

RCO: Optional Configurations

Optional configurations include:

7. The RCO can also be configured to wait for a ready point to hold a predefined value before executing any logic.
8. Once all logic has been executed, the site can also be configured to set a Done point.

The Done point informs the PLC that all signals for this decision have been sent, and to wait for a verification point to be set by the PLC acknowledging completion, before returning to trigger detection.

Tracker Structure

- Tracker and Application Processes
- Production Tracking and Other CIMPLICITY Features
- CimView Screens

Tracker and Application Processes

The Tracker module performs using a series of configurable resident processes:

Process	Description
prtc.exe	Production Tracking Data Collector
prts.exe	Production Tracking Data Server
prtv.exe	Production Tracking Configuration Validation Program

Additional resident processes can be configured for Routing and Control Objects (RCO). Production Tracking accepts:

- Unsolicited messages from application processes that request Item and Tracking Region data.
- Messages that request Item tracking data and Tracking Region data modifications.

Application processes:

- Can send messages to Production Tracking to initiate production start for serialized and non-serialized Items.
- Will communicate with Production Tracking through a limited set of Production Tracking application functions used to send the requests.

Production Tracking and Other CIMPLICITY Features

PRT: And Point Management
PRT: And Alarm Management
PRT: And Users

PRT : And Point Management

Production Tracking interfaces with CIMPLICITY software's Point Management module in order to receive CIMPLICITY software point data values.

Production Tracking is capable of receiving point data from multiple Point Management processes.

The interface with Point Management also

- Transfers information on Items entering and exiting Tracking Regions,
- Provides Point Management with information on Tracking Region status,
- The quantity of Items in a Tracking Region and
- The quantity of Items by type in a Tracking Region.

Configuration data specifies the information Production Tracking will provide to Point Management.

PRT: And Alarm Management

Production Tracking interfaces with Alarm Management in order to notify Alarm Management of alarm conditions.

Production Tracking can be configured to generate alarms when exception conditions are encountered as Items are tracked.

Users are notified of alarms based on their assigned role and their view of configured Resources.

PRT: And Users

Production Tracking interfaces with users through a graphic user interface.

Graphic display of Region tracking data and Item attribute data is provided through CimEdit/CimView.

The User Interface provides users with a view of all regions and the data contained within those regions.

CimView Screens

Operator user interaction occurs in customized CIMPLICITY CimView screens.

1. Set up for the Tracker Tutorial

1. Set up for the Tracker Tutorial

Exercise 1.1 (page 16)	Plan the Tracker Factory Layout
Answer 1.1 (page 17)	Plan the Tracker layout.
1.2 (page 17)	Create a CIMPTRACK Project with a TRACKING Resource
1.3 (page 18)	Review the Workbench Components for Tracker
1.4 (page 19)	Open the Tracker Configuration Window

Exercise 1.1. Plan the Tracker Factory Layout


[Planning \(page 156\)](#) is the key to smoothly developing and executing your Tracker project.

Think about putting together a jigsaw puzzle. Is it easier to put together with or without a picture to use as a reference? Apply your answer to Tracker. For this tutorial create a rough representation of the factory floor as your planning step.

1. Review the information on the Tracker Tutorial Scenario and Process Details pages. Pay close attention to:
 - Components in the process,
 - Areas that are described and
 - The direction and type of movement through the system.
2. Plan the production process and translate this into the architectural terms of Tracker.

Assume that there are:

Simple LED beams set up between all the regions that will toggle points to tell when an item has passed from one region to the other.

 **Note:** When you create your own Tracker project, try to anticipate and map out many other details that you will become aware of as you go through this tutorial.

Answer 1.1. Check your Rough Sketch of the Factory Floor

Does your sketch look something like this?

The Tutorial Subject/Process Details provided the information for this layout.

Note: You will be referring to this sketch and expanding on it throughout the tutorial.

1.2. Create a CIMPTRACK Project with a TRACKING Resource

You use the same procedure to create the CIMPTRACK project as any other CIMPLICITY project.

- CIMPTRACK will require the following
- Options to be enabled: Database Logger (both A&E & App and Points), System Sentry, and Tracker.
- A TRACKING resource, which is created using the same procedure as any other resource.
- The TRACKING resource must have ADMINISTRATOR identified as a user for the project.

A (page 17)	Create a CIMPTRACK Project
B (page 18)	Create a TRACKING Resource

1. Create a CIMPTRACK Project
 1. Open the CIMPLICITY Workbench.
 2. Select File>New>Project on the Workbench menu bar.

The Create As dialog box opens.

3. Do the following.

A	Find the directory in which you want to create the project.
B	Name the project CIMPTRACK for this tutorial.
C	Make sure that the following are checked in the Options box: Database Logger (both A&E & App and Points), System Sentry, and Tracker.
D	Click Create.

The Project Properties dialog box displays.

4. Check Enable project broadcast.

Result: The Workbench will display and will be ready for you to begin Tracker configuration.

a. Create a TRACKING Resource

You will have to select a CIMPLICITY resource when you create some Tracker components.

Create a TRACKING resource for this exercise.

5. Do the following.

A	Select Project>Security> Resources in the Workbench left-pane.
B	Click the New button on the Workbench toolbar.

A New Resource dialog box opens.

6. Do the following.

A	Enter TRACKING in the Resource ID field
B	Click OK.

A Resource Definition dialog box opens.

7. Do the following.

A	Enter TRACKER Resource in the Description field.
B	Add ADMINISTRATOR to the Users for this resource box.
C	Click OK.

The TRACKING resource is added to the Workbench right-pane resource list.

1.3. Review the Workbench Components for Tracker

- Tracker Icons in the Workbench
- Application Windows that Apply to Tracker

Tracker Icons in the Workbench

There are several icons in the Workbench that apply directly to the basic Tracker application.

The icons' locations are as follows.

1.


A	Expand the Project folder.	The Tracker Configuration icon displays You will do Tracker configuration first.
B	Expand the Runtime folder> Production Tracking folder.	The following icons display. <ul style="list-style-type: none"> • Production Tracking UI • Routing Control Objects UI These are runtime applications that you will work in later.
C	Note the Screens icon under the Project folder.	CimEdit/CimView screens, which you will work in to create the custom User Interface, open through the Screens icon.

Application Windows that Apply to Tracker

Tracker includes both configuration and runtime windows. CimEdit/CimView also plays an integral role in Tracker configuration and runtime operation.

Major components are as follows.

1	Tracker Configuration User Interface (TrackerCfg_UI)	Configuration
2	Production Tracking User Interface (PRT_UI)	Runtime
3	CimEdit/CimView	Configuration/Runtime
4	Routing Control Objects User Interface (RCO_UI)	Runtime

 **Note:** The first window you will work in will be the TrackerCfg_UI.

1.4. Open the Tracker Configuration Window

 **Important:** Before you open the Tracker Configuration window, create a SQL database named CIMPTRACK

A (page 20)	Open the Tracker Configuration Window
B (page 20)	Configure SQL Server for Tracker

1. Open the Tracker Configuration Window

The first window you will work with is the Tracker Configuration Window (TrackerCfg_UI).

Double-click the **Tracker Configuration** icon in the Workbench right pane.

The first time you open the TrackerCfg_Ui you will have to configure and/or select the Tracker database.

1. Configure SQL Server for Tracker

The first time you open the TrackerCfg_UI an Options dialog box displays.

You can use it to create a SQL Server database connection for Tracker.

The SQL server connection is needed to store the Tracker configuration data.

If you know how, go ahead and configure the connection.

Name the connection CIMPTRACK.

Use this section to check or follow a procedure for your configuration.

1 (page 20)	Open the ODBC Data Source Administrator
2 (page 21)	Create a CIMPTRACK Data Source
3 (page 22)	Apply the CIMPTRACK Data Source to Tracker

1. Open the ODBC Data Source Administrator

The first time you open the TrackerCfg_UI, an Options dialog box displays through which you can configure the connection to the Tracker database.

1. Enter sa in the **Login Id** field.

Note: Leave the **Password** field blank.

1. Click ODBC Admin.

Result: The ODBC Data Source Administrator dialog box opens.

1. Create a CIMPTRACK Data Source

1. Select the System DSN tab in the ODBC Data Source Administrator dialog box.
2. Click Add.

1. Do the following.

1	Choose SQL Native Client <nn>.
2	Click Finish.

1. Do the following in the Create a New Data Source to SQL Server dialog box.

1	Enter the following	
	Name	CIMPTRACK
	Description	Tracker Tutorial
	Server	(local) or a SQL database that has been assigned to this project, e.g. computer name\CIMPLICITY.
2	Click Next.	

1. Do the following for the Login:

1	Check With SQL Server Authentication using a login ID and password entered by the user.
2	Enter the Login ID and Password required by your SQL database.
3	Click Next.

1. Do the following.

1	Change the default database to CIMPTRACK.
2	Click Next to move through the following dialog boxes until the ODBC Microsoft SQL Server Setup dialog box displays.

1. Test the data source to verify the connection to the database.

1. Click OK.

The ODBC Administrator dialog box displays the new CIMPTRACK connection.

1. Click OK to close the ODBC Data Source Administrator dialog box.

The ODBC Data Administrator closes; the RCO Database Connection dialog box displays.

1. Apply the CIMPTRACK Data Source to Tracker

Do the following.

A	Enter the password you entered on the ODBC Login screen in the Password field
B	Click OK.

Result: The ODBC Data Administrator closes and the TrackerCfg_UI displays two top-level folders.

- CIMPTRACK
- PRT

You will work in both during this tutorial.

2. *Configure Production Tracking (PRT)*

2. *Configure Production Tracking (PRT)*

[Production Tracking \(PRT\) \(page 11\)](#) is a specialized database designed for the purpose of tracking products as they progress through a factory.

You set up the definitions for the PRT database in the TrackerCfg_UI.

Following is the recommended order for configuring components in your PRT database.

You already have made a rough sketch of areas that will become regions, groups and routes.

In this section, you will define them in detail. In addition, you will create and describe the item types will go through the factory process.

2.1 (page 24)	Configure PRT Regions and Groups.
Exercise 2.1 (page 29)	Add Additional Regions to CIMPTRACK
2.2 (page 30)	Configure PRT Item Types and Classes
Exercise 2.2 (page 31)	Add Additional Item Types to CIMPTRACK
2.3 (page 32)	Configure PRT Routes.
Exercise 2.3 (page 36)	Add Additional Routes to CIMPTRACK
2.4 (page 37)	Review Default Status Codes
2.5 (page 38)	Configure a PRT Graphic Display
2.6 (page 42)	Define Attributes for the Serialized Items
2.7 (page 43)	Enter Service Definitions.
2.8 (page 44)	Validate the CIMPTRACK PRT Configuration.

 **Note:**

- This section provides some exercises that you will have to do on your own. However, it mainly provides detailed configuration descriptions, so you can have hands on practise configuring the many details involved with the PRT setup.

- Tracker provides a [Wizard \(page 168\)](#) that takes you through the configuration steps. However, this tutorial does not use the Wizard, which is described in other Tracker documentation.

2.1. Configure PRT Regions and Groups

2.1. Configure PRT Regions and Groups

The PRT Region is configured in the PRT Region dialog box, which is opened in the Tracker Configuration window>PRT>Advanced tree.

- PRT Region Configuration Dialog Box: Open
- PRT Region Configuration Dialog Box: Sections/Tabs

PRT Region Configuration Dialog Box: Open

Open the PRT Region Configuration dialog box as follows.

1. Expand the **PRT** tree in the Tracker window left pane.
2. Do the following.

A	Double-click Regions .
B	Click in the TrackerCfg_UI right pane.

The PRT Region Configuration dialog box opens in a basic view.

2.1.1. Enter SCHEDULE Region Basic Specifications

If you [check your factory floor sketch \(page 17\)](#), you will see that the schedule items area is the area in which the process begins.

You will define basic specifications for the SCHEDULE region first.

A (page 24)	Region Basic Specifications
B (page 25)	Group (ASSEMBLY) Configuration

1. Region Basic Specifications

[Basic specifications \(page 169\)](#) for the SCHEDULE region as follows:

Enter the following.

	Field	Value
1	ID	SCHEDULE
2	Description	Production Schedule
3	Locations	500
4	Items per Location	1
5	Resource	TRACKING

 **Note:**

- If you did not create the TRACKING resource before opening the Tracker Configuration window, you can

1. Right-click the Popup Menu button to the right of the ' field.
2. Select New on the Popup menu.

A New Resource dialog box opens.

1. Follow the same procedure you would have followed if you opened the New Resource dialog box through the Workbench.

- You will fill in the **Group** field next.

1. Group (ASSEMBLY) Configuration

The SCHEDULE region will be in an ASSEMBLY group, which needs to be created

1. Click to the right of the **Group** field in the PRT Region Configuration dialog box.
2. Select New in the popup menu.

The [PRT Group Configuration \(page 191\)](#) dialog box opens.

3. Enter the following.

A	Group	ASSEMBLY
B	Label	ASSEMBLY_1
C	Description	Electronic Assembly
D	Resource	TRACKING

4. Click OK.

The PRT Group Configuration dialog box closes; the ASSEMBLY group is ready to be selected in the PRT Region Configuration dialog box>Group field.

 **Note:**

- You can also create groups directly through the TrackerCfg_UI tree.
- If you double-click Groups after you complete configuration for this group, you will see that ASSEMBLY displays in the group list.

2.1.2. Enter SCHEDULE Region Advanced Specifications

Click on the PRT Region Configuration dialog box>Region tab if the **Advanced** section is not displaying.

Most of the **Advanced** fields were automatically filled in as soon as you entered SCHEDULE in the **ID** field.

Enter the [one value \(page 26\)](#) that was not entered automatically and leave the automatically entered values.

	Field	Value
A	Enter the following.	
	Label	SCHED
B	The following fields are automatically filled in with the appropriate values.	
	Service	PRT_DC
	Tracking Type	Sequential
	Region Type	Normal
	Status Point	SCHEDULE.TK_STAT
	Auto Move Point	SCHEDULE.TK_AUTMOV
	Item Qty Point	SCHEDULE.TK_ITMQTY

2.1.3. Enter SCHEDULE Region OOS Specifications

Click if the OOS section is not displaying..

In the [Out of Seq \(page 170\)](#) section, you can specify where out of sequence (OOS) items should go.

A (page 27)	Detainment Region
B (page 28)	Remaining OOS Configuration

1. Detainment Region

1. Click the Popup menu button to the right of the **Detainment Region** field in the SCHEDULE region's PRT Region Configuration dialog box.
2. Select New on the Popup menu.

A new PRT Region Configuration dialog box opens; the **Group** and **Resource** fields are automatically filled in, as follows.

Group	ASSEMBLY
Resource	TRACKING

3. Click Advanced.
4. Fields that you fill in and that are automatically filled in are as follows.

Field		Value
Enter the following values.		
1	ID	DETAINMENT
2	Description	Electronic Assembly Detainment Area
3	Locations	400
4	Items per Location	4
5	Label	DETAIN
6	Tracking Type	Detainment. Note: The default is Sequential and needs to be changed.
7	The following values are automatically entered. They are correct for the DETAINMENT region.	
	Service	PRT_DC
	Region Type	Normal
	Status point	DETAINMENT.TK_STAT
	Auto Move point	DETAINMENT.TK_AUMOV

Field	Value
Item Qty Point	DETAINMENT.TK_ITMQTY

5. Click OK.

Result

- a. The PRT Configuration dialog box for the DETAINMENT region closes.
- b. The PRT Region Configuration dialog box for the SCHEDULE region displays again.
- c. DETAINMENT Is entered in the Detainment Region field.

Note: Alarming and Logging will not be configured for the DETAINMENT region.

- a. Remaining OOS Configuration

Fill in the other two fields in the Out of Seq section of the SCHEDULE region PRT Region Configuration dialog box as follows.

1	OOS cycle	25
2	OOS Action	Delay Detain

2.1.4. Review the SCHEDULE Region Configuration

The PRT Region Configuration dialog box includes a large amount of data.

Therefore, it is important that you make sure your SCHEDULE region's PRT Region Configuration dialog box>Region tab looks like this and you understand what you entered before you continue.

2.1.5. Enter SCHEDULE Region Alarming/Logging Specifications

The region should have its alarming and logging features enabled.

- Region Alarming
- Region Logging

Region Alarming

1. Select the Alarming tab in the PRT Region Configuration dialog box.
2. Do the following

Result: All of the [check boxes \(page 171\)](#) on the tab are selected.

A	Check Enable Alarms.
B	Check Select ALL.

Region Logging

3. Select the Logging tab in the PRT Region Configuration dialog box.
4. Do the following.

All of the [check boxes \(page 172\)](#) on the tab are selected.

A	Check Master Logging Enable.
B	Check Select ALL.

2.1.6. Finish SCHEDULE Region Configuration

1. Click OK in the PRT Region Configuration dialog box.
2. Click the Save button on the TrackerCfg_UI menu bar.

A message displays asking you if you want Tracker to auto create missing region points.

1. Click Yes.

Result: Tracker creates the points; two new regions display in the Tracker window.

Result: You have just created and configured the SCHEDULE region.

Exercise 2.1. Add Additional Regions to CIMPTRACK

Your tracking model needs to have regions for each area in your [factory floor sketch \(page 17\)](#).

1. Create Additional Regions.
2. Add more regions according to the following chart. Remember to enable Alarming and Logging as you did for the schedule region.
3. Leave the following entries, which will automatically display until they are changed.

Field	Entry
Group	ASSEMBLY
Resource	TRACKING

Note: Refer to the steps for configuration of the Schedule Region if needed.

If you refer to the [rough sketch \(page 17\)](#) of the factory floor, you will see that these regions have been identified for the manufacturing process.


When you finish this procedure, they will be entered in the PRT database.

Region ID	Locations	Items per Location	Tracking Type	Region Type	OOS
SCHEDULE	500	1	Sequential	Normal	25
LOAD	1	2	Sequential	Combine	0
ROBOT_CONV	50	2	Sequential	Normal	1
LOAD_CONV	300	1	Sequential	Normal	25
MACHINE0	50	2	Sequential	Normal	25
MACHINE1	50	2	Sequential	Normal	25
PACK_CONV	50	2	Sequential	Normal	25
UNLOAD	1	2	Sequential	Disperse	1
PALLET	1	4	Sequential	Normal	1
WAREHOUSE	250	4	Pool	Normal	NA
DETAINMENT	400	4	Detainment	Normal	NA

Already configured

1. Review the Completed Region List

1. Check to see that all your required regions appear in the configuration window when you are finished.
2. Click the Save button on the TrackerCfg_UI toolbar to save the configuration.
3. Click Yes when the message box asks you if Tracker should auto-create the missing region points.

 **Important:** Make sure your configuration is saved when you have completed the exercise.

2.2. Configure PRT Item Types and Classes

The PRT Item Type is configured in the PRT Item Type dialog box, which is opened in the Tracker Configuration window>PRT>Advanced tree.

A (page 31)	Open the PRT Item Type Configuration Dialog Box
-----------------------------	---

B <i>(page 31)</i>	Enter Item Type 00 Specifications
C <i>(page 31)</i>	Finish 00 Item Type Configuration

1. Open the PRT Item Type Configuration Dialog Box
1. Expand the **PRT** tree in the Tracker window left pane.
2. Do the following.

A	Double-click Item Types .
B	Click in the TrackerCfg_UI right pane.

A PRT Item Type Configuration dialog box opens.

1. Enter Item Type 00 Specifications

Item 00 is the first item type for the carrier that is used to move the electronic blocks. Specifications for the 00 item type demonstrate how to enter values in this dialog box.

Fill in or check the following in the [PRT Item Type Configuration \(page 176\)](#) dialog box..

1	Item Type ID	00
2	Description	Assembly Carrier
3	Item Type Code	1
4	Item Class	CARRIER
5	Tracking Type	Serialized

1. Finish 00 Item Type Configuration

Click OK in the PRT Item Type Configuration dialog box.

The new 00 item type is listed in the Tracker Configuration window>Item Types list.

Exercise 2.2. Add Additional Item Types to CIMPTRACK

Add more item types according to the following chart.

Note: Refer to the steps for configuration of the item types, if needed.

A (page 32)	Create additional Item Types
B (page 32)	Review the Completed Item Type List

1. Create additional Item Types


1. Create all the item types on the following table.
2. Enter the related specifications that are listed on the table for each item type.

Item: Type ID	Description	Item Type Code	Item Class	Tracking Type
00	Assembly Carrier	1	CARRIER	Serialized
25	Assembly Model 25	2	BLOCK	Serialized
36	Assembly Model 36	3	BLOCK	Serialized
60	Assembly Model 60	4	BLOCK	Serialized
99	Assembly Model 99	5	BLOCK	Serialized

1. Review the Completed Item Type List

1. Check to see that all your required item types are listed in the configuration window when you are finished.
2. Click the Save button on the Tracker Configuration window toolbar when the list is complete.
3. Double-click **Item Types**.

Result: The saved Item Type list displays in the Tracker Configuration window right-pane.

 **Important:** Make sure your configuration is saved when you have completed the exercise

2.3. Configure PRT Routes

Items follow routes from one region to another as they move through the Tracker model.

Each route from one region to the next needs to be configured.

A (page 33)	Open the PRT Route Configuration Dialog Box
B (page 33)	Enter Route Source/Destination Regions

C (page 34)	Enter Route Transition Type/Point ID
D (page 35)	Enable Process first point change
E (page 36)	Finish SCHEDULE to LOAD Route Configuration

1. Open the PRT Route Configuration Dialog Box

1. Expand the **PRT** tree in the Tracker window left pane.
2. Do the following.

A	Double-click Routes .
B	Click in the TrackerCfg_UI right pane.

The PRT Route Configuration dialog box opens in a basic view.

1. Enter Route Source/Destination Regions

Select the following basic route specifications.

rect 0, 51, 21, 75 [\(page 33\)](#)

rect 0, 77, 21, 101 [\(page 34\)](#)

rect 142, 104, 221, 126 [\(page 34\)](#)

1 (page 33)	Source Region
2 (page 34)	Destination Region

1	Source Region
---	---------------

1. Click the Browser button to the right of the **Source Region** field.

A PRT Browse window opens.

1. Do the following.

1	Select SCHEDULE for the Source region
---	---------------------------------------

2	Click OK.
---	-----------

Result: SCHEDULE is now the Source Region.

2	Destination Region
---	--------------------

1. Click the Browser button to the right of the Destination region field.

The PRT Browse window opens again.

1. Select LOAD for the **Destination Region**.
2. Click OK.

Result: LOAD is now the Destination Region.

1. Enter Route Transition Type/Point ID

Transition points can move items from region to region based upon location or identification.

Each route in your factory model will need a transition type and point.

Transition type and point for the SCHEDULE to LOAD route are as follows.


rect -1, 109, 22, 129 [\(page 34\)](#)

rect -1, 131, 22, 151 [\(page 34\)](#)

1 (page 34)	Transition Type
2 (page 34)	Transition Point ID

1	Transition Type
---	-----------------

Select Automatic Item ID giving Item ID in the **Transition Type** dropdown menu.

 **Note:** Limit Switch displays as the Transition type as the default value. That is not correct for this route.

2	Transition Point ID
---	---------------------

Create the transition point, X_LOAD1, as follows.

1. Click the Popup Menu button to the right of the **Transition Point ID** field.
2. Select New on the Popup menu.

A New Point dialog box opens.

1. Enter and select the following.

1	Point ID	X_LOAD1
2	Type	Virtual
3	Class	Text

1. Click OK.

The X_LOAD1 Point Properties dialog box opens.

1. Select the General tab.
2. Enter and select the following.

The Point Properties dialog box opens.

1. Enter and select the following on the Virtual tab.

1	Description	Positive ID for Schedule to Load
2	Data type	STRING_20
3	Elements	1

1. Click OK.


3	Enable
---	--------

The PRT Route Configuration dialog box displays again with the basic criteria for the SCHEDULE to LOAD route completed.

1. Enable Process first point change

Do the following.

1	Advanced button	Click Advanced.
2	Process first point change.	Check.

 **Note:** No other PRT Route Configuration dialog **Advanced** section will be configured for this tutorial.

1. Finish SCHEDULE to LOAD Route Configuration

Click OK in the PRT Route Configuration dialog box.

The new route displays in the Tracker window.

Exercise 2.3. Add Additional Routes to CIMPTRACK

You saw that a route goes from one region to another.

You need to create a route for the movement from each region to region that your items will make.

Consult the [rough sketch \(page 17\)](#) you created at the beginning of the tutorial to figure out what routes you have to configure.

Make a list of Source to Destination regions.

A (page 36)	Create additional Routes
B (page 37)	Review the Completed Route List

1. Create Additional Routs

Did you figure out most of the Source to Destination routes?

1. Create all the routes on the following table.
2. Enter the related specifications that are listed on the table for each route.

Note: Refer to the steps for configuring the [SCHEDULE to LOAD \(page 32\)](#) route, if needed.

Several of the point types are BOOL; one is STRING 20. The reason will become clear as you progress through the tutorial.


Source Region	Destination Region	Transition Type	Transition Point ID	Data Type
SCHEDULE	LOAD	Automatic Item ID Giving Item ID	X_LOAD1	STRING_20
LOAD	ROBOT_CONV	LIMIT SWITCH	X_ROBOT_CONV	BOOL

Source Region	Destination Region	Transition Type	Transition Point ID	Data Type
ROBOT_CONV	MACHINE0	LIMIT SWITCH	X_MACHINE0	BOOL
LOAD_CONV	LOAD	LIMIT SWITCH	X_LOAD2	BOOL
MACHINE0	PACK_CONV	LIMIT SWITCH	X_PACK_CONV0	BOOL
MACHINE1	PACK_CONV	LIMIT SWITCH	X_PACK_CONV1	BOOL
PACK_CONV	UNLOAD	LIMIT SWITCH	X_UNLOAD	BOOL
UNLOAD	LOAD_CONV	LIMIT SWITCH	X_LOAD_CONV	BOOL
UNLOAD	PALLET	Automatic Item ID Giving Item ID	X_PALLET	STRING_20
PALLET	WAREHOUSE	LIMIT SWITCH	X_WAREHOUSE	BOOL
ROBOT_CONV	MACHINE1	LIMIT SWITCH	X_MACHINE1	BOOL

Already configured

- a. Review the Completed Route List

Check to see that all your required routes appear in the configuration window when you are finished.

 **Important:** Make sure your configuration is saved when you have completed the exercise.

2.4. Review Default Status Codes

You can specify the various status conditions that exist for the CIMPTRACK regions and items (e.g. hold, normal, waiting) by editing Status Configurations.

However, since CIMPLICITY comes with default status conditions for items, regions and internal processes, it is not necessary to edit these.

Select PRT>Advanced>**Status Codes** in the TrackerCfg_UI left-pane to see the default status codes.

The four default Status conditions/Bit Set for items are:

	Status Code	Label
1	DLAYD	delayed
2	EHOLD	external hold
3	HOLD	internal hold

	Status Code	Label
4	INORML	normal
The seven default Region Status conditions are:		
	Status Code	Label
1	RCAPEXC	capacity exc
2	REHOLD	external hold
3	RFULL	region full
4	RINLOCK	in lock
5	RNORML	normal
6	ROUTLOCK	out lock
7	ROUTSEQ	out of seq

2.5. Configure a PRT Graphic Display

2.5. Configure a PRT Graphic Display

The PRT graphic display configuration enables you to perform various counts on region item data or to display item contents into array points.

- PRT Graphic Display Configuration Dialog Box: Open
- PRT Graphic Display Configuration Dialog Box: Tabs

PRT Graphic Display Configuration Dialog Box: Open

1. Expand the **PRT** tree in the Tracker window left pane.
2. Do the following.

A	Double-click Graphics Display .
B	Click in the TrackerCfg_UI right pane.

A PRT Graphics Display Configuration dialog box opens.

PRT Graphic Display Configuration Dialog Box: Tabs

The PRT Graphic Display Configuration dialog box has two tabs. Each requires some configuration.

rect 175, 373, 205, 397 [2.5.3. Finish PRT Graphic Display Configuration \(page 42\)](#)

rect 55, 46, 85, 70 [2.5.2. Enter Graphic Display Global Specifications \(page 40\)](#)

rect 20, 46, 50, 70 [2.5.1. Enter PRT Graphic Display General Specifications \(page 39\)](#)

Demo	
2.5.1 (page 24)	Enter PRT Graphic Display General Specifications
2.5.2 (page 26)	Enter Graphic Display Global Specifications
2.5.3 (page 26)	Finish PRT Graphic Display Configuration

2.5.1. Enter PRT Graphic Display General Specifications

Select the General tab in the PRT Graphic Display Configuration dialog box.

Important:

- The **Display Point** must be created/selected first in order have access to the appropriate values in the other fields.
- The fields on the PRT Graphic Display Configuration dialog box>General tab above the **Display Point** field must be configured in order to access the Global tab.

A (page 39)	Display Point
B (page 40)	Display Point Enabled Top Fields

1. Display Point

1. Click the Popup Menu button to the right of the **Display Point** field.
2. Select New on the Popup menu.

A New Point dialog box opens.

3. Enter and select the following.

A	Point ID	RED_COUNT
B	Type	Virtual
C	Class	Analog

4. Click OK.

The RED_COUNT Point Properties dialog box opens.

5. Leave the defaults on the General tab, including:

Data Type	Int
Elements	1

6. Select the Virtual tab.

7. Enter and select the following.

A	Initialization	Initialized
B	Initial value	0
C	Calculation	None

8. Click OK.

Result: RED_COUNT is entered in the Display Point field.

1. Display Point Enabled Top Fields

Select the following options.

A	Region	SCHEDULE
B	Attribute	PRT_ITEM_STATUS
C	Operator	Equals (=)
D	Value	1

2.5.2. Enter Graphic Display Global Specifications

Select the Global tab in the PRT Graphic Display Configuration dialog box.

Entries on this tab are **Reset Point** and **Refresh Rate**.

A reset point forces the counters to do a global update when the reset point is updated.

rect -1, 95, 26, 119 ([page 41](#))

rect -1, 69, 26, 93 ([page 41](#))

A (page 41)	Reset Point
--	-------------

B (page 41)	Refresh Rate
-----------------------------	--------------

1. Reset Point

1. Click the Popup Menu button to the right of the **Reset Point** field.
2. Select New on the Popup menu.

A New Point dialog box opens.

1. Enter and select the following.

1	Point ID	PRTRESET
2	Type	Virtual
3	Class	Boolean

1. Click OK.

The PRTRESET Point Properties dialog box opens.

1. Leave the defaults on the General tab, including:

Data Type	BOOL
Elements	1

1. Select the Virtual tab.
2. Make sure the following are selected.


1	Initialization	None
3	Calculation	None

1. Click OK.

Result: PRTRESET is entered in the Reset Point field.

1. Refresh Rate

Enter a number higher than 500 in the **Refresh Rate** field.

 **Note:** 500 is the default.

You cannot leave the Global tab if a number lower than 500 is entered in the field. A message will alert you to increase the number to 500 or higher.

2.5.3. Finish PRT Graphic Display Configuration

1. Click **OK** in the PRT Graphic Display Configuration dialog box.


The configured PRT display for the SCHEDULE region is listed in the TrackerCfg_UI.

1. Click the Save button on the TrackerCfg_UI toolbar.

Result: The graphic display configuration is saved.

2.6. Define Attributes for the Serialized Items

You can define the attributes that the serialized items may have in the [prt_attributes.cfg \(page 256\)](#) file.

 **Note:** It is possible to add attributes dynamically. However, adding attributes to prt_attributes.cfg avoids the trouble of continuously having to add the attribute for each item that is created. You can also write a BCE script that handles this, but when you edit prt_attributes.cfg the attributes are always there.

1. Click Tools>Command Prompt on the Workbench menu bar.

The Command window opens in the CIMPTRACK directory..

2. Type `cd master` to change to the CIMPTRACK master directory.
3. Press Enter.
4. Type `notepad prt_attributes.cfg`.

The prt_attributes.cfg file opens.

 **Important:** As you enter items.


- Separate items by the pipe symbol |.
- Assign attributes only for the serialized items.

In CIMPTRACK the two serialized items are:

5. Carriers
6. Blocks
7. Enter the following attributes.

Following is a list of the attributes.

```
CPU SER # BASE PLATFORM DRIVE HOUSING SUB_ASSEMBLY A SUB_ASSEMBLY B ASSEMBLY
HOUSING POWER SUPPLY PROD START TIME AUTOCELL MACHINE AUTOCELL TEMP AUTOCELL
PRESS AUTOCELL TIME DVD MANUALS WARRANTY PACK TIME
```

 **Note:** When you enter the attributes into the prt_attributes.cfg file, begin the list with ITEMCLASS|BLOCK so the attributes will be assigned to the blocks only; not to the carriers.

Following is the list in the entry format.

ITEMCLASS|BLOCK

```
CPU SER #|BASE PLATFORM|DRIVE HOUSING|SUB_ASSEMBLY A|SUB_ASSEMBLY
B|ASSEMBLY HOUSING|POWER SUPPLY|PROD START TIME|AUTOCELL
MACHINE|AUTOCELL TEMP|AUTOCELL PRESS|AUTOCELL TIME|DVD|MANUALS|
WARRANTY|PACK TIME
```

8. Save prt_attributes.cfg.
9. Close Notepad.
10. Exit the Command Prompt window.

The attributes are now in the CIMPTRACK project. You do not have to do anything else to this file for CIMPLICITY to use it.

2.7. Enter System Definitions

A (page 44)	Open the System Definition Configuration Dialog Box
B (page 44)	Enter System Definition Values
C (page 44)	Finish System Definitions Configuration

1. Open the System Definition Configuration Dialog Box

1. Expand the **PRT** tree in the Tracker window left pane.
2. Do the following.

A	Double-click System Definitions .
B	Right-click the default entries in the TrackerCfg_UI right-pane.
C	Select Edit on the Popup menu.

Result: The selected System Definition Configuration dialog box opens.

1. Enter System Definition Values

Enter the following [system definition \(page 180\)](#) values for the CIMTRACK project.

1	Interface request	10
2	API synchronous	10
3	API interested process	100
4	Data Collector main	20
5	Data Server main	20
6	Data Server throttle	20
7	DS interested process port	50
8	Data Collector aux port	10

1. Finish System Definitions Configuration

1. Click OK in the System Definition Configuration dialog box.

The dialog box closes. The system definitions are changed in the TrackerCfg_UI.

2. Click the Save button on the TrackerCfg_UI toolbar.

The revised system definitions are saved.

2.8. Validate the CIMPTrack PRT Configuration

2.8. Validate the CIMTRACK PRT Configuration

Before you move to the next section make sure that your PRT Configuration is valid and error free. You can do this by using two validation tools.

2.8.1 (page 45)	Prepare the PRT Configuration for Validation
2.8.2 (page 46)	Use Validation Tools to Verify the PRT Configuration

2.8.1. Prepare the PRT Configuration for Validation

Simply do the following to prepare your configuration for validation.

A (page 45)	Save the PRT Configuration
B (page 45)	Start the CIMPTRACK Project

1. Save the PRT Configuration

1. Click the Save button on the TrackerCfg_UI toolbar if it is enabled.

If there are still missing points, a message asks you to auto create them.

2. Click Yes.

Result: The missing points are created.

a. Start the CIMPTRACK Project

3. Display the CIMPTRACK - CIMPPLICITY Workbench.

4. Click the Configuration Update button on the Workbench toolbar to perform a configuration update.

A message displays to confirm the configuration update.

5. Click OK.

6. Click the Run button on the Workbench toolbar.

The CIMPTRACK project starts running and is ready for the PRT validation.

2.8.2. Use Validation Tools to Verify the PRT Configuration

You can use either or both of the available validation tools to make sure that your PRT Configuration is free of errors.

- Tracker PRTV Utility
- CIMPLICITY Status Log

Tracker PRTV Utility

Now you can validate your PRT configuration.

1. Do the following.

A	Make sure that CIMTRACK is running.
B	<ol style="list-style-type: none"> a. Click the Dynamic Configuration button. A Logon dialog box will open. a. Enter ADMINISTRATOR; Click OK.
C	Select Tools>Command Prompt on the Workbench menu bar.

The Command window opens in the CIMTRACK directory.

2. Click the Dynamic Configuration button.

A Logon dialog box will open.

3. Enter ADMINISTRATOR; Click OK.

4. Do the following.

	Action	Entry	Description
A	Type	<code>prtv trkerror.txt</code>	<ul style="list-style-type: none"> • <code>prtv</code> runs the PRT Validation utility, which writes any errors to a file named <code>trkerror.txt</code>. • You can enter any name for the error text file.
B	Press	Enter.	<ul style="list-style-type: none"> • The PRTV Configuration Utility checks each of the PRT files and alerts you if the Tracker configuration has errors as follows. Configuration verification report generated in file <code>trkerror.txt</code>. CIMTRACK has errors!
C	Type	<code>notepad trkerror.txt.</code>	Notepad, a text editor, will open <code>trkerror.txt</code>

Result: The result depends on whether or not the PRTV Validation tool found errors.

Errors were found	A <code>trkerror.txt</code> file opens in Notepad listing the errors.
-------------------	---

No errors were found	<ul style="list-style-type: none"> • A line in the Command Prompt window will report that No errors were detected in Production Tracking configuration. • trkerror.txt is located in the project root and can be opened at any time.
----------------------	--

5. Exit the Command window.

CIMPLICITY Status Log

The CIMPLICITY Status Log errors include any errors reported when you used the PRTV utility.

6. Click the Status Log button on the CIMPLICITY Workbench to open the Status Log.

7. Double-click an error line if there are errors.

A Detail dialog box will display the information in an easy-to-read format.

3. Use the Production Tracking User Interface (PRT_UI)

3. Use the Production Tracking User Interface (PRT_UI)

A (page 47)	PRT_UI: Open
B (page 48)	PRTUI: Procedures

PRT_UI: Open

The [Production Tracking User Interface \(page 208\)](#) (PRT_UI) enables you to:

- Send items to and from different regions,
- Change the status of a region or item,
- Locate a particular item,
- Delete an item,
- And more.

This section guides you through performing many of these functions.

Do the following in the CIMPLICITY Workbench to open the PRT_UI.

1	Make sure: <ul style="list-style-type: none"> • CIMPTRACK project is running. • Dynamic configuration is enabled.
2	Double-click Runtime>Production Tracking> Production Tracking UI in the Workbench left or right-pane.

Result: The PRT_UI window opens.

PRTUI Procedures

3.1 (page 48)	Add Blocks to the SCHEDULE region
Exercise 3.1 (page 51)	Add Additional Blocks to the SCHEDULE Region
3.2 (page 52)	Add Carriers to the LOAD_CONV region
Exercise 3.2 (page 53)	Add Additional Carriers to the LOAD_CONV Region
3.3 (page 54)	Simulate the assembly production line manually
Exercise 3.3 (page 58)	Continue to Move Items throughout the Tracking Model

3.1 Add Blocks to the SCHEDULE Region

3.1. Add Blocks to the SCHEDULE Region

If you check your factory layout you will see that the SCHEDULE region is the first region that items going through the factory will enter.

As an introduction to how PRT_UI works, you will first add blocks manually to this region.

3.1.1 (page 49)	Add a Block to the SCHEDULE region.
--	-------------------------------------

3.1.2 (page 50)	Check the Block's Attributes
Exercise 3.1 (page 51)	Add Four Additional Blocks to the SCHEDULE Region

3.1.1. Add a Block to the SCHEDULE Region

A (page 49)	Open an Add Item Dialog Box for the SCHEDULE Region
B (page 49)	Enter General Information about the Block
C (page 50)	Set the Block Item Status
D (page 50)	Finish ABLK234 Configuration

1. Open an Add Item Dialog Box for the SCHEDULE Region

Do the following in the PRT_UI.

1	Select SCHEDULE in the Region Id dropdown menu.
2	Click the Add Item button on the PRT_UI toolbar.

An Add item dialog box opens.

1. Enter General Information about the Block

1. Select the [General Information \(page 214\)](#) tab in the Add Item dialog box.

2. Enter/select the following.

	Field	Value	Note
A	Item ID	ABLK234	This ID has to be unique because serialized items require a unique item ID.
B	Item Class ID	BLOCK	Think of all of the electronic parts making up your product as a single Block product.
C	Item Type	36	

Note: You do not need to enter anything else on the General Information tab.

a. Set the Block Item Status

3. Select the Item Status tab in the Add item dialog box.

4. Do the following.

A	Select normal in the Status column.
B	Click Set Active. The normal status is now active.

a. Finish ABLK234 Configuration

Click OK in the Add Item dialog box.

Result: ABLK234 is listed in the PRT_UI SCHEDULE region.

3.1.2. Check the Block's Attributes

The ABLK234 block should be assigned attributes that you entered in the [prt_attributes.cfg file \(page 42\)](#).

A (page 50)	Open the ABLK234 Block's Modify Item Dialog Box
B (page 50)	Check the General Information Tab
C (page 51)	Check the Attributes for ABLK234
D (page 51)	Close the Modify Item Dialog Box

1. Open the ABLK234 Block's Modify Item Dialog Box

Do the following in the PRT_UI.

1	Select the ABLK234 line.
2	Click the Modify Item button. Note: you can also double-click the ABLK234 line.

The Modify Item Dialog box opens.

1. Check the General Information Tab

Values that you entered when you created the item display on the General Information tab in the Modify Item dialog box.

Note: A Reference ID, which was entered automatically, is not manually used in this tutorial.

1. Check the Attributes for ABLK234

Select the Attributes tab in the Modify Items dialog box.

The attributes you entered in the [prt_attributes.cfg file \(page 42\)](#) are listed in the **Attributes** box. These attributes apply to BLOCK items.

	Feature	Description	Note
A	Attribute Type	Standard Attributes are selected.	
B	Attributes	Attributes you entered in the TrackerCfg_UI are listed	Attributes are stored in the project's backing files.

a. Close the Modify Item Dialog Box

Click OK to close the Modify Item dialog box.

 **Note:** You can also edit the block item's [status \(page 50\)](#) in the Modify Item dialog box.

Exercise 3.1. Add Additional Blocks to the SCHEDULE Region

A (page 51)	Add the Additional Blocks
B (page 52)	Review the Blocks in the PRT_UI Interface


1. Add the Four Additional Blocks

Add four more Block items to the SCHEDULE region.

Item ID	Item Type
BBLK234	99
CBLK234	36
DBLK234	99

Item ID	Item Type
EBLK234	60

1. Review the Blocks in the PRT_UI Interface

 **Note:** Remember that the SCHEDULE region has 500 locations and 1 item per location.

When you add an item to the region, PRT adds it after the selected location.

If SCHEDULE was configured to allow more than 1 item per location, when you add an item to the region, PRT would add it:

- To the selected location if it is not full.
- After the selected location if it is full.

3.2. Add a Carrier to the LOAD_CONV Region

A (page 52)	Open an Add Item Dialog Box for the LOAD_CONV Region
B (page 52)	Enter General Information about the CARRIER
C (page 53)	List ABLK234 in the PRT_UI SCHEDULE Region

1. Open an Add Item Dialog Box for the LOAD_CONV Region

Do the following.

1	Select LOAD_CONV in the Region Id dropdown menu.
2	Click the Add Item button on the PRT_UI toolbar.

An Add Item dialog box opens.

1. Enter General Information about the CARRIER

1. Select the [General Information \(page 214\)](#) tab in the Add Item dialog box.

2. Enter/select the following.

	Field	Value	Note
A	Item ID	ACAR567	
B	Item Class ID	CARRIER	You will need a carrier to carry the block through the manufacturing process.
C	Item Type	00	

Note: You do not need to enter anything else on the General Information tab.

- a. Finish ACAR567 Configuration

Click OK in the Add Item dialog box.

Result: ACAR567 is listed in the PRT_UI_LOAD_CONV region.

- a. Check the Modify Item Dialog Box>Attributes Tab
3. Open the Carrier's Modify Item dialog box.
 4. Select the Attributes tab.

No attributes are listed because you had limited the attribute assignment to blocks when you entered the list in the [prt_attributes.cfg \(page 42\)](#) file.

Exercise 3.2. Add Additional Carriers to the LOAD_CONV Region

A (page 53)	Add Additional Carriers
B (page 53)	Review the Carriers in the PRT_UI Interface

1. Add the Additional Carriers

Add four more Carriers to the LOAD_CONV region.

If you check the [rough sketch \(page 17\)](#) of the factory floor you will see that these carriers pick up a block in the LOAD region, carry it through the process, unload it and circle back to pick up another block.

When you go further through the tutorial, you will need as many carriers as there are blocks that are in the process of being manufactured.

1. Review the Carriers in the PRT_UI Interface

Remember that Carriers are also serialized, so you have to specify a unique item ID for them, as well.

3.3. Simulate the Assembly Production Line Manually

3.3. Simulate the Assembly Production Line Manually

To move an item, you have to be in the region where the item is located.

To fetch an item, you have to be in the region where you want the item to be.

3.3.1 (page 54)	Move a Block from the SCHEDULE to LOAD region.
3.3.2 (page 55)	Move a Carrier into the LOAD region.
3.3.3 (page 56)	Fetch a Block into the ROBOT_CONV region.
Exercise 3.3 (page 58)	Continue to move regions throughout the Tracking Model.

3.3.1. Move a Block from the SCHEDULE to LOAD Region

A (page 54)	Open the Destination Dialog Box
B (page 55)	Select a Destination for CBLK234
C (page 55)	Confirm that the Block Moved

1. Open the Destination Dialog Box

Do the following.

1	Select the SCHEDULE region.
2	Select the CBLK234 block in Location 3 to move.
3	Click the Move Item button on the PRT_UI toolbar.

Result: The Destination dialog box opens.

1. Select a Destination for CBLK234

1. Select the LOAD region in the **Region ID** dropdown list.

Notes The LOAD region is where you will load the Block onto the Carrier.

 **Note:**

- The LOAD region is where you will load the Block onto the Carrier.
- Insert is checked by default.

2. Click OK.

The SCHEDULE region displays in the PRT_UI.

1. Confirm that the Block Moved

The available blocks should have changed in the SCHEDULE and LOAD regions.

- SCHEDULE Region
- LOAD Region

SCHEDULE Region


The SCHEDULE region now should have four blocks.

Blocks that were in Locations 4 and 5 should now be in Locations 3 and 4.

LOAD Region

CBLK234 is in the LOAD region's Location 1.

3.3.2. Move a Carrier into the LOAD Region

 **Tip:** Follow the same procedure you used to move a block from the SCHEDULE region into the LOAD region.


Clear the Insert check box in the Destination dialog box.

A checked Insert will insert a new empty location at the specified location and put the new item there.

A (page 56)	Move a CARRIER to the LOAD Region
B (page 56)	Confirm that the Carrier Moved

1. Move a CARRIER to the LOAD Region.
2. Select the Carrier CCAR567 to move to the LOAD region.
3. Click the Move Item button on the PRT_UI toolbar.

A Destination dialog box opens.

 **Note:** Clear the Insert check box in the Destination dialog box.

A checked Insert will insert a new empty location at the specified location and put the new item there.

4. Click OK.

Result: The LOAD_CONV region displays.

5. Confirm that the Carrier Moved.

The available carriers should have changed in the LOAD_CONV and LOAD regions.

The LOAD_CONV region now should have **four** blocks, not five.

Blocks that were in Locations 4 and 5 should now be in Locations 3 and 4.

Both the block and the carrier are now in the LOAD region.

- CBLK234 is in the LOAD region's Location 1.
- CCAR567 is in the LOAD region's Location 1.

The LOAD region was configured to allow [2 items per location \(page 29\)](#).

Do you remember why you need to clear the Insert check box? Review what happens when you insert or add an item to a region. Check the PRT Region Configuration dialog box; how many locations are in the LOAD region; how many items per location.

3.3.3. Fetch a Block into the ROBOT_CONV Region

You can fetch an item into a region from a region as well as move it from a region into a region.

A (page 57)	Open a Fetch Item Dialog Box
B (page 57)	Fetch CBLK234
C (page 58)	Make sure CBLK234 was Fetched

1. Open a Fetch Item Dialog Box

Note: Make a note of the Item ID of a block in the LOAD region. For this exercise you will use CBLK234.

Do the following.

1	Select ROBOT_CONV in the Region ID field.
2	Select File>Fetch on the PRT_UI window menu bar. Tip: You can also press Ctrl+T on the keyboard.

The Fetch Item dialog box opens.

1. Fetch CBLK234

1. Do the following.

A	Enter CBLK234 in the Item ID field.
B	Press the Tab key.

Result: PRTUI automatically enters the information about the item.

For this exercise the information is:

Field	Value Entered	
Reference ID	Automatically generated. This value will be different on different systems.	
Item Type	36	
Item Status	I0	
From		
	Region ID	LOAD
	Location	1
To		
	Region ID	ROBOT_CONV

Field	Value Entered	
	Location	-1

 **Note:**

- A message will report if the item is not found.
- Leave Fetch to next location clear.

2. Click OK.

The ROBOT_CONV region displays.

1. Confirm that the CBLK234 was Fetched

The available Block/carrier should have changed in the ROBOT_CONV and LOAD regions.

- LOAD_CONV Region
- LOAD Region

ROBOT_CONV Region

Both the block and the carrier are now in the ROBOT_CONV region.

- CBLK234 is in the ROBOT_CONV region's Location 1.
- CCAR567 is in the ROBOT_CONV region's Location 1.

Note: The ROBOT_CONV region was configured to allow 2 items per location.

LOAD Region

The LOAD region is now empty.

Exercise 3.3. Continue to Move Items throughout the Tracking Model

A (page 58)	Move items into the Warehouse
B (page 59)	Delete Items from the Tracking Model

1. Move items into the Warehouse

Now that you can add, move and fetch items

1. Continue to move the items through the tracking model and into the warehouse to get hands on experience with how it works.

Remember that you can only have **one** location in the **Combine Items** and **Disperse Items** regions (LOAD and UNLOAD).

2. Make sure you move out the current items before you add new ones in.
3. Try moving Blocks from the **ROBOT_CONV** region to either the **MACHINE0** or **MACHINE1** regions out of sequence.

What happens to the skipped over Carrier/Blocks?

 **Tip: Tips**

- Did you map out routes, number of locations in the region, number of items in the location?

If you did, moving Blocks and Carriers through the system should be very straight-forward.

- Remember to clear the Insert checkbox in the Destination dialog box when you move a second item into a location.
 - a. Delete Items from the Tracking Model

When all of the items are in the warehouse,

4. Select the items.
5. Click the Delete button on the PRT_UI toolbar.

4. Export and Import Production Tracking Data

What if you have to stop the production line for repairs, but you add some blocks to the schedule region while the repairs are going on?

You can tell Production Tracking that you have made changes while Tracker was down.

- Export File (.prtx)
- Import File (.prtm)

Export File (.prtx)

The .prtx export file exports your current Tracking configuration to a text file. You can then edit that text file and modify the sections you need to update.

The Tracker export utility will use this file to know what to export.

1. Do the following.

A	Make sure that CIMPTRACK is running.
B	<ul style="list-style-type: none"> a. Click the Dynamic Configuration button. A Logon dialog box may open. a. Enter ADMINISTRATOR; Click OK.
C	Select Tools>Command Prompt on the Workbench menu bar.

The Command window opens in the CIMPTRACK directory.

2. Click the Dynamic Configuration button.
A Logon dialog box may open.
3. Enter ADMINISTRATOR; Click OK.
4. Change to the CIMPTRACK Master directory.
5. Type the following command.

```
notepad prtexp.prtx
```

Where

```
notepad
```

is the text editor

```
prtexp.prtx
```

is the name of the file you will create.

Note: You can give the file any name. However, you must give it a .prtx extension.

Notepad opens: a message asks you if you want to create the file prtexp.prtx.

6. Click Yes.
7. Enter the [following lines in this file. \(page 296\)](#)

```
| -*
```

```
*
```

```
ALL
```

8. Save your changes and exit Notepad.

Result: You have just created a file that the export utility will use to export all of your production tracking data to a file.

Import File (.prtm)

The .prtm import file imports the data back into your runtime production tracking process when you bring it back online.

 **Important:** Make sure the project is running.

9. Type the following command at the CIMPTRACK Master directory command prompt.

```
prt_export prtexp.prtx ALL LOCAL
```

10. Press Enter.

The PRT export utility dumps the ASSEMBLY group, displaying the number of items per region in PRT_UI.

Note: Since you just deleted all of the items in PRT_UI each region has 0 items.

If a region displays 1 or more items, go to that region in PRT_UI and delete it/them.

A new file, `prtexp.prtm`, is created.

11. Take some time to examine this file and determine what the lines are saying.

Review detailed explanations about the fields in the file ([prtexp.prtm \(page 297\)](#)).

Note: `PRT_DC` = PRT Data Collector Resident Process (configured Service in the Tracker window).

5. Create a Simple CimView Screen

5. Create a Simple CimView Screen

The PRT_UI window has an important role in working with Tracker.

However, it is difficult to get an overall picture of the tracking process using the PRT_UI window.

A CimView screen

- Can show you all of the regions, along with corresponding information.
- Allows for business logic defined access to PRT data; PRT_UI does not.
- Basic CIMPTRACK Shape on a CimEdit Screen: Create
- Simple CimView Screen: Procedures

Basic CIMPTRACK Shape on a CimEdit Screen: Create

1. Start CimEdit.
2. Create some rectangle blocks and lines to represent the rough tracking model that you created when you started the tutorial.

Your screen layout should look similar to this:

Simple CimView Screen: Procedures

5.1 <i>(page 62)</i>	Configure a Quantity Text Object for the SCHEDULE Region
Exercise 5.1 <i>(page 63)</i>	Configure/Display the Quantity for all Regions
5.2 <i>(page 64)</i>	Configure a Toggle Button for the LOAD to ROBOT_CONV Route
Exercise 5.2 <i>(page 66)</i>	Create the Remaining Toggle Buttons
5.3 <i>(page 68)</i>	Configure a Button for the SCHEDULE to LOAD Route
Exercise 5.3 <i>(page 68)</i>	Create a Variable Setpoint Button for the UNLOAD to PALLET Route

5.1. Configure a Quantity Text Object for the SCHEDULE Region

A text point in each box that will show you the quantity of items that are currently in each region. Check your configuration or follow the procedures.

i Tip: Check each region's PRT Region Configuration dialog box in the Tracker Configuration window for the information you need. These points were automatically created when you created the regions. All the item quantity points have the suffix `_ITMQTY`.

1. Select Drawing>Text>Text String on the CimEdit Ribbon bar.
2. Place the new Text string near the **SCHEDULE** region.
3. Open the text string's Properties - Object dialog box.
4. Select the Text tab.
5. Select and enter the following options.

A	Multiline check box	Clear the check box.
B	String field	Enter the number of ###'s to accommodate the maximum number of characters required for the region's quantity.. number of locations*quantity in a location Example The SCHEDULE region can have 500 locations*1 item per location. The maximum number of items in the region, therefore, is 500. Enter ### to accommodate 3 characters..
C	Horiz. alignment	Check the Center radio button.
D	Expression	<ol style="list-style-type: none"> a. Click the Popup Menu button. b. Select Browse Point ID The Select a Point browser opens. <ol style="list-style-type: none"> a. Select SCHEDULE.TK_ITMQTY. SCHEDULE.TK_ITMQTY is the point that was automatically entered in the SCHEDULE's TrackerCfg_UI>PRT Region Configuration dialog box>Item Qty Point field

6. Click the Popup Menu button.
7. Select Browse Point ID
The Select a Point browser opens.
8. Select SCHEDULE.TK_ITMQTY.
SCHEDULE.TK_ITMQTY is the point that was automatically entered in the SCHEDULE's **TrackerCfg_UI>PRT Region Configuration dialog box>Item Qty Point** field
9. Click OK.
10. Place the text object in the SCHEDULE region box.

Exercise 5.1. Configure/Display the Quantity for all Regions

A (page 64)	Create Text Objects for all Regions
B (page 64)	Test the Region Quantities on the CimView Screen

1. Create Text Objects for all Regions

Repeat creating text objects for all of the regions.

When you are done your screen should look similar to this.

1. Test the Region Quantities on the CimView Screen

Click the Home>Runtime>Test Screen button on the CimEdit Ribbon bar.

A test CimView screen opens.

The item quantities displaying on the regions should match the quantities in the PRT_UI regions.

If you:

- Deleted all of the items, all of the numbers on the CimView screen should be 0.
- Retained the items from the previous exercises or added more the numbers should reflect those quantities.

1.

5.2. Configure a Toggle Button for the LOAD to ROBOT_CONV Route

For most of the regions, you can simply create buttons that update the associated transition point. This can either be a toggle action, or an absolute setpoint action, or whatever action causes the point to be updated. Begin by creating a toggle button to transition items from the LOAD to ROBOT_CONV region.

A (page 65)	Select a Button for the LOAD to ROBOT_CONV Route
B (page 65)	Create a Mouse Up Event
C (page 65)	Create a Toggle Setpoint Action

D (page 66)	Close the Dialog Boxes
--------------------------------	------------------------

1. Select a Button for the LOAD to ROBOT_CONV Route

1. Select Drawing>Text>Text Button on the CimEdit Ribbon bar.
2. Place a text button on the LOAD to ROBOT_CONV route on the CimEdit screen.
3. Open the button's Properties - Object dialog box.
4. Select the **Text Button** tab.
5. Do the following.

A	Multiline check box	Clear
B	String field	Delete the default Button text.
C	Button Color (Optional)	Change the button color so it will contrast with the screen and other object colors.

a. Create a Mouse Up Event

6. Select the Events tab in the button's Properties dialog box.

7. Do the following.

A	Event field	Select Mouse Up in the drop down list.
B	Action	<ol style="list-style-type: none"> a. Click the Popup button to the right of the Action field. b. Select New Procedure on the Popup menu.

A Procedure Information dialog box opens.

1. Create a Toggle Setpoint Action

Enter the following in the Procedure Information dialog box.

1	Procedure name field	Enter RobotConv.
2	New button	Click New.
3	Action type field	Select Toggle setpoint in the drop down list.
4	Point ID field	Select or enter X_ROBOT_CONV.

You created the **Transition Point ID**'s, which began with X_, when you configured the routes in the Tracker Configuration window.

Note: You can also select Absolute setpoint in the Action type field. If you do, enter 0 or 1 in the **Value** field.

1. Close the Dialog Boxes

Click OK to close the open dialog boxes, including the button's Properties dialog box.

Items can now be moved from the LOAD to ROBOT_CONV regions by toggling the route's button.

Exercise 5.2. Create the Remaining Toggle Buttons

You know that a limit switch triggers the transition on the route between regions. Routes that were configured with a limit switch require toggle buttons.

You can review how you created the transition points by going through the routes' [PRT Route Configuration \(page 32\)](#) dialog boxes.

1. Configure buttons for the other routes, except:

- SCHEDULE to LOAD
- UNLOAD to PALLET

When you are finished your CimEdit screen should look similar to this.

1. Add some items to regions in the PRT_UI.
2. See what happens to the quantities on your CIMPTRACK CimView screen.
3. Click buttons on the routes for the regions with items.

5.3. Configure a Button for the SCHEDULE to LOAD Route

The routes are different when you move blocks from the:

- SCHEDULE region to the LOAD region
- UNLOAD region to the PALLET region.

 **Note:** You selected an Automatic Item ID Giving Item ID Transition Type instead of a Limit Switch when you created these routes.

For these routes, you will have to prompt the user to tell you the item ID of the block, so Tracker knows which block to move. This can be done by creating a variable setpoint action that will allow a user to type in the item ID of the block.

Begin by configuring a button for the SCHEDULE region to the LOAD region.

A (page 67)	Select a Button for the SCHEDULE to LOAD Route
B (page 67)	Create a Mouse Up Event
C (page 67)	Create a Variable Setpoint Action
D (page 68)	Close the Dialog Boxes
E (page 68)	Test the SCHEDULE to LOAD Route Button

1. Select a Button for the SCHEDULE to LOAD Route

1. Place a text button in the SCHEDULE to LOAD route.

2. Open the button's Properties - Object dialog box.

3. Do the following.

A	Multiline check box	Clear
B	String field	Delete the default Button text.
C	Button Color (Optional)	Change the button color so it will contrast with the other route buttons.

a. Create a Mouse Up Event

4. Select the Events tab in the button's Properties dialog box.

5. Do the following.

A	Event field	Select Mouse Up in the drop down list.
B	Action	a. Click the Popup button to the right of the Action field. b. Select New Procedure on the Popup menu.

Result: A Procedure Information dialog box opens.

a. Create a Variable Setpoint Action

Enter the following in the Procedure Information dialog box.

1	Procedure name	XLOAD1
2	New	Click New.
3	Action type	Variable setpoint
4	Point ID	X_LOAD1.

SCHEDULE to LOAD route is one of the two routes that has a transition text point.

a. Close the Dialog Boxes

Click OK to close the open dialog boxes, including the button's Properties dialog box.

Result: Items can now be moved from the SCHEDULE to LOAD regions after you:

a. Test the SCHEDULE to LOAD Route Button

6. Make a note of an item ID that you want to move in the PRT_UI window>SCHEDULE region.


7. Open the CIMPTRACK CimView test screen.

8. Click the new SCHEDULE to LOAD route's button.

A Setpoint dialog box opens.

9. Enter the item ID to be moved in the **New value** field.

Example

 **Note:** In the real world, there would be some sort of bar code reader that would automatically read in the serial/item number.)

Exercise 5.3 Create a Variable Setpoint Button for the UNLOAD to PALLET Route

1. Repeat the Variable Setpoint procedure for the **UNLOAD to PALLET** route.

Your CimEdit screen should now look similar to this.

1. Add some items to CIMPTRACK in the PRT_UI interface and experiment with the CimView screen.

When you move a carrier and a block into LOAD


1. Check the number that displays on the CIMPTRACK CimView screen.
2. Check the PRT_UI to see what items are really in the region.

Do you remember why a 1, 2, 3, etc. might display in CimView as the quantity in the combine ROBOT_CONV region when [2, 4, 6 \(page 58\)](#), etc. items are really in the region?

6. Use BCE Extensions to Modify Tracking Information

6. Use Basic Control Engine Extensions to Modify Tracking Information

While you could sit at your computer and move things around all day, the whole point of production tracking is to let the computer do most of the work; to decide where items should go and to track them when they go there.

 **Note:** After initially being shown how to create part of the script you will be asked to continue on your own. Since the scripting is basic for CIMPLICITY, it is recommended that you attempt to write the sections described in the exercises first and then check what you wrote in the script topics.

6.1 (page 70)	Create a Basic Control Engine Script for Blocks
Exercise 6.1 (page 73)	Create a Basic Control Engine Script for Carriers
Script 6.1 (page 73)	Check the Script to Add Carriers Automatically
Exercise 6.2 (page 74)	Create a Script to add Blocks Automatically
Script 6.2 (page 74)	Check the Script to add Blocks Automatically
6.3 (page 74)	Create a Script to add Attributes to Block Items

Exercise 6.3 (page 78)	Add Attributes for Carrier Items to the Script
Script 6.3 (page 79)	Check the Script for Adding Carrier Attributes

6.1. Create a Basic Control Engine Script for Blocks

6.1. Create a Basic Control Engine Script for Blocks

This script will prompt the user for 10 block item ID's and corresponding model numbers.

You will then write code to add the items to the SCHEDULE region.

6.1.1 (page 70)	Create an Add Blocks Button for the Script
6.1.2 (page 72)	Test the Block Script in the CIMPTRACK CimView Window

6.1.1. Create an Add Blocks Button for the Script

A (page 70)	Create an Add Blocks Button
B (page 71)	Create a Mouse Up Event
C (page 71)	Write a Script for the Add Blocks Button
D (page 72)	Finish the Add Blocks Configuration

1. Create an Add Blocks Button

1. Place a button on the CIMPTRACK screen.
2. Name the button Add Blocks.

Tip: Replace the default Button entry in the **Text Button>String** field.

3. (Optional) Select a different color for the button face.
 - a. Create a Mouse Up Event
4. Select the Events tab in the button's Properties dialog box.
5. Do the following.

A	Event field	Select Mouse Up in the drop down list.
B	Action	<ol style="list-style-type: none"> a. Click the Popup button to the right of the Action field. b. Select New Script on the Popup menu.

Result: An Edit Script window opens.

- a. Write a Script for the Add Blocks Button

The script includes the following sections

6. Declare a variable that's going to hold the item ID that the user enters.

Enter:

```
Dim itemid As String Dim model As String
```

7. Create the input box that will prompt the user for an item ID and model type for the block.

Enter:

```
itemid = InputBox$("Enter BLOCK item ID, please", "Item ID Entry", "")
model = InputBox$("Enter BLOCK model type, please (valid entries: 25,36,60,99)", "Model Type Entry", "")
```

8. Take the value of the `itemid` string variable and create a new block item with the `itemid` string variable as the new block item's ID.

Declare a new object of type `prtitem`. Put this statement with your other variable declarations:

Enter:

```
Dim item As New prtItem
```

9. Add the code to add this new item to your schedule region.

Enter:

```
'Set the item properties item.regionid="SCHEDULE" item.itemid=itemid
item.itemtypeid=model item.regionloc=-1 item.exthold 0 item.groupid="MAIN" item.inthold 0
'Add the item to PRT item.Add
```

a. Finish the Add Blocks Configuration

10. Compile the script.
11. Click File>Close on the Edit Script window menu bar to close the window.
12. Click OK to close the Properties dialog box.

The CimEdit screen now has an Add Blocks button.

6.1.2. Test the Block Script in the CIMPTRACK CimView Window

Use the new Add Blocks button in CimView to add a block in the PRT_UI SCHEDULE region.

A <i>(page 72)</i>	Test the CimView Screen
B <i>(page 72)</i>	Check the PRT_UI SCHEDULE Region

1. Test the CimView Screen
 1. Make sure your project is running.
 2. Click the Add Blocks button.

An Item ID Entry dialog box should open.

3. Do the following.

A	Enter an Item Id in the text field (e.g. B38521).
B	Click OK.

A Model Type Entry dialog box:

- Should open.
- Reports the choices, which were entered in the script, that you have for valid entries.

4. Do the following.

A	Enter a Model Type in the text field (e.g. 36).
B	Click OK.

a. Check the PRT_UI SCHEDULE Region

The block that you entered through CimView should be listed in the PRT_UI SCHEDULE region.

Create a loop that will create 5 new blocks and as for the Item ID and Item Type before each block is created.

Hint: Review the For...Next statement.

Exercise 6.1. Create a Basic Control Engine Script for Carriers

1. Create another button named Add Carriers.
2. Create a script that automatically adds 10 carriers to the **LOAD_CONV** region.
3. Open the CIMPTRACK CimView window and add some carriers.

The carriers will be listed in the PRT_UI LOAD_CONV region.

Script 6.1. Check the Script to Add Carriers Automatically

The following script will add 10 carriers to the LOAD_CONV region.

```
Sub OnMouseUp(x As Long, y As Long, flags As Long)
```

```
Dim itemid As String Dim model as String Dim item As New prtItem
```

```
For i = 1 To 10
```

```
'Prompt user for the Carrier item id and model 'itemid = InputBox$("Enter CARRIER item id, please", "Item ID Entry", "") 'model = "00"
```

```
'Random generate item id Randomize Number = Random(1,5000) itemid = "C" & Number model = "00"
```

```
'Set the item properties item.regionid="LOAD_CONV" item.itemid = itemid
```

```
item.itemtypeid = model item.regionloc = -1 item.exthold 0 item.groupid = "MAIN" item.inthold 0 'Add the item to PRT item.Add
```


```
Next i
```

```
End Sub
```

Exercise 6.2. Create a Script to add Blocks Automatically

Instead of prompting the user for the item ID's, it would be a lot easier on if you just had your scripts add unique items without even bothering to ask for user input.

Change the script to add 10 blocks automatically. Even assign model types for the blocks.

 **Note:** Review the If...Then...Else and Randomize statements in the CIMPLICITY documentation.

Script 6.2. Check the Script to add Blocks Automatically

The following script will add 10 blocks automatically.

Since this is basic for CIMPLICITY, it is recommended that you attempt to write it first and then check what you wrote.

```
Sub OnMouseUp(x As Long, y As Long, flags As Long)
```

```
Dim itemid As String Dim model As String Dim item As New prtItem Randomize For i =
1 To 10 'Prompt user for item ID and model 'itemid=InputBox$("Enter BLOCK item ID,
please","Item ID Entry","") 'model=InputBox$("Enter BLOCK model type, please (valid entries:
25,36,60,99)","Model Type Entry","") 'Random generate item ID and model type Number
= Random(1,5000) itemid="B" & Number If(number Mod 4)=0 Then model="25"
ElseIf(number Mod 4)=1 Then model="36" ElseIf(number Mod 4)=2 Then model="60"
Else model="99" End If 'Set the item properties Item.regionid="SCHEDULE"
item.itemid=itemid item.itemtypeid=model item.regionloc=-1 item.exthold 0
item.groupid="MAIN" item.inthold 0 'Add the item to PRT item.Add Next i
```

```
End Sub
```

 **Note:** If necessary, review the randomize statement in the Basic Control Engine documentation.

6.3. Create a Script to add Attributes to Block Items

You can use a script to apply attributes to blocks that you create using the CIMPTRACK CimView screen.

A (page 75)	Check the Modify Item dialog box>Attributes Tab in PRT_UI
-------------------------------------	---

B (page 75)	Create a New Region Object in the Add Blocks Button's Script
C (page 76)	Create a Set of String Variables that will Hold the Attribute Names
D (page 76)	Set up the Attributes for the Block Items
E (page 77)	Check the PRT_UI window to Verify that the Attributes were Added
F (page 77)	Assign Random Values to Selected Block Item Attributes
G (page 77)	Test the new Add Blocks Button Script

1. Check the Modify Item dialog box>Attributes Tab in PRT_UI

1. Open the PRT_UI

2. Open the Modify Item dialog box for any of the blocks that you added through CimView.

3. Select the Attributes tab.

You can see that there are no attributes.

That is because when you created the attributes before you used the [prt_attributes.cfg \(page 42\)](#) file. So they will be listed with an item when you add it through the PRT_UI, not when you add it using code.

You have to write more code to add attributes to a particular item.

a. Create a New Region Object in the Add Blocks Button's Script

4. Open the script that runs when you click the Add Blocks button.

5. Create a new region object after all of your existing code, but before the `End Sub` statement:

Enter:

```
Dim Region As New PrtRegion
```

6. Specify the region where the items exist that you want to change.

Enter:

```
Region.Id="SCHEDULE"
```

- a. Create a Set of String Variables that will Hold the Attribute Names

Notice that you add the attributes `CUST ORDER #` and `COLOR` to the block items in this step.

Enter:

```
A1$ = "CUST ORDER #" A2$ = "COLOR" A3$ = "CPU SER #" A4$ = "BASE PLATFORM"
A5$ = "DRIVE HOUSING" A6$ = "SUB_ASSEMBLY A" A7$ = "SUB_ASSEMBLY B" A8$
= "ASSEMBLY HOUSING" A9$ = "POWER SUPPLY" A10$ = "PROD START TIME" A11$
= "AUTOCELL MACHINE" A12$ = "AUTOCELL TEMP" A13$ = "AUTOCELL PRESS"
A14$ = "AUTOCELL TIME" A15$ = "DVD" A16$ = "MANUALS" A17$ = "WARRANTY"
A18$ = "PACK TIME" value$ = " "
```

- a. Set up the Attributes for the Block Items

7. Create a loop that counts from 0 to the (total number of items - 1) in the region.

This will get the total number of items that exist in the region.


Enter:

```
Region.GetItemList
```

8. Now that you have the item list, you can determine how many items exist.
 - a. Use the `itemcount` property.
 - b. Loop through and modify the items to include all of the appropriate attributes.

Enter:

```
For j=0 To Region.ItemCount - 1  Region.Item(j).SetAttr A1$, Value$
  Region.Item(j).SetAttr A2$, Value$  Region.Item(j).SetAttr A3$, Value
$  Region.Item(j).SetAttr A4$, Value$  Region.Item(j).SetAttr A5$, Value
$  Region.Item(j).SetAttr A6$, Value$  Region.Item(j).SetAttr A7$, Value
$  Region.Item(j).SetAttr A8$, Value$  Region.Item(j).SetAttr A9$, Value
$  Region.Item(j).SetAttr A10$, Value$  Region.Item(j).SetAttr A11$, Value
$  Region.Item(j).SetAttr A12$, Value$  Region.Item(j).SetAttr A13$, Value
$  Region.Item(j).SetAttr A14$, Value$  Region.Item(j).SetAttr A15$, Value
$  Region.Item(j).SetAttr A16$, Value$  Region.Item(j).SetAttr A17$, Value$
  Region.Item(j).SetAttr A18$, Value$  Region.Item(j).Modify Next j
```

 **Note:** Since the `value$ (page 76)` variable is an empty string, the script that will be tested

- Will create the attributes.

- Will not assign any values.
- a. Check the PRT_UI window to Verify that the Attributes were Added

9. Open the PRT_UI window.

10. Select the SCHEDULE region.

11. Open the Modify Item dialog box>Attributes tab for any item created in CimView using the new script.

Result: The attributes should display .

- a. Assign Random Values to Selected Block Item Attributes

You can use a script to assign random selected attribute values automatically.

For this project, have values assigned to the following in the SCHEDULE region.

- Customer Order #'s
- Colors (RED, GREEN and BLUE)

12. Add the following code to the script.

A	<p>Note: After For j=0 To Region.ItemCount - 1 Enter:</p> <pre> '****Random cust. order numbers and colors***** Order\$ = " " Color\$ = " " Number = Random(5000,9999) Order\$ = Number If(Number Mod 3) = 0 Then Color\$ = "RED" ElseIf(Number Mod 3) = 1 Then Color\$ = "GREEN" Else Color\$ = "BLUE" End If '****End Random generation***** </pre>
B	<p>Note: Replace existing code to add two specific values to the .modify method Change:</p> <pre> Region.Item(j).SetAttr A1\$, Value\$ Region.Item(j).SetAttr A2\$, Value\$ </pre> <p>To:</p> <pre> Region.Item(j).SetAttr A1\$, Order\$ Region.Item(j).SetAttr A2\$, Color\$ </pre>

13. Compile the script.

14. Close the Edit Script window.

15. Close the Properties - Object dialog box.

- a. Test the new Add Blocks Button Script

16. Open the CIMPTRACK CimView screen.

17. Click the Add Blocks button.

The Item number should increase in the SCHEDULE region.

18. Open the PRT_UI window>SCHEDULE region.

19. Open the Modify Item dialog box>Attributes tab for any of the listed blocks.

The CUST ORDER # and COLOR attributes should have assigned values.

Exercise 6.3. Add Attributes for Carrier Items to the Script


A (page 78)	Add Attributes to the Add Carriers Button Script
B (page 78)	Test the new Add Carriers Button Script

1. Add Attributes to the Add Carriers Button Script

1. Create a similar script for carriers in your LOAD_CONV region as you did for your blocks.

2. Use the following attribute list to add the attributes.

CYCLE COUNT LAST CYCLE LAST CLEAN DATE ON-LINE DATE SUPPLIER
--

 **Note:** Don't worry about adding any values to the attributes you create for the carriers (like you did for the blocks.).

a. Test the new Add Carriers Button Script

3. Open the CIMPTRACK CimView screen.

4. Click the Add Carriers button.

The Item number should increase in the LOAD_CONV region.

5. Open the PRT_UI window> LOAD_CONV region.

6. Open the Modify Item dialog box>Attributes tab for one of the carriers.

The attributes you included in the script should be listed.

Script 6.3. Check the Script for Adding Carrier Attributes

This code assigns attributes to the carriers.

Since this is basic for CIMPLICITY, it is recommended that you attempt to write it first and then check what you wrote.

```
Dim Region As New PrtRegion 'Add Attributes for Carriers
Region.Id="LOAD_CONV" A1$ = "CYCLE COUNT" A2$ = "LAST CYCLE" A3$ = "LAST CLEAN Date" A4$ = "ON-LINE DATA" A5$ = "SUPPLIER" value$ = " "
Region.GetItemList For j = 0 To Region.ItemCount - 1
  Region.Item(j).SetAttr A1$, value$  Region.Item(j).SetAttr A2$, value$  Region.Item(j).SetAttr A3$, value$  Region.Item(j).SetAttr A4$, value$  Region.Item(j).SetAttr A5$, value$
  Region.Item(j).Modify Next j
```

7. Use BCE Extensions to Control the ROBOT_CONV to MACHINE Process

Now that you have your blocks and carriers created 'automatically' by using BCE scripts, you can now move ahead to use a BCE script to control your process.

You will create a script that looks at the COLOR attribute for your block item and move it to the appropriate machine.

Say that

- MACHINE0 is responsible for red and green blocks.
- MACHINE1 is responsible for assembling blue blocks.

A (page 80)	Delete Two Buttons
B (page 80)	Create a Machine Button
C (page 80)	Configure a Script for the Machine Button
D (page 81)	Review the Script for the Machine Button

E (page 81)	Test the Machine Button Configuration
--	---------------------------------------

1. Delete Two Buttons

Delete the two buttons that move the items from

- ROBOT_CONV to MACHINE0
- ROBOT_CONV to MACHINE1

1. Create a Machine Button

1. Place a single button in the center of the process
 - After the ROBOT_CONV region.
 - Between the two MACHINE regions.
2. Open the button's Properties dialog box.
3. Select **Text Button** in the Properties dialog box left-pane.
 - a. Clear the Multiline checkbox.
 - b. Enter Machine in the **String** field.
4. Select a **Button color**.
 - a. Configure a Script for the Machine Button
5. Select **Events** in the Properties dialog box left-pane.
6. Select a `Mouse Up` event.
7. Create a new script.
 - a. First create a new object that represents your region.

Enter:

```
Dim region As New prtregion
```

- a. Assign this region's id property to the ROBOT_CONV region.

Enter:

```
Region.Id="ROBOT_CONV"
```

- a. Get the item list for this region.

Enter:

Region.GetItemList

- a. Examine the COLOR attribute for the block item.

Because two items can exist in each location, you have to make sure that you are looking at the correct item within the first location.

Your item count is going to be twice as large as the number of locations that are currently filled.

Example

If there are 5 locations.

There will be 10 items, beginning with 0

- Items 0 and 1 in location 1
- Items 2 and 3 in location 2
- Items 4 and 5 in location 3
- Items 6 and 7 in location 4
- Items 8 and 9 in location 5.

As the block and carrier can exist in one location and the order in which they exist is dependant on which item entered that location first, it is up to you to determine where your block item exists within a particular location.

The structure below will determine if you are looking at the correct item.

Enter:

```
If 'the first item's item type id is 00' then 'set the index to the second item in location 1 Else
'set the index to the first item in location 1 Endif If the color attribute(index) is red or green,
update the X_MACHINE0 transition point If the color attribute(index) is blue, update the
X_MACHINE1 transition point
```

- a. Compile the script.
- b. Close the Edit Script window.
- c. Close the Properties - Object dialog box.
- a. Review the Script for the Machine Button

```
Sub OnMouseUp(x As Long, y As Long, flags As Long) Dim region As New prtregion
Region.Id="ROBOT_CONV" Region.GetItemList If region.item(0).itemtypeid =
"00" Then j = 1 Else j = 0 End If If region.item(j).attr(1).value = "RED"
Or region.item(j).attr(1).value="GREEN" Then PointSet "X_MACHINE0",1 Else
PointSet "X_MACHINE1",1 End If End Sub
```

- a. Test the Machine Button Configuration

Check that the count in the MACHINE regions on the CimView screen corresponds with the items in PRT_UI.

 **Note:** Remember to make sure there are blocks and carriers in the ROBOT_CONV region before you start.

CimView Screen

Click the Machine button a few times.

Either the MACHINE0 region and the MACHINE1 region display 1 or more items, depending on what color block was selected.

If you continue to click the Machine button, you will see that the items are dispersed to MACHINE0 and MACHINE1, based on the block color.

PRT_UI

Open the PRT_UI>MACHINE0 or MACHINE1 region (depending on which region the items move to).

A block and carrier should be listed in each location.

8. Create a Routing Control Object to Modify and Route Items

8. Create a Routing Control Object to Modify and Route Items

While you have shown that you can control your process using BCE Tracker extensions, you also have the ability to control your process using RCO.

With RCO, you don't have to learn a lot of code; simply fill in the blanks.

8.1 <i>(page 83)</i>	Create a Screen to Update Points that will Hold your Attributes
Exercise 8.1 <i>(page 88)</i>	Configure Additional Buttons for the attentry.cim Screen
8.2 <i>(page 89)</i>	Set up an Attribute Maintenance List for your RCO Configuration
8.3 <i>(page 90)</i>	Configure an AddParts Routing Control Site

8.4 (page 97)	Apply the ADDPARTS_DONE Point to the attrentry.cim Screen
8.5 (page 99)	Test the AddParts RCO
8.6 (page 100)	Add Another Function Block to the AddParts RCO
8.7 (page 102)	Implement the RCO Move Block/Carrier Configuration

8.1. Create a Screen to Update Points that will Hold your Attributes

8.1. Create a Screen to Update Points that will Hold your Attributes

Your first Routing Control Site will be responsible for updating certain attributes after the block and carrier have been merged in the **LOAD** region.

This site will then move the block/carrier to the **ROBOT_CONV** region.

 **Tip:** Refer to your Tracking model for which attributes are going to be updated.

In order for you to update the attributes with real data, you must provide some means to enter in that data, either automatically or manually. Your solution will be to create a screen that will allow a user to modify certain attributes for the block.

8.1.1 (page 83)	Create a new CimEdit Screen (attrentry.cim)
8.1.2 (page 85)	Configure CPU Serial # Entry/Base Platform Button

8.1.1. Create a new CimEdit Screen (attrentry.cim)

A (page 84)	Create a New CimEdit Screen
B (page 84)	Create an Add Parts Button on the CIMTRACK Screen
C (page 85)	Test t the Add Parts Button

D (page 85)	Set up the attrentry Basic Text Objects
E (page 85)	Add OK/CANCEL Buttons to the attrentry Screen

1. Create a New CimEdit Screen

1. Click the File button>New Window on the CIMPTRACK CimEdit screen.

A new CimEdit screen opens.

2. Save the screen as attrentry.cim.
 - a. Create an Add Parts Button on the CIMPTRACK Screen
3. Make sure the project is running.
4. Create a button under the LOAD region
5. Name the button Add Parts.

Note: (Optional) Change the button's default color.

6. Open the button's Properties - Object dialog box.
7. Select the Event tab.
8. Select a Mouse Up event on the Events tab.
9. [Open the Procedure Information \(page 64\)](#) dialog box to create a new procedure.
10. Do the following.

1	Procedure name	Enter OpenAddParts.
2	New	Click New.
3	Action type	Select Open screen.
4	Screen name	Select attrentry.cim.
5	Base project	Select CIMPTRACK.

 **Note:**

- If the Base project field is not included in the Procedure Information dialog box, make sure your project is running.
- Leave the default values for the other options.

11. Close all of the dialog boxes.

The Add Parts button is ready to be tested.

1. Test the Add Parts Button

Click the Add Parts button.

The Attentry screen should open when you click the button.

1. Set up the attentry Basic Text Objects

Create text objects for the title and to identify the fields, as follows.

Title	Add Block Attributes
Text List	CPU Serial # Base Platform Drive Housing Sub Assembly A: Sub Assembly B: Assembly Housing Power Supply

1. Add OK/CANCEL Buttons to the attentry Screen

Add OK and CANCEL buttons to the attentry screen.

Note: You will configure the buttons later.

The attentry CimEdit screen should look something like this.

8.1.2. Configure CPU Serial # Entry/Base Platform Button

A (page 85)	Configure the CPU Serial # Entry Field
B (page 86)	Configure a Toggle Action for a Base Platform Button
C (page 87)	Create an Animated Expression for the Base Platform Button

1. Configure the CPU Serial # Entry Field

The **CPU Serial #** field will enable a user to enter the serial number for the item the attributes will be assigned to.

1. Place a text object on the attrentry screen to the right of **CPU Serial #**.
2. Open the text object's Properties dialog box.
3. Select **Text**.
4. Do the following.

A	Multiline	Clear the check box.
B	Font	(Optional) Configure the font.
C	String	Enter ENTER HERE.
D	Expression	a. Create a new text virtual point. Hint. Select New Point on the Popup menu. a. Name the point CPUSerial. No initialization is required.
E	Setpoint action	Check the check box.

5. Place the text object to the right of CPU Serial #.
6. Surround ENTER HERE with a box that is configured with No fill.
 - a. Configure a Toggle Action for a Base Platform Button
7. Place a button on the CimEdit screen to the right of **Base Platform**.
8. Open the button's Properties dialog box.
9. Select **Text Button** in the Properties dialog box left-pane.
10. Do the following.

Feature	Action
Multiline checkbox	Clear
String field	Enter BASE.
Button color	Select Black.

11. Select **Colors** in the Properties dialog box left-pane.
12. Select White in the **Fill Color** field.
13. Select **Events** in the Properties dialog box left-pane.
14. Create a `Mouse Up` event.
15. Create a new procedure for the `Mouse Up` event.
16. Do the following in the Procedure Information dialog box.

A	Procedure name	Enter BASEPLATFORM.
B	New	Click New.
V	Action type	Select Toggle setpoint.
D	Point ID	<p>Create a new point.</p> <p>a. Click the Popup Menu button to the right of the Point ID field; select New.</p> <p>b. Create a new point that is:</p> <ul style="list-style-type: none"> • Named BASEPLATFORM. • Virtual • Boolean <p>Note: No initialization is required.</p>

17. Click OK.

Result: The Procedure Information dialog box closes.

a. Create an Animated Expression for the Base Platform Button

18. Select **Color Animation** in the open **Base Platform** box's Properties dialog box.

19. Click Edit.

An Expression List Attribute Animation dialog box opens.

20. Enter the following two expressions.

A	Expression	Enter <code>BASEPLATFORM EQ 1</code>
	1	<p>Fill</p> <p>a. Check the Fill checkbox. b. Select Solid in the drop down menu. c. Select Green.</p> <p>The expression/configuration are listed in the Expression box.</p>
	2	<p>Text/Font</p> <p>a. Enter YES. b. Select a font/type/size.</p> <p>Example</p> <ul style="list-style-type: none"> • @Arial Unicode MS. • Bold • 10
		New
		Click New.
B	Expression	Enter <code>BASEPLATFORM EQ 0</code>
	1	<p>Fill</p> <p>a. Check the Fill checkbox. b. Select Solid in the drop down menu. c. Select Red.</p> <p>The expression/configuration are listed in the Expression box.</p>

2	Text/Font	a. Enter NO. b. Select a font/type/size . Example <ul style="list-style-type: none"> • @Arial Unicode MS. • Bold • 10
---	-----------	---

21. Check the Fill checkbox.

22. Select Solid in the drop down menu.

23. Select Green.

The expression/configuration are listed in the **Expression** box.

24. Enter YES.

25. Select a **font/type/size**.

Example

- @Arial Unicode MS.
- Bold
- 10

26. Check the Fill checkbox.

27. Select Solid in the drop down menu.

28. Select Red.

The expression/configuration are listed in the **Expression** box.

29. Enter NO.

30. Select a **font/type/size**.

Example

- @Arial Unicode MS.
- Bold
- 10

31. Close the dialog boxes.

The attrentry.cim screen should look similar to this.

Exercise 8.1. Configure Additional Buttons for the attrentry.cim Screen

Configure buttons for the remaining block attributes. When you are done, the attrentry.cim screen should look similar to this.

8.2. Set up an Attribute Maintenance List for your RCO Configuration

Make sure SQL Server/MSDE is running before starting the RCO Configuration application as RCO configuration uses a SQL Server database to store its information.

A (page 89)	Lock RCO Configuration
B (page 89)	Create an Attribute Maintenance List

1. Lock RCO Configuration

You need to lock RCO configuration to allow you to edit the configuration.

This ensures that the RCO configuration is copied locally before configuration changes are allowed.

1. Open the TrackerCfg_UI.
2. Right-click the CIMPTRACK folder.
3. Select Lock on the popup menu.
 - a. Create an Attribute Maintenance List

You can include all of the attributes that you created for the Blocks in the RCO Maintenance list.

4. Click Tools>Attribute Maintenance on the TrackerCfg_UI menu bar.

An Attribute Maintenance list box opens.

5. Click the New button on the Attribute Maintenance toolbar.

The Attribute Maintenance dialog box opens.

6. Create a CPU SER # attribute, as follows.


Field	Enter
Description	CPU Serial #
Attribute	CPU SER #

7. Click OK.

8. Do the same thing for all of the other attributes that you created for the blocks.

- BASE PLATFORM
- DRIVE HOUSING
- SUB_ASSEMBLY A
- SUB_ASSEMBLY B
- ASSEMBLY HOUSING
- POWER SUPPLY
- PROD START TIME
- AUTOCELL MACHINE
- AUTOCELL TEMP
- AUTOCELL PRESS
- AUTOCELL TIME
- DVD
- MANUALS
- WARRANTY
- PACK TIME
- COLOR

The Attribute Maintenance list box will display all the entered attributes.

 **Note:** The attribute list in this tutorial is designed to provide you with examples of what attributes can be and give you an idea of how they function in a real situation. You will use the COLOR attribute the most. However, you will include other attributes in scripts. You will also see where you can access them in the Tracker system.

9. Click OK in the Attribute Maintenance list box.

8.3 Configure an AddParts Routing Control Site

8.3. Configure an AddParts Routing Control Site

You have already created a script that moves blocks/carriers from the ROBOT_CONV region to either the MACHINE0 or MACHINE1 region.

You will now create an RCO that moves blocks/carriers from the ROBOT_CONV region to either the MACHINE0 or MACHINE1 region.

8.3.1 (page 91)	Create the AddParts Site
8.3.2 (page 91)	Create a New Trigger Sequence

8.3.3 (page 92)	Create a New Decision
8.3.4 (page 93)	Create a New Output Logic Module (OLM)
Exercise 8.3.4 (page 96)	Configure Additional Function Blocks the AddParts OLM
8.3.5 (page 96)	Activate the New RCO Site

8.3.1. Create the AddParts Site

1. Right-click **CIMPTRACK**.
2. Select New>Folder on the RCO extended Popup menu
3. Name the folder AddPrtRC.
4. Right-click **AddPrtRC**.
5. Select New>Routing Control Object on the RCO extended Popup menu
An RCO named **New_Site** is added to the **AddPrtRC** tree.
6. Click the Popup Menu button to the right of the **RLM Point** field.
7. Select New.
8. Create the following point.

8.3.2. Create a New Trigger Sequence for AddParts

A (page 91)	Add a New Trigger Sequence to the AddParts RCO
B (page 92)	Configure the Trigger Details

1. Add a New Trigger Sequence to the AddParts RCO
 1. Right-click Triggers in the **AddParts** tree.
 2. Select New Trigger Sequence on the Popup menu.

A new trigger displays.

3. Change the new trigger name to Done.
 - a. Configure the Trigger Details
4. Click the New button above the **Trigger Points** list box.

A Trigger Details dialog box displays.

5. Do the following.

A	Trigger Point	Create a new point, as follows.	
		Name	ADDPARTS_DONE
		Type	Virtual
		Class	Boolean
		Note: This is the point you will update after you click the OK button on your Add Block Attributes screen.	
B	Trigger Type	Check On Update.	
C	OK button	Click OK.	

Your screen should look like this.

8.3.3. Create a New Decision

A (page 92)	Add a New Decision to the AddParts RCO
B (page 92)	Configure the AddParts Decision Details

1. Add a New Decision to the AddParts RCO
 1. Right-click Decisions in the **AddParts** tree.
 2. Select New Decision on the Popup menu.
 3. Change the new decision name to AddParts.
 - a. Configure the AddParts Decision Details

Do the following.

1	Decision ID	Enter 1.
2	Description	Enter Add Parts RCO Decision.
3	Source	Select LOAD.
4	Destination	Select ROBOT_CONV.

The new AddParts decision is ready for an output logic module (OLM).

8.3.4. Create a New Output Logic Module (OLM)

The Output Logic Module will set the attributes of your product with the point values that you set using the CIMPLICITY screen.

A (page 93)	Create an <code>AddParts</code> Output Logic Module
B (page 93)	Select the Set Attribute with Point Function Block
C (page 94)	Configure the Set Attribute with Point Block

1. Create an AddParts Output Logic Module

1. Click the Popup Menu button to the right of the Output Logic Module field.
2. Select New on the Popup menu.

An Output Logic Wizard opens.

3. Enter the following.

1	Name	AddParts
2	Description	Add Parts RCO Output Logic
3		Click the New button on the Output Logic Wizard toolbar.

A Select a Function Block browser opens.

- a. Select the Set Attribute with Point Function Block
4. Expand the **Core** folder in the Select a Function Block browser.

5. Double-click **Set Attribute with Point**.

A Set Attribute with Point dialog box opens.

1. Configure the Set Attribute with Point Block

Do the following.

rect 191, 236, 215, 252 ([page 96](#))

rect -3, 151, 21, 167 ([page 95](#))

rect -3, 136, 21, 152 ([page 95](#))

rect -3, 121, 21, 137 ([page 95](#))

rect -3, 107, 21, 123 ([page 95](#))

rect -3, 93, 21, 109 ([page 94](#))

rect -3, 37, 21, 53 ([page 94](#))

1 (page 94)	Description
2 (page 94)	Region Id
3 (page 95)	Region Location
4 (page 95)	Item Class
5 (page 95)	Attribute ID
6 (page 95)	Point Name
7 (page 96)	OK Button

1	Description
---	-------------

Enter Set CPU Ser #.

2	Region Id
---	-----------

1. Double-click **Region Id**.

A P1 dialog box opens.

1. Select the LOAD region in the **Region Id** dropdown menu.

1. Click OK.

3	Region Location
---	-----------------

Leave the default First Region Location value in the **Region Location** field.

4	Item Class
---	------------

1. Double-click **Item Class**.

A P3 dialog box opens.

1. Select the BLOCK class in the **Item Class** dropdown menu.

1. Click OK.

5	Attribute ID
---	--------------

1. Double-click **Attribute Id**.

A P4 dialog box opens.

1. Select the CPU Serial # class in the **Attribute Id** dropdown menu.

1. Click OK.

6	Point Name
---	------------

1. Double-click **Point Name**.

A P5 dialog box opens.

1. Click the Popup Menu button to the right of the Point Name field; select Browse on the Popup menu.

A Select a Point browser opens.

1. Select CPUSERIAL.

Note: CPUSERIAL is the point that you created for the attrentry.cim screen

1. Click OK.

7	OK Button
---	-----------

Click OK on the Set Attribute with Point dialog box.

The Output Logic Wizard now has Set CPU Ser # in the Function Blocks list box.

Exercise 8.3.4. Configure Additional Function Blocks the AddParts OLM

1. Continue to [create function blocks \(page 346\)](#) for the AddParts logic module to set the values of the other attributes.

Use these two values for all of the blocks.

Source	LOAD
Item Class	BLOCK

The blocks to configure are:

Base Platform Drive Housing Sub Assembly A Sub Assembly B Assembly Housing Power Supply

Each block is listed in the Output Logic Wizard when you finish its configuration

1. Click OK.

Result: The AddParts logic module is entered in the AddParts Decision>Output Logic Module field.

8.3.5. Activate and Verify the New RCO Site

You have successfully created a new RCO site!

A (page 97)	Activate the new RCO Site
-----------------------------	---------------------------

B (page 97)	Verify the RCO Configuration
C (page 97)	Unlock AddPrtRC

Activate the new RCO Site

1. Right-click the **AddPrtRC** folder.
2. Select Activate on the Popup menu.
3. Click the Save button on the TrackerCfg_UI menu bar.

Verify the RCO Configuration

4. Click the Error Dialog button on the TrackerCfg_UI toolbar. A blank Error Dialog box opens.
5. Click the Verify button on the TrackerCfg_UI toolbar.
Result: The Error Dialog report box will list any errors that it finds; if there are no errors the report box will remain blank.

Note: If errors are listed, click Goto to help you find where they are.

Unlock AddPrtRC

6. Right-click CIMPTRACK in the TrackerCfg_UI left-pane.
7. Clear the Lock check mark on the Popup menu.

Note: Notice that Activate and other options are now disabled.

8. Exit the TrackerCfg_UI.
9. Stop the project.

 **Note:** Perform a **CIMPLICITY configuration update**.

10. Restart the CIMPTRACK project.

8.4. Apply the *ADDPARTS_DONE* Point to the *attrentry.cim* Screen

Open the attrentry CimEdit screen

The OK and CANCEL buttons can now be configured with event/actions.

A (page 98)	Configure the OK Button Event/Procedure
B (page 98)	Configure the Cancel Button Event/Procedure

1. Configure the OK Button Event/Procedure

1. Select the OK button object.
2. Open the Properties - Object dialog box.
3. Create a Mouse Up Event.
4. Create a new OK procedure.
5. Add the following two actions in the Procedure Information dialog box.

A	Action Type	Absolute setpoint
	Point ID	ADDPARTS_DONE
	Value	1
	Confirmed	Checked
B	Action type	CloseScreen
	Confirmed	Clear

6. Click OK to close the dialog boxes.
7. Save the attrentry screen.

Result: the OK button is ready to be tested.

- a. Configure the Cancel Button Event/Procedure
8. Open the Properties dialog box for the CANCEL button object.
9. Create a Mouse Up Event.
10. Create a new CANCEL procedure.
11. Add the following two actions in the Procedure Information dialog box.

A	Action Type	Absolute setpoint
	Point ID	ADDPARTS_DONE

	Value	0
	Confirmed	Checked
B	Action type	CloseScreen
	Confirmed	Clear

12. Click OK to close the dialog boxes.

13. Save the attrentry screen.

the CANCEL button is ready to be tested.

8.5. Test the AddParts RCO

Note: If you had cleared items from

Take a few minutes to test your configuration and to see what is happening in the CIMPTRACK model.

A (page 99)	Move a Block and Carrier to the LOAD Region
B (page 99)	Check the LOAD Region in PRT_UI
C (page 100)	Look at the Block's Attributes before Opening the attrentry Screen
D (page 100)	Associate the Attributes with the Block

1. Move a Block and Carrier to the LOAD Region

1. (If CIMPTRACK is not running) restart the CIMPTRACK project.

2. Open the CIMPTRACK CimView screen.

3. Move a block from SCHEDULE to LOAD.

4. Move a carrier from LOAD_CONV to LOAD.

Result: The LOAD region displays 1 item.

a. Check the LOAD Region in PRT_UI

5. Open PRT_UI.

6. Select the LOAD region.

Result: The block you selected to move and a carrier should be listed.

a. Look at the Block's Attributes before Opening the attrentry Screen

7. Open the block's Modify Item dialog box.

8. Select the Attributes tab.

Result: The block has values for two attributes only.

- COLOR
- CUST ORDER #

a. Associate the Attributes with the Block

9. Click the Add Parts button on the CIMPTRACK CimView screen.

The attrentry.cim screen opens.

Note: (If the Status bar is not visible) Click View>Status Bar on the attrentry screen's menu bar.

10. Enter the block's Item ID in the **CPU Serial #** field.

11. Click the other fields so they all are green.

You will see in the status bar that the point value is set to 1, as you click.

12. Click the OK button, which you have associated with the ADDPARTS_DONE point.

a. Look at the Block's Attributes after Using the attrentry Screen

13. Select the PRT_UI.

14. Display the Block's PRT_UI>Modify Item dialog box>Attributes tab.

The selected attributes for the blocks should now have a TRUE value.

Note: The **CPU Serial #** value is based on the number that was randomly generated by your script.

8.6. Add Another Function Block to the AddParts RCO

A (page 101)	Open a Move Item Function Block
------------------------------------	---------------------------------

B (page 101)	Configure the Move Item Function Block
C (page 102)	Add the LOADtoROBOTCONV to the AddParts List

1. Open a Move Item Function Block

1. Select the **AddParts** decision in the TrackerCfg_UI left-pane.
2. Click Popup Menu button to the right of the **Output Logic Module** field.
3. Select Edit on the Popup menu.

The Output Logic Wizard for the AddParts logic module opens.

4. Click the New button on the AddParts Output Logic Wizard toolbar.

The Select a Function Block browser opens.

5. Expand the **PRT** folder.
6. Double-click [Move Item \(page 455\)](#).

The Move Item dialog box opens.

- a. Configure the Move Item Function Block

7. Select the following.

	Parameter	Value	Comment
A	Source Region Id	LOAD	
	Source Region Location	First Region Location	Default:
B	Destination Region Id	ROBOT_CONV	
	Destination Region Location	Last Region Location	Default
	Item Type		Leave blank so whatever type that is in the LOAD region will be moved.
	Item Class		Leave blank so the Block and Carrier will be moved.
C	Insert	ADD	

8. Click OK to close the Move Item dialog box.

9. Add the LOADtoROBOTCONV to the AddParts List
10. Check that the LOADtoROBOTCONV item is listed at the end of the Function Blocks list in the Output Logic Wizard.
11. Click OK to close the Output Logic Wizard.

The OLM should now be able to move a block/carrier from the LOAD to the ROBOT_CONV regions.

8.7. Implement the RCO Move Block/Carrier Configuration

A (page 102)	Move a Block/Carrier from the LOAD to the ROBOT_CONV Region
B (page 102)	Confirm that the Move Succeeded

1. Move a Block/Carrier from the LOAD to the ROBOT_CONV Region
 1. Open CIMPTRACK CimView screen and PRT_UI window.
 2. Make sure that a block and carrier are in the LOAD region.

Note: Check the block's **item ID** in PRT_UI.

Since the previous block/carrier did not move they will still be in the LOAD region unless you moved them manually.

3. Click Add Parts to open the Add Block Attributes screen..
4. Enter (or re-enter) the block model number.
5. Click the buttons to display YES..
6. Click OK.

The attribute values should be YES; the block and carrier should move to the ROBOT_CONV region.

1. Confirm that the Move Succeeded

You will see that your CimEdit/RCO configuration has caused changes in all of the following applications.

- CimView Screen LOAD Region
- PRT_UI LOAD Region
- PRT_UI ROBOT_CONV Region
- Block Attributes
- RCOUI AUTO COMPLETE

CimView Screen LOAD Region

On the CIMPTRACK CimView screen, the

- LOAD region displays 0
- ROBOT_CONV displays at least 1.

PRT_UI LOAD Region

The PRT_UI LOAD region should be empty.

PRT_UI ROBOT_CONV Region

The ROBOT_CONV region should list the block/carrier that were in the LOAD region.


Block Attributes

The Block's attributes that were listed on the attrentry screen should be True.

RCOUI AUTO COMPLETE

This is the first time you will open the RCOUI for this tutorial. Because you created an RCO, the RCOUI should report that activity.

1. Double-click **Production Tracking>Routing Control Objects UI** in the Workbench left-pane.
2. The `AddParts` decision should be COMPLETE in the RCOUI.

 **Note:** So far you have only one Routing Control Site so the RCOUI will open immediately. When you have more than one site (next exercise) a list will open for you to select which site to work with.

9. Create a Split Routing RCO Site

9. Create a Split Routing RCO Site

You have already created:

- A script that moves blocks/carriers from the ROBOT_CONV region to either the MACHINE0 or MACHINE1 region.
- An RCO that moves blocks/carriers from the LOAD region to the ROBOT_CONV region.

You will now:

- Create an RCO that moves blocks/carriers from the ROBOT_CONV region to either the MACHINE0 or MACHINE1 region.
- Revise the Machine button procedure to accommodate the new RCO.

Once you have the items in the ROBOT_CONV region, you have to move them to either the MACHINE0 or MACHINE1 region.

This time you will create an RCO site that is going to use two possible decisions. Instead of writing a script, you will create a Routing Logic Module that will make the decision for you.

9.1 <i>(page 105)</i>	Create a SPLIT RCO Site.
9.2 <i>(page 106)</i>	Create a RCONV_MACH0 Decision for the SPLIT Site
Exercise 9.2 <i>(page 107)</i>	Create a RCONV_MACH1 Decision for the SPLIT Site
9.3 <i>(page 108)</i>	Create a Routing Logic Module
9.4 <i>(page 108)</i>	Revise the MACHINE Button Configuration on the CIMPTRACK Screen
9.5 <i>(page 109)</i>	Create an RLM to Move Blocks by Color

Exercise 9.5 (page 110)	Add Additional Functions to the ToRegionbyColor RLM
---	---

9.1. Create a SPLIT RCO Site

! **Important:** Make sure the Tracker Configuration is locked.

The new RCO site will require an RLM Point and Trigger sequence. The procedures to create them are the same that you followed for the [AddParts \(page 90\)](#) RCO site.

The details are as follows.

A (page 105)	Create the new SPLIT site
B (page 105)	Create a SPLIT_MACHINE_RLM Point
C (page 106)	Create a Trigger Sequence for the SPLIT RCO

1. Create the new SPLIT site

Create the following.

Folder Name	RtESplit
Site Name	SPLIT (page 91)
Description	Move to selected MACHINE region

1. Create a SPLIT_MACHINE_RLM Point

Create the following point for the **RLM Point** field.

RLM Point	SPLIT_MACHINE_RLM	
	Type/Class	Virtual/Text
	Properties	String_80
	Resource ID	Tracking
	Initialization	Initialized
	Initial value	Next_Highest_Order

1. Create a Trigger Sequence for the SPLIT RCO

Configure the following [trigger sequence \(page 91\)](#).

Trigger Name	Item_Ready	
Trigger Point	SPLIT_MACHINE_READY	
	Type/Class	Virtual/BOOLEAN
Trigger Details	Point	SPLIT_MACHINE_READY
	Type	On Update

9.2. Create a RCONV_MACH0 Decision for the SPLIT Site

The SPLIT RCO will require two decisions to send block/carriers to either the MACHINE0 or MACHINE1 region.

This first decision will move items to MACHINE0.

A (page 106)	Create a RCONV_MACH0 Decision
B (page 106)	Configure a Move to Machine 0 OLM
C (page 107)	Finish the Move to Machine 0 OLM Configuration


1. Create a RCONV_MACH0 Decision

Enter the following general information about the decision.

Decision Id	0
Decision Name	RCONV_MACH0
Description	Optional
Source	ROBOT_CONV
Destination	MACHINE0

1. Configure a Move to Machine 0 OLM

1. Create a new Output Logic Module

Name	Move to Machine 0	
Function Block	Set Point with Value (page 428)	
	Description	Machine 0
	Point Name	X-MACHINE0
	Value	0
 Note: You created X_MACHINE0 when you completed the route list during the PRT configuration.		

2. Click OK.

The Set Point with Value dialog box closes.

- a. Finish the Move to Machine 0 OLM Configuration

The Move to Machine 0 output logic includes one function block.

Click OK.

The Output Logic Wizard dialog box closes. There is now a decision to move items to Machine 0.

Exercise 9.2. Create a RCONV_MACH1 Decision for the SPLIT Site

The second decision for the SPLIT RCO will move blocks/carriers to the MACHINE1 region.

1. Create a new decision named RCONV_MACH1.
2. Repeat the procedure for the RCONV_MACH0 decision, except change the 0's to 1's.

The Output Logic Wizard should look like this.

1. Click OK.

Result: The Output Logic Wizard dialog box closes. There are now two decisions.

- One to move items to Machine0.
- One to move items to Machine1.

9.3. Create a Routing Logic Module

1. Click to open the Function Block Browser.
2. Select [Select Next Highest Order Decision \(page 529\)](#).
3. Enter Select Next Highest Order Decision.
4. Click OK.
Select Next Highest Order Decision should display in the **Module List** box.

9.4. Revise the MACHINE Button Configuration on the CIMPTRACK Screen

Previously you had created a Machine button that moves items on the CIMPTRACK.cim assembly line using a script.

You can create a new Machine button or revise the existing one so it will move items using the RCO SPLIT configuration.

A (page 108)	Revise the Machine Button Configuration
B (page 109)	Test the Machine Button

1. Revise the Machine Button Configuration
 1. Open the Machine button's Properties - Object dialog box.
 2. Select the Mouse Up event to trigger a new procedure (instead of the script).
 3. Configure the new procedure as follows.


A	Procedure name	ToMachine
B	Actions	
	Toggle Setpoint	
	Point ID	SPLIT_MACHINE_READY
	Absolute Setpoint	
	Point ID	SPLIT_MACHINE_RLM

	Value	Next_Highest_Order (page 105)
--	-------	---

4. Close the Machine button's:
 - Procedure Information dialog box.
 - Properties - Object dialog box.

Result: The SPLIT RCO is ready to be tested.

- a. Test the Machine Button
5. Open the CimView screen.
6. Make sure enough items are in the ROBOT_CONV region.
7. Note the number of items in the MACHINE 0 and MACHINE 1 regions.
8. Click Machine button one or more times.
9. Note the number of items in the MACHINE 0 and MACHINE 1 regions.

 **Note:** Make sure you understand what happened, including what the [Select next Highest Order Decision \(page 529\)](#) function block does, before you continue.

9.5. Create an RLM to Move Blocks by Color

You can use RLMs to move items around the factory based on selected attributes. In CIMPTRACK the attribute that will send items to Machine 0 or Machine 1 is the block's `Color` attribute.

When the RLM is completed, blocks (using a carrier) will move based on the following criteria.

Block Color	Will Move to:
Green	Machine 0
Red	Machine 1
Blue	The machine that did not get the last Block/Carrier.

Begin the configuration as follows.

A (page 110)	Create a new RLM
B (page 110)	Configure a MACH 0 GREEN Function Block

1. Create a new RLM

Create the following new routing Logic in the SPLIT RCO site.

Name	Decision_by_Attribute
Description	Target region by color

1. Configure a MACH 0 GREEN Function Block

A MACH 0 GREEN function block will move any Green block (using a carrier) to Machine 0.

1. Open a Select Decision by Attribute Value dialog box.

The requested data will control where the Green blocks/carriers will move to, based on the selected attribute and decision.

Description	MACH 0 GREEN	
Tag	Select/enter the following.	
	Region Location	First Region Location
	Item Class	BLOCK
	Attribute ID	COLOR
	Attribute Start Character	0
	Attribute Length	0
	Value to Compare	GREEN
	Decision to select	RCONV-MACH0
	Condition	=
	Comparison	Alphabetic

2. Click OK.

MACH 0 GREEN is listed in the ToRegionbyColor RLM box.

Exercise 9.5. Complete the Decision_by_Attribute RLM Configuration

Additional configuration for the Decision_by_Attribute RLM includes procedures that you have performed previously in this tutorial.

Tracker Configuration Window

Green blocks will now move to the Machine0 region.

However function blocks need to be configured for Red and Blue blocks.

Block Color	Will Move to:
Green	Machine 0
Red	Machine 1
Blue	The machine that did not receive the last Block/Carrier.

Red Blocks to Machine 1

Create a new function block to send the Red blocks to Machine 1.

Name the block MACH 1 RED.

The selected function block will be the same type you used for Green blocks. However, select/enter the values that will send Red blocks to Machine 1.

Blue Blocks Based on the Last Move

Blue blocks can move to either Machine 0 or Machine 1 depending on which machine did not receive the last block/carrier.

Note: You created this function block type [previously \(page 108\)](#).

Note: Remember to perform the required procedures to update the CIMPLICITY project with the new Decision_by_Attribute RLM.

CimEdit CIMPTRACK Screen

Remember to modify the Machine button configuration.

Decision_by_Attribute RLM Test

Test the Decision_by_Attribute RLM

Answer 9.5. Review the Decision_by_Attribute RLM Configuration

If the Block_by_Attribute configuration performs correctly it most likely includes this configuration.

Tracker Configuration Window

Two functions need to be added to the Decision_by_Attribute RLM

Red Blocks to Machine 1

Blue Blocks Based on the Last Move

CimEdit CIMPTRACK Screen

Exercise 9.6. Create a Block by Attribute RLM

Another routing possibility is that RCO will choose between the two decisions based on the color that is already in either Machine 0 or Machine 1.

For example, some possibilities are:

If:	The next position in LOAD is:	Then RCO will choose:
Machine 0 has GREEN block/carriers	GREEN	RCONV_MACH0
Machine 1 has RED block/carriers	GREEN	RCONV_MACH0
Machine 0 has GREEN block/carriers	RED	RCONV_MACH1

 **Note:** Name the RLM, `Block_by_Attribute`.

Remember there are blocks with three different colors that have to be moved.

A (page 112)	Choose Block by Attribute Parameters
B (page 113)	Test the Block by Attribute RLM

1. Choose Block by Attribute Parameters

1. Keep the `Always Evaluate` parameter in mind when you select the options

Parameter	Description	
<code>Always Evaluate</code>	Hard, Soft or Breakable. This parameter controls what happens when the extension eliminates all available decisions.	
	Hard	All decisions will remain eliminated. Enter a wait state after the logic script has completed.

Parameter	Description	
	Soft	Roll back any decisions eliminated by this extension.
	Breakable	Roll back any decisions eliminated by this extension that have a danger point value of True (non zero). The routing modules will first use the danger defined for each individual decision; and, if no danger point has been configured for the decision, then the site-wide danger point will be used. If no danger point can be found for a decision, then the decision remains eliminated.


If you still have questions, review other details about the Block by Attribute function.

2. Remember before you complete the Block by Attribute Logic that there are:
 - Two regions.
 - Three colors.

It is also possible that one or both of the machine regions start out with no block/carriers.

Make sure that all of the colors are accounted for.

- a. Test the Block by Attribute RLM
3. Perform the required procedures to update the CIMPPLICITY project with the new Block_by_Attribute RLM.
4. Remember to Modify the Machine button's configuration (the way you did for the Next_Highest_Order RLM.

 **Note:** Think of other conditions for moving items from region to region and experiment with different routing logic modules in the SPLIT site and in other sites. If you haven't already, automate the SCHEDULE to Load, LOAD_CONV to LOAD and LOAD to ROBOT_CONV routes.

Answer 9.6. Review the Block by Attribute Function Configuration

If the `Block_by_Attribute` configuration performs correctly it most likely includes this configuration.

Tracker Configuration Window

Two functions need to be added to the Block_by_Attribute RLM

1. Block by attribute

1. Select Next Highest Order Decision

CimEdit CIMPTRACK Screen

You have to modify the Machine button configuration again to use the Block_by_Attribute RLM instead of the other two RLMs you created.

10. Use Logging with Tracker

10. Use Logging with Tracker

- Alarm Viewer: Open
- Alarm Viewer: Procedures

Alarm Viewer: Open

Double-click the **Database Logger** icon in the Workbench.

Result: The Database Logger opens.

There are three tables that can contain logging information for Tracking purposes. There are:

- PRT_GRP
- PRT_ITEM
- PRT_REG.

Alarm Viewer: Procedures

10.1 <i>(page 115)</i>	Configure a PRT_GRP Table
Exercise 10.1 <i>(page 116)</i>	Create PRT_ITEM and PRT_REG Tables
10.2 <i>(page 116)</i>	Select the Data to Log

10.3 (page 119)	Use the Global Parameter Synchronization Capability
--	---

10.1 Configure a PRT_GRP Table

A (page 115)	Create a PRT_GRP Table
B (page 115)	Configure the PRT_GRP Maintenance Events
C (page 115)	Use the Defaults for the Remaining Maintenance Event Tabs

1. Create a PRT_GRP Table

1. Select File>New Table on the Database Logger window menu bar.

A New Table dialog box opens.

2. Do the following.

A	Table identifier	Enter PRT_GRP.
B	Table type	Check Application table (Externally defined).
C	OK button	Click.

Result: The PRT_GRP Table Properties dialog box opens.

a. Configure the PRT_GRP Maintenance Events

3. Select the Maintenance Events tab in the PRT_GRP Table Properties dialog box.

4. Do the following.

A	Time/gated maintenance	Clear the check boxes: <ul style="list-style-type: none"> • Periodic • Synchronized
B	Triggered maintenance	a. Check Logging count. b. Enter 500
C	OK button	Click.

a. Use the Defaults for the Remaining Maintenance Event Tabs

5. Check Logging count.

6. Enter 500

7. Do the following on other tabs in the PRT_GRP Table Properties dialog box.

Tab	Action
Logging Attributes	Click Use Defaults.
Maintenance Actions	Click Use Defaults.
Connection	Check Use default alarm connection.
Advanced	Check Use default.

8. Close the PRT_GRP Table Properties dialog box.

The CIMPLICITY Database Logger Configuration window should look similar to this.

Exercise. 10.1. Create PRT_ITEM and PRT_REG Tables

1. Follow the same procedure you used for the PRT_GRP table.

When the table is created the default size is 200.

The size needs to be 1700.

2. Change the size by editing a datalog.idt file.

Editing the datalog.idt file is described in detail in the following Tracker Guide topic:

[Step 1. Edit the datalog.idt. \(page 591\)](#)

When you have created/configured the tables the:

- Database Logger window should look similar to this.
- PRT_ITEM line in the datalog.idt should look like this.

```
PRT_ITEM|DL_RP|1700|1000|0|100|1|0|1||0|PI|0||$LOGGING|X|0|0|0||0|5|1|0|0|3|0||C|C|0|
86400||-1|1|0|1000|60
```

10.2. Select the Data to Log

A (page 117)	Open the data_field.idt File
B (page 117)	Copy/Paste the List in the data_field.idt File
C (page 119)	Finish the data_field.idt Configuration
D (page 119)	View Logged Data

Open the data_field.idt File

In order to log specific data information, you have to edit a configuration file.

1. Open a Command Prompt window in the CIMPTRACK Workbench.
2. Switch to the CIMPTRACK project's Master directory.

```
cd master
```

3. Type the following at the Master directory command prompt.

```
idtpop data_field
```

Note: The datafield.dat file contains the fields the you want to log to each table.

For each of your tables you are going to log various information concerning your groups, regions and items.

This information is logged based on the configuration of your regions.

4. Press Enter.
5. Type the following to open data.field.idt in a text editor.

```
Notepad data_field.idt
```

Result: The data_field.idt file opens in the text editor.

Copy/Paste the List in the data_field.idt File

6. Copy/paste the following list at the end of the data_field.idt file.

```

|-* IDT file generated by IDTPOP utility v1.0
* RECORD: DATA_FIELD DLRP Field Definitions
*
* 0 TABLE_ID          DLRP Table Identifier
* 1 field_id           DLRP Field to be logged
* 2 cimp_type          CIMPLICITY Data Type of Field
* 3 field_len          Size of Field, in cimp_type units for merging /creation
* 4 offset             Offset of field in DLAP buffer for merging / creation
* 5 ins_fld_len        Size of Field, in cimp_type units for reading
* 6 ins_offset         Offset of field in DLAP buffer for reading
*
PRT_GRP|event_type|12|14|1|14|1
PRT_GRP|group_id|12|16|15|16|15
PRT_GRP|ext_hold_active|12|1|31|1|31
PRT_GRP|ext_hold_reason|12|40|32|40|32
PRT_GRP|ext_service_id|12|32|72|32|72
PRT_GRP|comment|12|40|104|40|104
PRT_GRP|user_id|12|32|144|32|144


PRT_ITEM|event_type|12|14|1|14|1
PRT_ITEM|entry_time|12|21|15|21|15
PRT_ITEM|modify_bitmask|12|4|36|4|36
PRT_ITEM|region_1_id|12|16|40|16|40
PRT_ITEM|region_1_loc|12|5|56|5|56
PRT_ITEM|region_2_id|12|16|61|16|61
PRT_ITEM|region_2_loc|12|5|77|5|77
PRT_ITEM|tracking_type|12|3|82|3|82
PRT_ITEM|item_id|12|20|85|20|85
PRT_ITEM|reference_id|12|20|105|20|105
PRT_ITEM|item_type_id|12|16|125|16|125
PRT_ITEM|assoc_item_id|12|20|141|20|141
PRT_ITEM|item_status|12|8|161|8|161
PRT_ITEM|int_hold_active|12|1|169|1|169
PRT_ITEM|int_hold_reason|12|40|170|40|170
PRT_ITEM|ext_hold_active|12|1|210|1|210
PRT_ITEM|group_id|12|16|211|16|211
PRT_ITEM|num_valid_atts|12|3|227|3|227
PRT_ITEM|att_1_name|12|16|230|16|230
PRT_ITEM|att_1_value|12|16|246|16|246
PRT_ITEM|att_2_name|12|16|262|16|262
PRT_ITEM|att_2_value|12|16|278|16|278
PRT_ITEM|att_3_name|12|16|294|16|294
PRT_ITEM|att_3_value|12|16|310|16|310
PRT_ITEM|att_4_name|12|16|326|16|326
PRT_ITEM|att_4_value|12|16|342|16|342
PRT_ITEM|att_5_name|12|16|358|16|358
PRT_ITEM|att_5_value|12|16|374|16|374
PRT_ITEM|att_6_name|12|16|390|16|390
PRT_ITEM|att_6_value|12|16|406|16|406
PRT_ITEM|att_7_name|12|16|422|16|422
PRT_ITEM|att_7_value|12|16|438|16|438
PRT_ITEM|att_8_name|12|16|454|16|454
PRT_ITEM|att_8_value|12|16|470|16|470
PRT_ITEM|att_9_name|12|16|486|16|486
PRT_ITEM|att_9_value|12|16|502|16|502
PRT_ITEM|att_10_name|12|16|518|16|518
PRT_ITEM|att_10_value|12|16|534|16|534
PRT_ITEM|att_11_name|12|16|550|16|550
PRT_ITEM|att_11_value|12|16|566|16|566
PRT_ITEM|att_12_name|12|16|582|16|582
PRT_ITEM|att_12_value|12|16|598|16|598
PRT_ITEM|att_13_name|12|16|614|16|614
PRT_ITEM|att_13_value|12|16|630|16|630
PRT_ITEM|att_14_name|12|16|646|16|646
PRT_ITEM|att_14_value|12|16|662|16|662
PRT_ITEM|att_15_name|12|16|678|16|678
PRT_ITEM|att_15_value|12|16|694|16|694
PRT_ITEM|att_16_name|12|16|710|16|710
PRT_ITEM|att_16_value|12|16|726|16|726
PRT_ITEM|att_17_name|12|16|742|16|742
PRT_ITEM|att_17_value|12|16|758|16|758
PRT_ITEM|att_18_name|12|16|774|16|774
PRT_ITEM|att_18_value|12|16|790|16|790
PRT_ITEM|att_19_name|12|16|806|16|806
PRT_ITEM|att_19_value|12|16|822|16|822
PRT_ITEM|att_20_name|12|16|838|16|838
PRT_ITEM|att_20_value|12|16|854|16|854
PRT_ITEM|att_21_name|12|16|870|16|870
PRT_ITEM|att_21_value|12|16|886|16|886
PRT_ITEM|att_22_name|12|16|902|16|902
PRT_ITEM|att_22_value|12|16|918|16|918
PRT_ITEM|att_23_name|12|16|934|16|934
PRT_ITEM|att_23_value|12|16|950|16|950
PRT_ITEM|att_24_name|12|16|966|16|966
PRT_ITEM|att_24_value|12|16|982|16|982

```


Finish the data_field.idt Configuration

7. Save the changes.
8. Close Notepad.
9. Type the following at the Command prompt.

```
scpop data_field
```

 **Note:** Result: You are now ready to log Tracker data.

View Logged Data

10. Do a project configuration update in the CIMPLICITY Workbench.
11. Restart CIMPTRACK.
12. Move blocks and carriers around the factory out of the correct order.

 **Note:** Review the [logging you enabled for regions \(page 29\)](#) for ideas to "break the rules."

13. Examine the three tracking tables and see what kind of data is displayed (if you have SQL Server).

10.3. Use the Global Parameter Synchronization Capability

The **auto_move_point** field in the PRT_REGION file supplies the ID of a text point to be set when an item moves into a tracking region due to a "point-based" automatic move.

This point is set according to the value of the NUM_SETPT_PARAM parameter in the global parameters file.

1. Expand the Project>Advanced folder in the Workbench left-pane.
2. Double-click the **Project Parameters** icon .
A New Global parameter dialog box displays.
3. Enter NUM_SETPT_PARAM in the New Global Parameter dialog box.
4. Click OK.

Global Parameter - NUM_SETPT_PARAM dialog box opens.

5. Enter one of the following values.

Value	Sets the point with the:
1	Item ID (Also if the parameter is not present.)
2	Item ID and Type
3	Item ID, Type and Reference ID
4	Item ID, Type Reference ID and Source Region

Note: You view the points values in the Point Control Panel when you move items through the system.

6. Click OK.

A Warning message displays.

7. Click OK.

11. Configure Alarming for Tracker

A (page 120)	Find the Tracker Alarms in the Workbench
B (page 121)	Change the Routing for each Alarm
C (page 121)	Test the Alarms

1. Find the Tracker Alarms in the Workbench

1. Select **Project>Alarms** in the Workbench left pane.

2. Scroll down the right pane until you find the PRT and RCO alarms.

A	CIMPLICITY alarms.
B	PRT alarms.
C	RCO alarms.

- a. Change the Routing for each Alarm
3. Make sure the project is not running.
4. Double-click PRT_DEL_REG_FULL or any of the other alarms.
The Alarm Definition dialog box for that alarm opens.
5. Select the Alarm Routing tab.
6. Add your user role to the **Configured roles for alarm** box.
Your role most likely is SYSMGR.
7. Close the Alarm Definition dialog box.
8. Repeat this procedure for all of the PRT and RCO alarms.
9. Do a CIMPTRACK project configuration update.
10. Restart the CIMPTRACK project.
 - a. Test the Alarms
11. Do the following.

A	Make sure the CIMPTRACK project is running.
B	Expand Runtime in the Workbench left-pane.
C	Double-click Alarm Viewer.

The Alarm Viewer opens.

12. Click Login! on the Alarm Viewer menu bar.
13. Open the PRT_UI.
14. Move blocks and carriers around the factory out of the correct order.

Note: Review the [alarms you enabled for regions \(page 28\)](#) for ideas to "break the rules."

15. Watch what happens.

Conclusion

If you completed this tutorial, you have taken a huge step in working with Tracker.

- By using a tracking model, you were able to configure your tracking system by editing the necessary data files.
- You were then able to use the PRT UI to add, modify, move, and delete items.
- Using the Export and Import programs, you were able to take a snapshot of your production process and modify your snapshot while tracking was down.
- When tracking came back up, you were able to import the new snapshot and continue tracking process.
- By using CimView screens, you created a run-time model of your tracking process with information and control abilities.
- By using Basic Control Engine extensions, you were able to add, modify, and delete items, and to create decision control capabilities into our process.
- Using RCO and RCO UI you were able to modify item attributes, set points, and automatically route items based on various scenarios.
- Congratulations!

Appendix

Script Keys

- Tracker Training Script Key 1
- Tracker Training Script Key 2
- Tracker Training Script Key 3

Tracker Training Script Key 1

Sub OnMouseUp(x As Long, y As Long, flags As Long)

Dim itemid As String Dim model as String Dim item As New prtItem

For i = 1 To 10

'Prompt user for the Carrier item id and model 'itemid = InputBox\$("Enter CARRIER item id, please", "Item ID Entry", "") 'model = "00"

'Random generate item id Randomize Number = Random(1,5000) itemid = "C" & Number model = "00"

'Set the item properties item.regionid="LOAD_CONV" item.itemid = itemid

item.itemtypeid = model item.regionloc = -1 item.exthold 0 item.groupid = "MAIN" item.inthold 0
'Add the item to PRT item.Add

Next i

Dim Region As New PrtRegion

'Add Attributes for Carriers

Region.Id = "LOAD_CONV" A1\$ = "CYCLE COUNT" A2\$ = "LAST CYCLE" A3\$ = "LAST
CLEAN DATE" A4\$ = "ON-LINE DATE" A5\$ = "SUPPLIER" value\$ = " "

Region.GetItemList

j=0

for j = 0 to Region.ItemCount - 1

Region.Item(j).SetAttr A1\$, value\$ Region.Item(j).SetAttr A2\$, value\$ Region.Item(j).SetAttr
A3\$, value\$ Region.Item(j).SetAttr A4\$, value\$ Region.Item(j).SetAttr A5\$, value\$
Region.Item(j).Modify

Next j

End Sub

Tracker Training Script Key 2

Sub OnMouseUp(x As Long, y As Long, flags As Long)

Dim itemid As String Dim model as String Dim item As New prtItem

Randomize

For i = 1 To 10

'Prompt user for item id and model 'itemid = InputBox\$("Enter BLOCK item id, please", "Item
ID Entry", "") 'model = InputBox\$("Enter BLOCK model type, please (valid entries:
25,36,60,99)", "Model Type Entry", "")

'Random generate item id and model type Number = Random(1,5000) itemid = "B" & Number If
(number Mod 4) = 0 Then

model = "25"

ElseIf (number Mod 4) = 1 Then

model = "36"

ElseIf (number Mod 4) = 2 Then

```
model = "60"
```

```
Else
```

```
model = "99"
```

```
End If
```

```
'Set the item properties item.regionid="SCHEDULE" item.itemid = itemid item.itemtypeid = model
item.regionloc = -1 item.exthold 0 item.groupid = "MAIN" item.inthold 0 'Add the item to PRT
item.Add
```

```
Next i
```

```
Dim Region As New PrtRegion
```

```
Region.Id = "SCHEDULE"
```

```
A1$ = "CUST ORDER #" A2$ = "COLOR" A3$ = "CPU SER #" A4$ = "BASE PLATFORM"
A5$ = "DRIVE HOUSING" A6$ = "SUB_ASSEMBLY A" A7$ = "SUB_ASSEMBLY B" A8$ =
"ASSEMBLY HOUSING" A9$ = "POWER SUPPLY" A10$ = "PROD START TIME" A11$ =
"AUTOCELL MACHINE" A12$ = "AUTOCELL TEMP" A13$ = "AUTOCELL PRESS" A14$
= "AUTOCELL TIME" A15$ = "DVD" A16$ = "MANUALS" A17$ = "WARRANTY" A18$ =
"PACK TIME" value$ = " "
```

```
Region.GetItemList
```

```
j=0
```

```
for j = 0 to Region.ItemCount - 1
```

```
***Random cust. order numbers and colors*****
```

```
Order$ = " " Color$ = " " Number = Random(5000,9999) Order$ = Number If (Number Mod 3) = 0
Then
```

```
Color$ = "RED"
```

```
ElseIf (Number Mod 3) = 1 Then
```

```
Color$ = "GREEN"
```

```
Else
```

```
Color$ = "BLUE"
```

```
End If
```

End Random generation**

```
Region.Item(j).SetAttr A1$, Order$ Region.Item(j).SetAttr A2$, Color$ Region.Item(j).SetAttr
A3$, value$ Region.Item(j).SetAttr A4$, value$ Region.Item(j).SetAttr A5$, value$
Region.Item(j).SetAttr A6$, value$ Region.Item(j).SetAttr A7$, value$ Region.Item(j).SetAttr
A8$, value$ Region.Item(j).SetAttr A9$, value$ Region.Item(j).SetAttr A10$, value$
Region.Item(j).SetAttr A11$, value$ Region.Item(j).SetAttr A12$, value$ Region.Item(j).SetAttr
A13$, value$ Region.Item(j).SetAttr A14$, value$ Region.Item(j).SetAttr A15$, value$
Region.Item(j).SetAttr A16$, value$ Region.Item(j).SetAttr A17$, value$ Region.Item(j).SetAttr
A18$, value$ Region.Item(j).Modify
```

Next j

End Sub

Tracker Training Script Key 3

```
Sub OnMouseUp(x As Long, y As Long, flags As Long) Dim region As New prtregion region.id =
"ROBOT_CONV" Region.GetItemList If region.item(0).itemtypeid = "00" Then
```

```
J = 1
```

```
Else
```

```
J = 0
```

```
End If If region.item(j).attr(1).value = "RED" Or region.item(j).attr(1).value = "GREEN" Then
```

```
PointSet "X_MACHINE0",1
```

```
Else
```

```
PointSet "X_MACHINE1",1
```

```
End If End Sub
```

Chapter 3. Tracker Guide

About Tracker (Base)

The Tracker basic option is the premier software for tracking and making decisions for building products through the production process.

The primary components of the base Tracker option are:

circle 373, 64, 47 [About Production Tracking \(page 126\)](#)

circle 168, 300, 56 [About the Routing Control Objects Module \(page 324\)](#)

PRT <i>(page 126)</i>	database tracks products as they move through a production process.
RCO <i>(page 324)</i>	uses its data to perform enhanced production routing decisions at runtime.

Production Tracking: PRT

About Production Tracking

Production Tracking (PRT) is a [specialized database \(page 128\)](#) designed to track products as they move through a production process.


The following topics provide a detailed description of PRT and how it is configured.

- Production Tracking system definitions and guidelines.
- Production Tracking Model layout planning.
- Production Tracking Configuration
- Production Tracking User Interface (PRT_UI).

PRT Technical Reference.

PRT is fully integrated with CIMPLICITY software's base system functionality to enhance its already powerful monitoring capability in a full range of computer-integrated manufacturing environments.

The CIMPLICITY base system functionality (Point Management, Alarm Management, Database Logging facilities, and a full-functioned User Interface) enables you to collect data for reporting and to view data in lists, graphic status displays, and alarms.

 **Note:** If you have installed Tracker Extended Production, you will find that PRT's capabilities are further enhanced. However, it is recommended that you first understand the Tracker basic components, Production Tracking and Routing Control Objects.

The Tracker module also has the ability to integrate with many of the industry-standard data communications products.

There are also several optional application subroutines for customizing your application.

Production Tracking System Definitions and Guidelines

Production Tracking System Definitions and Guidelines

It is important to have a basic understanding of the parts that make up a production tracking (PRT) system and how they relate to each other.

The following sections provide the PRT definitions and guidelines.

1 (page 129)	Items
2 (page 137)	Item types
3 (page 137)	Item classes
4 (page 138)	Regions
5 (page 145)	Region Locations
6 (page 146)	Routes
7 (page 147)	Transition types
8 (page 154)	Groups

9 <i>(page 155)</i>	PRT services
10 <i>(page 155)</i>	PRT and CIMPLICITY software interface components

PRT Database Hierarchy

The PRT database supports the tracking model and houses all of the tracking and configuration data. The following diagram illustrates the hierarchy within the database. Note that data at the base level are related to one or more database types in the next level up. This kind of relational database serves to simplify and expedite searches for items in the tracking model.

rect 25, 67, 169, 161 [9. PRT Services: Defined \(page 155\)](#)
 rect 27, 178, 168, 244 [8. Groups: Defined \(page 154\)](#)
 rect 25, 254, 170, 341 [4. Regions: Defined \(page 138\)](#)
 rect 26, 350, 170, 438 [5. Region Locations: Defined \(page 145\)](#)
 rect 197, 426, 334, 483 [2. Item Types: Defined \(page 137\)](#)
 rect 194, 495, 335, 551 [Item Attributes: Defined \(page 132\)](#)
 rect 305, 165, 417, 239 [3. Item Classes: Defined \(page 137\)](#)
 rect 307, 266, 416, 338 [2. Item Types: Defined \(page 137\)](#)

PRT Overview Graphic

The following (non-scale) diagram illustrates the primary concepts behind Production Tracking.

[\(page 128\)](#)

1 <i>(page 147)</i>	Region Entry Point	Point where a device informs the tracking model that a particular item has entered the region.
2 <i>(page 145)</i>	Region Location	Physical area within a region that holds items; more than one item can reside in a region location.
3 <i>(page 138)</i>	Region	Bounded by devices at entry and exit points.
4 <i>(page 147)</i>	Region Exit Point	Point where a device informs the tracking model that a particular item has exited the region.
5 <i>(page 129)</i>	Item and Attributes	Part/Assembly in the manufacturing process.
6 <i>(page 154)</i>	Group	Holds regions that are either physically or functionally associated - all regions must belong to a group. Welder Lines 1 and 2 are functionally grouped together.

7 (page 137)	Item Type	Parts with similar characteristics are logically grouped together, as with parts in Welder Lines 1 and 2.
8 (page 155)	Operator Interface	Points where the operator can interact with the tracking system.
9 (page 146)	Route	An item's region to region progress through the production process

1. Items

1. Items: Defined

The primary purpose of Production Tracking is to monitor the progress of a product as it is being manufactured in your factory.

Products, however, are rarely made up of just one or two parts; products are frequently comprised of several hundred different parts. For example, a personal computer or an automobile each has literally hundreds of parts that make up the whole.

Tracker has the flexibility to monitor parts in production as they come together to make up your finished product.

An Item represents the physical part or assembly that is being tracked. It is assigned an Item Id and/or a Reference ID for identification purposes (serialized items only).

Item types and features include:

Item Types		
	Serialized items	Defined
		Guidelines
	Non-serialized items	Defined
		Guidelines
Item Features		
	Item attributes	Defined
	Item status	Defined
	Item associations	Defined
		Parent-Child
		Tracking Rules
	Items	Guidelines

Serialized Items

Serialized Items: Defined

Serialized items are the parts or assembly that you want to positively identify as they are tracked through production.

Serialized items in the core Tracker (Order Execution Management provides increased capabilities.):

- Have one or two unique identifiers.
- Have 40 dynamic attributes that are configured by the user.
- Allow positive identification as they progress through the tracking system.
- Must be tagged so that they can be recognized by a sensor device, e.g. a bar code or radio tag.

The following diagram shows a computer monitor (serialized item) identified by the tracking system. The sensor reads the barcode on the monitor and relays the data back to the tracking system where an operator can access it.

Example

1	CIMPLICITY server with the Tracking system.
2	A sensor reads the bar code on an item and relays the data back to the tracking system.
3	Bar code on a serialized item's part.
4	Serialized items that are uniquely identified by their serial number.

Serialized Items: Guidelines

1. When a serialized Item first starts through the production process, PRT requests the:
 - Item type,
 - Status and
 - Attribute data.
2. As the item goes through the production process, tracking data is collected and logged.

PRT processes items differently depending on the method used to start production.

If production start is originated by a:

CIMPLICITY Point	Configuration data determines if Production Tracking will request an external process for additional information.
Manual transaction or Application interface message	Additional tracking data may be included in the message. Also, an external process may be queried for additional information.

The following guidelines apply to serialized Items:

- Each serialized item is assigned a unique Item ID.
- Serialized items can also be identified and tracked by a reference ID.

A reference ID acts as an alternate key to locate the Item and Item ID.

- The reference and Item IDs can be automatically associated through configuration data.
- Reference and Item IDs can only be associated when an item is positively identified by a transition indicator.
- A CIMPLICITY Point must be configured to report the reference ID through the transition indicator.
- Reference and Item IDs can also be associated through an application interface message.
- Out of sequence errors apply to serialized items only. (Unexpected item errors apply to non-serialized items.)
- The route for a production start must be configured to provide the item ID if the item is to be serialized.

Non-Serialized Items

Non-Serialized Items: Defined

Non-Serialized items are parts, assemblies or other articles that you are not interested in positively identifying, but you want to know their location.

Non-Serialized items:

- Do not have a unique identifier.
- Hold two attributes, a parent ID and an item type.
- Do not require a tag for sensor recognition.
- Can be pallets, carriers, bins or other articles that are reused in production.

Example

The following diagram shows a cart, which is an item that is reused in a factory, and trips a limit switch that is at the entry point of a Carriage Return Region 5 (CR5). The tracking system is notified that an item has passed into CR5.

1	CIMPLICITY server with the Tracking system.
2	Simple limit switch signal the tracking system that something passed.
3	The server receives the signal.

Non-serialized Items: Guidelines

PRT provides the reserved Item type \$UNKNOWN, which can be associated with a non-serialized item at the start of production.

 **guide:** **Guidelines** that must be followed when using this association are:


- The association can only take place when a non-serialized Item passes a transition indicator.
- The transition indicator must be configured as an Item type detector.
- A CIMPLICITY Point must be configured to report positive identification of the Item type.
- An application interface message can associate a non-serialized Item with an unknown Item type.
- The Item type of the non-serialized item must be identified later in the production process.
- Unexpected Item errors will apply to Non-serialized Items. (Out of sequence errors will apply to Serialized Items only.)

Item Attributes: Defined

In the PRT database, an item is simply a record and the attributes are fields in the record that you configure, such as color or model number.

Attributes:

- Are configured by the user.
- Can change during the production process.
- Provide information that enables positive identification of an item and its location.
- Can be dynamically created from one item when it is associated with another.

 **Note:** You can positively identify a serialized item by assigning it a unique identifier, such as a serial or part number; you can also identify it by its attributes.

Example

The following diagram shows how an item can dynamically become an attribute of another item during production.


In a sample automobile factory six and eight-cylinder engines are being tracked, along with two and four-door car frames. These are all serialized items that are positively identified through the production process.

When the engine is inserted into a car frame, the engine dynamically becomes an attribute of that car frame in the tracking system.

1	The 6-cylinder engine is a serialized item.
2	The 2-door car frame is a serialized item.
3	The 6-cylinder engine is a:

- Serialized item.
- Attribute of the 2-door car frame.

This happens dynamically in the tracking system during production of the automobile.

 **Note:** Production Tracking counts PRT items based on:

- Region(s)
- Item class
- Various **item attributes** criteria

An item is included in a count if its attribute matches a configured set of criteria. In order for an item to be included in the count, the attribute value must be compared against a user-defined value.

Item Status: Defined

A Tracking Region can be configured to affect an Item's status due to Item-hold specifications.

For example, a shortage of an Item attribute occurs if a required physical Item attribute (e.g. A/C or seat style) is shown as being in short supply in a Region on the production line.

An Item's status can be modified by:

- An external process.
- A manual transaction.


An external process:

- Can modify an Item's status to Hold when shortage attributes exist.
- Is configured to provide Item-hold information for Serialized Items.
- Enables PRT to modify the status of all Items residing in a Tracking Group.
- Modifies the status to Hold for each new Item entering the Tracking Group.
- Can be configured to generate a CIMPPLICITY alarm when an Item's status is modified to Hold.
- Can deactivate Item-hold status on all affected Items in a Tracking Group.
- Will not impact a Hold placed by a manual transaction.
- Monitors the Item's status; and, if capable, physically holds the item at a pre-configured point on the production line.

A manual transaction:

- Can be executed to modify an Item's status to Hold.
- Can have comments attached to describe the reason for the Hold.
- Is held separately from a Hold placed by an external process.
- Can deactivate Item-hold status on all affected items.

- Will not impact a Hold placed by an external process.

 **Note:** The concept of item hold specifications apply only to Serialized Items.

Item Associations

Item Associations: Defined

Production Tracking permits Item associations to be formed:

- Automatically in a Combine Items Tracking Region

Automatic Item associations occur in Combine Items Tracking Regions where configuration data specifies the Item types that will participate in an Item association and the role each Item will play in the relationship. Each time an Item enters a Combine Items Tracking Region, where automatic Item associations occur, the Item type is checked to determine if this Item participates in an Item association. An automatic Item association takes place when each of the Items in the association have arrived in the Combine Items Tracking Region.

- By request through a manual transaction or an application interface message.
- Between two Item types, if your CIMPLICITY package includes extended attributes, when an item of its associated type is processed by CimBasic extensions or by certain RCO function blocks, e.g. Substitute Order. This type of association is created by adding an extended attribute, [ASSOCIATE \(page 223\)](#), to one of the Item types in the association. The value of this extended attribute is the Item type (Item Type ID) of the associated item.

Note: A complementary ASSOCIATE extended attribute is automatically added to the associated Item type that refers to the first Item type.

Item Associations: Parent-Child

A Parent-Child Item Association requires two Items:

- The first Item must be serialized and is the parent Item in the relationship.
- The second Item must be non-serialized and is the child Item in the relationship.

The Parent-Child Association:

- Makes non-serialized items serialized in the sense that they are now associated with an identifiable item.
- Ensures that both items stay logically connected.
- Ensures that both items become physically connected at the appropriate time in the manufacturing process.

Tracker allows the parent Item may have more than one associated child Item.

A parent Item and a child Item can be associated by request, whether they reside in the same tracking region or in separate regions.

The request must specify the:

- Sequential tracking region in which the child Item resides.
- Item type of the child Item (non-serialized).
- Sequence location of the child Item in the region.

The following diagram shows how a serialized item and a non-serialized item form a parent-child association. When two items comes together during production, and then must separate for processing, they form a parent-child association. By forming an association, the two parts can come together later on in the production process for assembly.

Example

[\(page 135\)](#)

1	A monitor is serialized and becomes the parent item when it is joined with the base.	
2	Drill region	A sensor reads the bar code on a serialized item.
3	Drill region	The monitor and base are joined to have holes drilled to match the two pieces.
		The base is non-serialized and becomes the child of the serialized item - the monitor.
4A	Monitor Inspection region	The monitor is separated from the base for inspection, but is still associated with the base as the parent item.
4B	Paint region	The base has been separated from the monitor to be painted, but is still associated with the monitor as the child item.
5	Assemble region	The Parent-Child association in the tracking system has placed the two matching pieces together for assembly.

Item Associations: Tracking Rules

Item Associations can be specified at runtime via Application Interface messages.

The message will provide the child item type and the region ID where that item is the:

- Next item to exit the region and
- Item ID of a parent item (serialized item) the child (serialized or non-serialized item) is to be associated with.
- Serialized Items may not be child members of an association.

Warning: The association of a non-serialized item to a serialized parent item can become corrupted or lost in the following three ways.

- If a sequence error is detected for a non-serialized item in a region where the item has an associated parent Item.
- If a non-serialized Item crosses a PRT service boundary and the PRT process for the first service is unable to provide the parent item data for the non-serialized item.
- A non-serialized item has an associated parent item and the non-serialized item is placed on an item carrier with another item of the same item type.

PRT has no way of re-associating the Items when an association is lost except through a manual user transaction.

Item Associations: ASSOCIATE Extended Attribute

Items can be associated by an extended attribute [ASSOCIATE \(page 223\)](#) . All attributes (system, standard and extended) of either item can be referenced within the program where ever the attributes of its associated item are available. The associated item's attributes appear with an @ prefix.

Items: Guidelines

Item Identifiers

- Processing of Item Identifier points works in the following manner:

1. The first character of the point value is checked when the point value is received.
2. If the first character:

Is not \$	The point is expected to contain the configured identifier and that Item is searched for and moved.
Is \$	The rest of the value is checked against the reserved words.

If the rest of the value is:


Found to be a reserved word	The corresponding action is taken
Not found to be a reserved word	The rest of the value is taken as an Item Type ID and processing continues as if an Item Type detector was activated.

3. If an Item identifier or Item type identifier is set and the Item cannot be found, an Item with that ID, reference ID or type (as appropriate) is created.
 - Currently, the only reserved word is TMOUT, and the action taken is to ignore the point.
 - An automatic item identifier is equivalent to a user transaction or an application interface message, in that it provides PRT with positive unique identification of the item.

If more than one entry transition indicator feeds a tracking region, items are stored in the region in the order in which the entry transition indicators are received by PRT.

2. Item Types: Defined

Item types are a way of logically grouping items together in order to simplify and expedite searches in the PRT database.

 **Important:** Every item defined in the tracking system **must** have an item type.

Item Types:

- Classify groups of items in the tracking system.
- Are independent of region, group or service.

An Item tracked by PRT must always reside in a Tracking Region.


Example

A sample automobile factory manufactures mini-vans and sedans, and has three engine sizes.

The following chart provides an example of grouping items together into types, e.g. Car doors.

3. Item Classes: Defined

Item classes are a way of grouping together item types in order to further simplify and expedite searches in the PRT database.

 **Important:** Although item types do not have to belong to an item class, it is strongly recommended that you create item classes. You can create just one item class to which all of your item types can belong.

Item Classes:

- Group item types together to create a larger subset in the database.
- Are independent of region, group or service.

Example

The item types in the previous example are grouped into four major classes, creating an organized and logical hierarchy of information.

The following chart demonstrates how you can create item classes based on defined item types.

1	Item types
2	Item classes

4. Regions

4. Regions: Defined

A tracking region is bounded by devices (an entry and an exit transition indicator) that inform the tracking model that an item has entered or exited the region. The Region's entry and exit indicators are responsible for providing an Item's location data.


PRT requires positive identification of an item at the entry to the first region configured for a PRT process (i.e. node). Positive identification consists of the Item ID or Reference ID for Serialized Items and the Item Type for Non-serialized Items.

- A Region Entry/Exit Indicator consists of
 - An automatic Item identifier,
 - An automatic Item Type detector,
 - A limit switch,
 - A User Interface transaction or
 - An Application Interface message.

Except for production start and production stop regions, the exit transition indicator for one region must always be configured to be the entry transition indicator for the destination region in the production line.

A full complement of the PRT configuration data resides on each Node.

- Configuration data determines if an Alarm is generated when:
 - An Item enters a Region, and that Item occupies a location exceeding the capacity of the Region,
 - An Item enters a Combine Items Tracking Region, and that Item's presence in the Region exceeds the capacity of a location in the Region.
 - A region must be a member of at least one Tracking Region Group. All regions in a tracking group must be configured for the same PRT process service ID.

 **Important:** Each tracking region must be associated with a primary and a secondary region type.

- Primary regions
- Secondary regions

Primary Region Types

Primary Region Types

A region represents the physical space on the factory floor through which parts/assemblies flow during the production process. Regions are bounded by devices that alert the tracking model when an item passes. Different kinds of regions require unique configuration.

The three primary region types are:

Pooled region	Defined
Sequential region	Defined
	Sequence Monitoring
Shifting region	Defined

Pooled Region: Defined

Pooled Region: Defined

A pooled region tracks that an item entered or exited the region, but does not track items sequentially.

An example of a pooled region would be a warehouse zone. Items going into a warehouse do not come out in the same order, and would therefore not be tracked in any particular sequence.

Example

1	A warehouse was configured as a Pooled region because the items will not enter and exit in the same order.
2	Bar codes on products are scanned by employees when the products enter and exit the region.

Pool Tracking Region Rules

A Pool tracking region maintains the Items in the tracking queue in the order the Items entered the region.

A Pool region must have the following when it monitors the **items** in the region by their:

- Unique Item identifiers
- An automatic Item Identifier, or
- A User transaction (specifying Item ID or Reference ID), or
- An Application Interface message (specifying Item ID or Reference ID) as a Region Entry Indicator, or

- Be preceded by a Sequential Tracking Region also monitoring the static sequence of the Items in the region by their unique Item or Reference Identifiers.
- Item Type
- An Item Type detector, or
- A User transaction (specifying Item Type), or
- An Application Interface message (specifying Item Type) as a Region entry Indicator, or
- Be preceded by a Sequential Tracking Region also monitoring the static sequence of the Items in the region by their Item Type.

Sequential Region

Sequential Region: Defined

A sequential region tracks the order in which items enter and exit the region.

Skipped items can be deleted or sent to a user-configured detainment region. An example of a sequential region would be an oven zone where items that are sent into the oven must come out in the same order or they will be exposed to heat beyond the allotted time limit.

Example

1	An operator interface removes defective parts from the line before they reach the oven.
2	Part #4 is defective and will be scrapped by the operator. Tracker will send the item to a user-configured detainment region.
3	Items are tracked in sequence. The first item in must be the first item out.


Sequential Region: Sequence Monitoring

A sequence error may be detected at the exit transition indicator of a Sequential Tracking Region when an Item does not arrive at the transition indicator in the expected order.

You will configure the action to be taken in a Tracking Region when an Item sequence error is detected.

The five configuration options are to:


- Immediately place the item in the corresponding detainment Region.
- Change the Item's status to DELAYED. Once the configured number of Items has exited the Region, the missing Item is placed in the corresponding detainment Region.
- Scrap the missing Item.
- Change the Item's status to DELAYED. Once the configured number of Items has exited the Region, the missing Item is scrapped.
- Do nothing - sequence monitoring disabled.

 **Note:** Sequential Tracking Regions of size one do not detect sequence errors.

Synchronization

In the event a sequence error is detected, PRT allows re-synchronization of the tracking environment, which can be done either manually or automatically. Further, if Automated Region Validation is configured, sequence processing occurs when the Items in the Region are reordered.

Options are:

<p>Manual Re-synchronization</p>	<p>The tracking environment can be manually re-synchronized by reconciling the Tracking Region queues using available functions in the PRT User Interface. Available functions are as follows:</p> <ul style="list-style-type: none"> • Add Item • Delete Item • Move Item • Fetch Item • Advance Item • Reorder Region
<p>Automatic Re-synchronization</p>	<p>Automatic re-synchronization takes place when a serialized Item exits a Sequential Tracking Region in an out-of-sequence condition and is then re-synchronized in the destination region. PRT will process the Item according to the configuration for detected sequence errors in the Region. PRT will:</p> <ul style="list-style-type: none"> • Ensure the serialized Item is correctly identified in the queue of the destination region. • Assume the sequence of the Items in the source region is incorrect until the missing Item is scrapped or the delayed Item is removed. • Identify the item in the destination region when it is removed from the detainment region and placed in the destination region. <p> Note: Regions configured with Automated Validation will potentially re-synchronize every Item as they enter the head location.</p>
<p>Items Scrapped</p>	<p>Production Tracking allows Items to be scrapped through a manual transaction or an application interface message. You can configure your Tracking Model to log the scrap event. PRT logs:</p> <ul style="list-style-type: none"> • The Item ID and Type for serialized Items • The Item Type for non-serialized Items. • Any comments entered for description purposes.

Sequential Tracking Region Rules

A Sequential region must have the following when it monitors the static sequence of the **items** in the region by their:

- Unique Item identifiers
- An automatic Item identifier or a User transaction (specifying Item ID or Reference ID), or
- An Application Interface message (specifying Item ID or Reference ID) as a Region Entry Transition Indicator, or

- Be preceded by a Sequential Tracking Region also monitoring the static sequence of the Items in the Region by their unique Item identifiers.
- Item type
- An Item type detector, or
- A User transaction (specifying Item type), or
- An Application Interface message (specifying Item type) as a Region Entry Transition Indicator, or
- Be preceded by a Sequential Tracking Region also monitoring the static sequence of the Items in the region by their Item Type.

Shifting Region: Defined

A shifting region supports empty locations intermixed with occupied locations within the region.

As the item moves through the region, it is moved ahead one region location, rather than moving directly to the end location in the region.

As new items are introduced into the region, current items in the region move ahead one region location; the item at the head location is moved to a Shifting Transition region, which is selected when the shifting region is configured.

An example of a shifting region would be in an automobile factory where empty carriers on the line are counted as empty locations.

Example

The region in this example is bounded by sensors and has three region locations.	
1	This carrier will enter the Shifting region to occupy region Location 3.
	The other locations in the region will shift forward one location.
2	This carrier occupies Location 2 and will shift to Location 1 when a new item enters the region.

Secondary Region Types

Secondary Region Types

Once you have determined the primary type for your regions, you must select a secondary type. Often, regions may not be linked along a straight line and can have multiple paths or may combine items into a common region, or disperse items into separate regions.

The three secondary region types are.

Normal region	Defined
Combine region	Defined
	Tracking rules

Disperse region	Defined
-----------------	---------

Normal Region: Defined

A normal region is a basic region through which an item or group of items flow.

For example if there are two assembly lines in a row, which are named Line 1 and Line 2, and employees are working on one item at a time in both lines, this would be configured as a normal region.

The primary region type for Line 1 and Line 2 could also be sequential region because each item must progress in the order in which it entered the region.

Example

1	Operator interface
2	Items are tracked through the plant.
	Each item occupies one region location.

Combine Region

Combine Region: Defined

A combine region allows multiple item entries to combine into one region location from separate regions. The combine region has one associated region location. An example of a combine region would be where items coming from three different conveyor belts converge into a robot region for processing. The robot region has only one region location that holds multiple items.

Entry Transition Indicators reside physically on the same Item Carrier in that region. Multiple Items may also enter through one Entry Transition Indicator by PRT receiving multiple notifications from the Transition Indicator. The Items themselves are in no way required to be associated.

Example

Combine Region: Tracking Rules

In a Combine region multiple items can enter the region through:

- Separate region Entry Transition Indicators and reside physically on the same Item Carrier in that region.
- One Entry Transition Indicator by PRT receiving multiple notifications from the Transition Indicator. The Items themselves are in no way required to be associated.

Required for Combine regions

- A single Exit Transition Indication is required from the region.
- The region must be configured to have a region capacity of one.
- Positive Item Type detection is required for each Item entering the region.

Allowed in Combine regions

- The Items merely reside on the same physical carrier, they are not required to be associated in any way.
- Multiple region Entry Transition Indications are allowed.
- It is not required that all Items arrive at the region before the carrier exits the region. The Items that arrived at the carrier before a transition out of the region is detected are the Items that moved together with that carrier.

Other rules for Combine region items

- A Tracking Region designated to execute automatic Item Associations must be a Combine Items Tracking Region.

Disperse Region

Disperse Region: Defined

A disperse region is the opposite of a combine region in that dispersed items are separated into successive regions.

1. A single item carrier enters the region carrying multiple Items.
2. Multiple item carriers (or equivalent) will exit the region carrying one or more of the items which resided on the entry item carrier.

This example shows how these three items can be separated into three successive regions once they have been processed.

Example

Disperse Region: Tracking Rules

Required for disperse regions

- A single region entry transition indication is required.
- The region must be configured to have a region capacity of one.

- Positive item type identification is required for individually dispersed Items.
- If any of the items in a disperse region is serialized, it must be moved before the other items are dispersed from the region.

Allowed for disperse regions

- Multiple region exit transition indications are allowed.

Special item behavior in disperse regions

- If a region exit transition indicator has blanks specified for the type of item that may exit that point all items are tracked through that exit.
- If a region exit transition indicator has configured a specific item type for that exit, only Items of that type are dispersed from the group of Items.
- If the source region is not a disperse region, the exit transition will move all items at the source location. Otherwise, only the specified item is moved.

5. Region Locations: Defined

A region location is the physical space within a region in which an item or items reside.

Region Locations allow you to:

- Isolate items in the production process.
- Have multiple items reside in the same location.
- Track groups of items.

Example 1

There is a one-to-one ratio between items and region locations.

1	A conveyor belt has 7 locations.
2	One of the locations.

Example 2

Bin One region has only one region location that is capable of holding multiple items.

1	Bin One has one region location that holds multiple items.
2	A sensor informs the tracking model that items 1, 2, 3 and 4 have entered Bin One region.

 **Note:** If multiple Items reside at a single Tracking Region Location, one and only one of the Items may be a Serialized Item.

6. Routes: Defined

6. Routes: Defined

In most factories a product does not follow a straight line all the way through the manufacturing process. There can be several occurrences during production where a part or assembly heads down a specific path to have an option added or to be painted a different color. All of these different paths are called routes.

Routes:

- Link one or more regions.
- Direct the flow of parts and assemblies in production.

There can be multiple routes from one region to another.

Example

1	Device at region 1's entry point.
2	Device at a region 1's exit point and region 2's entry point.
3	The path linking the two regions is a route.
4	Device at region 2's exit point.

Routes: Guidelines

- Item movement is directed from the Source Tracking Region to the Destination Tracking Region and cannot reverse direction along a single route.
- In a Tracking Route, if the Item Type is specified as a blank field or the wildcard character, all Types in the region are assumed to be allowed to travel this route between the Source and Destination Tracking Regions.
- The Item Type may also contain a mix of text and wildcard characters.
- The wildcard character is an asterisk (*).
- PRT determines what Item to move to a Destination Region by using these two rules.

If positive identification of the item (either ID or type) through information provided by the transition indicator:

Is made available to PRT	The item or item type identified is moved from the source region to the destination region.
Not provided to PRT	The information PRT maintained about the Item(s) in the Source Region will be used.

7. Transitions

7. Transitions

Transitions refer to movement of an item from a source to destination region.

- Transition indicators: guidelines
- Transition types

Transition Indicators: Guidelines

A region transition indicator reports the movement of an item between two tracking regions.

- Each transition indicator defines unique routes that exist between two regions.

A transition indicator is therefore the:

- Exit transition indicator for the source tracking region and
- Entry transition indicator for the destination tracking region.
- The same transition indicator may specify a valid route between two regions for more than one Item Type.
- When a region entry transition Indicator is received by PRT, PRT updates its tracking data regardless of what tripped the indicator.

PRT cannot distinguish between valid item transition indications it receives and accidental item transition indications it receives (e.g. person walks in front of limit switch, PRT receives the transition indication and updates the system to reflect Item movement).

- Production start must always occur in a tracking region with an entry transition indicator that is not the exit transition indicator for any other tracking region.

Transition types

Limit switch.
Type detector.
Item identifier.
Reference identifier.
Associate reference ID to Item.
Associate Item Type to item.
Positive edge trigger.

Negative edge trigger.

Limit Switch Action: Guidelines

The first location of the source region is checked for items.

Actions triggered by what items exist are:	
If	No items exist in the source region,
Then	No items are moved.
If	Multiple items exist at a location,
Then	All items at the location move.
Actions triggered by the specified handling of a moved item are:	
If	Not handled along the route,
Then	The item moves and a message is logged warning the user.
If	Not allowed in the destination region,
Then	The item is deleted and a message is logged warning the user.

 **Note:** Other **Guidelines for Limit Switch** include:


1. The transition point may be an integer or character string point.
2. The first 36 characters of a setpoint, if configured, are:

First 16 characters	Item type
Next 20 characters	Either the: <ul style="list-style-type: none"> • Item ID: for a serialized item • Associated or parent item ID: for a non-serialized item.

3. If a limit switch is used to transition an item and the source region is empty, the limit switch is ignored.
4. The route for a production start region can specify a limit switch as the transition indicator if the item type is explicitly defined for the route.

Type Detector Action Guidelines

Tracker cross-references the given integer with the item type.

 **Important:** A **PRT_TYPE TRN** file must be configured to describe the cross-reference between the integer entered and the item type to be moved.

Actions triggered in the region are:

If	The item type is found in the source region
Then	The item is moved to the destination region.
If	The source region is a disperse region
Then	Only the item whose item type is cross-referenced with the given integer moves.
If	Items exist at a location and the source region is not a disperse region
Then	All items are moved.
Actions triggered by the handling of a moved item are:	
If	Not handled along the route
Then	The item moves and a message is logged warning the user.
If	Not allowed in the destination region
Then	The item is deleted and a message is logged warning the user.

Item Identifier Action: Guidelines

For all point values except values that begin with a \$

Tracker searches for the item with the given item ID in the entire model.

Actions triggered by what Tracker finds in the entire model are:	
If	The item is not found.
Then	A message is logged.
If	The source region is a disperse region.
Then	Only the given item is moved.
If	Multiple items exist at a location and the source region is not a disperse region.
Then	All items at the location are moved.
Actions triggered by the handling of a moved item are:	
If	Not handled along the route
Then	The item moves and a message is logged warning the user.
If	Not allowed in the destination region
Then	The item is deleted and a message is logged warning the user.

For point values that begin with a \$

If the point value begins with a \$:

- The remainder of the point value is used to identify the item type to move.
- The transition type code becomes a **Type Detector** at this point.

- Tracker searches for the given item type and moves the item to the destination.

Again, in this case:

Actions triggered by what Tracker finds in the entire model are:	
If	The source region is a disperse region.
Then	Only the item with the given item type is moved.
If	Multiple items exist at a location and the source region is not a disperse region.
Then	All items at the location are moved.
Actions triggered by the handling of a moved item are:	
If	Not handled along the route
Then	The item moves and A message is logged warning the user.
If	Not allowed in the destination region
Then	The item is deleted and A message is logged warning the user.

Reference Identifier Action Guidelines

For all point values except values that begin with a \$

Tracker searches for the item with a given reference ID in the entire model.

Actions triggered by what Tracker finds in the entire model are:	
If	The item is not found.
Then	A message is logged.
If	The source region is a disperse region.
Then	Only the given item is moved.
If	Multiple items exist at a location and the source region is not a disperse region.
Then	All items at the location are moved.
Actions triggered by the handling of a moved item are:	
If	Not handled along the route
Then	The item moves and A message is logged warning the user.
If	Not allowed in the destination region
Then	The item is deleted and A message is logged warning the user.

For point values that begin with a \$

If the point value begins with a \$:

- The remainder of the point value is considered the item type to move.
- The transition type code becomes a **Type Detector** at this point.
- Tracker searches for the item with the given item type and moves it to the destination.

Actions triggered by what Tracker finds in the entire model are:	
If	The source region is a disperse region.
Then	Only the given item is moved.
If	Multiple items exist at a location and the source region is not a disperse region.
Then	All items at the location are moved.
Actions triggered by the handling of a moved item are:	
If	Not handled along the route
Then	The item moves and A message is logged warning the user.
If	Not allowed in the destination region
Then	The item is deleted and A message is logged warning the user.

Reference ID Associated to Item Actions: Guidelines

Tracker checks the first location of the source region for items.

Tracker moves all serialized items in the location to the destination region gives each serialized item a reference ID of the point.

! **Important:** Reference ID's are unique. Therefore, if multiple serialized items are moved the last serialized item will exist at the destination region with all previous items in that region being overwritten.

Non serialized items do not have reference ID's so they are moved and not affected by the reference ID.

Actions triggered by what serialized items exist are:	
If	No serialized items are in the source region.
Then	A message is logged warning the user.
If	Multiple serialized items exist at a location.
Then	All items are moved.
Actions triggered by the handling of a moved item are:	
If	Not handled along the route
Then	The item moves and A message is logged warning the user.
If	Not allowed in the destination region

	Then	The item is deleted and A message is logged warning the user.
--	------	---

Item Type Associated to Item Action: Guidelines

Tracker searches the source region for the item type **\$UNKNOWN** .

If **\$UNKNOWN** is found the item type is changed to what the user entered.

Actions triggered by the result of the search for \$UNKNOWN are:		
If	No items with the \$UNKNOWN item type are in the source region.	
Then	A message is logged warning the user.	
If	The source region is a disperse region.	
Then	Only the item with the \$UNKNOWN item type will move.	
If:	Multiple items exist at a location and the source region is not a disperse region.	
Actions triggered by the handling of a moved item are:		
	If	Not handled along the route
	Then	The item moves and a message is logged warning the user.
	If	Not allowed in the destination region
	Then	The item is deleted and a message is logged warning the user.

Positive Edge Trigger Action: Guidelines

Tracker checks the first location of the source region for items.

Actions triggered by what items exist are:		
If	No items are in the source region.	
Then	A message is logged warning the user.	
If	Multiple items exist at a location.	
Then	All items are moved.	
Actions triggered by the specified handling of a moved item are:		
	If	Not handled along the route
	Then	The item moves and A message is logged warning the user.
	If	Not allowed in the destination region
	Then	The item is deleted and A message is logged warning the user.

 **guide: Guidelines for Positive Edge Trigger** include:

1. The transition point may be an integer or character string point.
2. The item transitions when the point goes from low (integer – 0, character string – null string) to high (integer – non-zero, character string – non null string)
3. The first 36 characters of a setpoint, if configured, are:

First 16 characters	Item type
Next 20 characters	Either the: <ul style="list-style-type: none"> • Item ID–for a serialized item • Associated or parent item ID–for a non-serialized item.

Negative Edge Trigger Action: Guidelines

Tracker checks the first location of the source region for items.

Actions triggered by what items exist are:	
If	No items exist in the source region.
Then	No items are moved.
If	Multiple items exist at a location.
Then	All items are moved.
Actions triggered by the specified handling of a moved item are:	
If	Not handled along the route.
Then	The item moves and a message is logged warning the user.
If	Not allowed in the destination region.
Then	The item is deleted and a message is logged warning the user.

guide: Guidelines for Negative Edge Trigger include:

1. The transition point may be an integer or character string point.
2. The item transitions when the point goes from high(integer – non-zero, character string – non null string) to low (integer – 0, character string – null string)
3. The first 36 characters of a setpoint, if configured, are:

First 16 characters	Item type
Next 20 characters	Either the: <ul style="list-style-type: none"> • Item ID–for a serialized item • Associated or parent item ID–for a non-serialized item.

Transition Types: Guidelines

When a destination region has an item enter the region, PRT will ask the source region about the item that just transitioned.

If the source region:

Is managed by another PRT process	That process will be requested for information about that item.
Has no information about the Item	<p>PRT will request the information from an external process. These requests will be continuously made each time the Item moves, until a response is available.</p> <ul style="list-style-type: none"> • The external process is defined to PRT through configuration data. • The configuration of an external process is optional. • PRT will only request an external process to provide information on serialized items.

8. Groups: Defined

Groups organize your regions by physical location or by function, and serve to simplify and expedite the search mechanism in the database.

A Tracking Region must be a member of at least one group. All regions in a tracking group must be configured for the same PRT process service ID.

Groups can hold multiple regions that can be organized by:

Example 1 (page 154)	Function.
Example 2 (page 154)	Physical location.

Example 1. Functional Group

1	Welder lines 1 and 2 are grouped together by function into a Welder group.
2	Robot regions 1 and 2 are grouped together by function into a Robot group.

Example 2. Physical Group

1	The regions in this physical area have been grouped into Group 1.
---	---

2	The regions in this physical area have been grouped into Group 2.
---	---

9. PRT Services: Defined

PRT Services are at the highest level of the PRT database hierarchy. PRT Services manage the configured groups and regions.

PRT Services:

- Manage groups and regions.
- Allow configuration of one part of the plant, while other parts are in production mode.
- Simplify and expedite database searches.
- Can run simultaneously with CIMPLICITY and other Services.
- Provide a snapshot of regions during production.

In PRT Service configuration, the External Process and External Hold Process cannot be the same process.

Example

The following diagram shows three PRT Services configured for a set of groups and regions. All three of the example services can run simultaneously providing an overall view of factory production.

In the event one part of the factory requires configuration, one of the services can be taken down to make the necessary adjustments. The other two services can continue to run.

Note that the services reside at the highest level of the hierarchy and present another method for streamlining searches in the PRT database.

1	CIMPLICITY server with the Tracking system.
2	Regions
3	Groups
4	Services

10. PRT and CIMPLICITY Software Interface Components

PRT interfaces with CIMPLICITY processes including:

- Point management.
- Alarm management.
- Application processes.
- CimEdit/CimView user interface.

Point Management Process and PRT Process

PRT interfaces with CIMPLICITY's Point Management module in order to receive CIMPLICITY Point data values. PRT is capable of receiving Point data from multiple Point Management processes.

Configuration data specifies the information Production Tracking will provide to Point Management.

The interface with Point Management:

- Transfers information on Items entering and exiting Tracking Regions
- Provides Point Management with information on Tracking Region status
- Calculates the quantity of Items in a Tracking Region and the quantity of Items by type in a Tracking Region.

Alarm Management Process to PRT Process

PRT interfaces with Alarm Management for notification of alarm conditions. PRT can be configured to generate alarms when exception conditions are encountered as Items are tracked. Users are notified of CIMPLICITY alarms based on their assigned role and their view of configured Resources.

Application Process to PRT Process

PRT accepts unsolicited messages from application processes requesting Item and Tracking Region data. Messages are also accepted requesting Item tracking data and Tracking Region data modifications. Application processes can send messages to PRT to initiate production start for serialized and non-serialized Items. Application processes will communicate to PRT through a limited set of PRT application functions used to send the requests.

User Interface (CimEdit/CimView) to PRT Process

Your tracking system can be viewed graphically using the CIMPLICITY CimView module. Custom displays and scripts can be created to show a detailed view of your production process. Also provided with Tracker is the PRT User Interface, which provides a view of all the regions and corresponding data. You can perform a wide variety of functions from the User Interface giving you increased control over your tracking model.

Production Tracking Model Layout Planning

Production Tracking Model Layout Planning

Designing your Tracking Model can be an involved process, especially if your factory is large and there are multiple production routes.

Therefore, by far the most efficient way to implement Tracker is to lay out detailed designs before you actually begin configuration.

Planning your Tracking Model involves:

Step 1 (page 157)	Draw a picture of the production process.
Step 2 (page 158)	Isolate regions.
Step 3 (page 158)	Name regions.
Step 4 (page 159)	Define routes.
Step 5 (page 160)	Isolate groups.
Step 6 (page 160)	Name groups.
Step 7 (page 161)	Define item types.
Step 8 (page 161)	Define item classes.
Step 9 (page 162)	Plan display points.
Step 10.	Make revisions until the layout accurately depicts the process.

Step 1. Draw a Picture of the Production Process: PRT Planning

Start designing your Tracking Model by drawing a simple picture of the production process to be tracked. In order to make a complete representation, be sure to include all of the following in your illustration:

- All possible areas and paths through which the product can flow; ensure there are no gaps to prevent data leaks.
- Operator interfaces and devices that indicate a transition of items from one area to another.
- Detainment areas or regions where product is held or scrapped.

Example of a simple manufacturing process

Included are:

- All possible production paths.
- Production flow, indicated with arrows.
- Operator interfaces and devices where transitions take place.

[\(page 158\)](#)

1	Regions.
2	Operator interface stations.
3	Sensors.
4	Devices where transitions take place.
5	Product flow by using direction arrows.
6	All possible paths in the production process.
7	Detainment region for scrapped goods.

Step 2. Isolate Regions: PRT Planning

Using your diagram, draw rectangles around logical areas along the production route to identify your tracking regions.

Regions must:

- Be contiguous.
- Cover every physical part of the production process.
- Not have any physical gaps to prevent lost or misrepresented data by the tracking model.
- Have a data collection device or operator interface at its boundaries to inform the tracking model when a product or carrier passes.

1	One of 13 regions.
2	Each region is bounded by a devices as noted on the diagram.
3	There are no physical gaps between regions.
4	Even the detainment (Scrap) region is marked as a region.

Step 3. Name Regions: PRT Planning

When naming your regions, the most important thing to strive for is consistency. It helps to devise a naming convention that makes sense for your organization, and then write in the names on your

diagram. By coming up with a standardized way of naming your regions, you prevent duplications in your database, and you can easily make additions to your Tracking Model in the future, if needed.

Some ideas for naming conventions are:

- Use the Workstation ID that marks either the entry or exit point of the region.
- Use the device ID that marks either the entry **OR** exit point of the region.
- Base names on the functionality of the region, e.g. Assembly Line1, Oven 2, Robot 3, etc.
- Base names on the location in the factory.
- Limit names to a specific number of characters, for example 10 characters maximum.

Example

1	All names are consistent with a chosen naming convention
2	Each region is named according to function
3	Each name has a maximum of 10 characters, which in the line with the example's chosen naming convention.

Step 4. Define Routes: PRT Planning

Once you have isolated and named the tracking regions on your diagram, you must define all the possible routes through which products can flow.

Each route will subsequently be associated with a CIMPPLICITY point that will send a signal to the tracking model when an item has passed down a specific path.

For this function, it is recommend that you use positive ID switches; however, existing hardware or special production conditions may require that a limit switch be used instead. An operator interface or some other data collection device will normally set the point associated with a route.

Using your tracking region diagram, indicate each and every possible route within the production process. Be sure to include routes to scrap or detainment regions as well.

Example

All routes are defined on the tracking region diagram using direction arrows.

Every possible route is indicated, including the area for scrapped goods. Multiple paths are clearly illustrated on the diagram.

1	Production flow stems from operator interfaces and other devices
2	Direction arrows to show product flow and define routes.
3	Multiple routes from one device.
4	Every possible route, including detainment or scrap regions.

Step 5. Isolate Groups: PRT Planning

When your tracking model is up and running, you will want to perform searches in the database for information on regions and items.

Tracker enables you to simplify searches by organizing your regions into groups. Depending on how you want to organize information in the database, you can group regions by function or by physical location. You may have one region in a group, or several regions in a group, but every region must belong to a group.

Once you have determined how you want to organize information, use your tracking region diagram to form groups.

Example

For this tracking model regions are grouped by physical location, except for the Scrap region, which is by function. Each group has at least one associated region, with most having two or more associated regions. Every region belongs to only one group.

1	One of the 5 groups in the model.
2	This group contains multiple regions and is grouped by physical location.
3	This group contains one region and is grouped by function.

Step 6. Name Groups: PRT Planning

Ideas for naming conventions include:

- Base names on the physical location of the group of regions.
- Base names on the functionality of the group of regions, e.g. Oven Group or Assembly Group.
- Limit names to a specific number of characters, for example 10 characters maximum.

Example

The naming convention in this example:

- Limits names to 10 characters each and
- Bases them on the location of the group.

1	All names are consistent with the chosen naming convention.
2	Each group is named according to the location in the factory
3	Each name has a maximum of 10 characters, which is in line with the example's naming convention.

Step 7. Define Item Types: PRT Planning

Item types are a logical way to identify items with similar characteristics. You must decide what kinds of items you are tracking through production and define item types that will identify them together into logical subsets.

Keep in mind all of the following when determining categories for your items.

- The level of detail in terms of parts/assemblies that you plan to track.
- The parts/assemblies that must be serialized to support positive identification.
- The parts/assemblies that will be non-serialized items.
- A standardized naming convention that is consistent for your organization.

Example

An appliance factory manufactures refrigerators, washers and dryers. The categories are very broad because many of the parts, such as nuts and bolts, are used in all three of the main products. The following chart illustrates the item type definitions.

1	Item types for the refrigerator
2	Item types for the dryer
3	Item types for the washing machine
4	Item types used for all three products and for articles that are reused during production.

Step 8. Define Item Classes: PRT Planning

Item classes are a broader category to include one or more item types, and provide another level in the database for simplifying and expediting searches.

Item classes are a way of further organizing your data into larger groups. Think on a broader level when defining your item classes. Use a standardized naming convention that is consistent for your organization. Item types do not have to belong to an item class but it is recommended.

! **Important:** Although item types do not have to belong to an item class, it is strongly recommended that you create item classes. Feel free to create just one item class to which all of your item types can belong.

Example

Using the earlier example of the appliance factory, four broad categories have been defined for the item classes. The following chart provides an example of the item class definitions.

[\(page 162\)](#)

1	Item class for the refrigerator's item types.
2	Item class for the dryer's item types.
3	Item class for the washing machine's item types
4	Item class for all other item types.

Step 9. Plan Display Points: PRT Planning

Step 9. Plan Display Points: PRT Planning

In order to plan display points, you must think about what data you want to collect from the tracking model and how you want to present it in CimView displays and scripts.

Data can be displayed on screen at a main operating station and at workstations on the plant floor to provide operators with runtime information about the product during production.

[CIMPLICITY setpoints \(page 162\)](#) can be configured to enhance your Tracking Model. There are numerous uses for setpoints.

Display points

Keep in mind all of the following when planning your display points.

- The kind of information needed at workstations on the plant floor and at main interface stations.
- The type of display needed for information: graphic or text representation.
- When to generate alarms when certain production conditions exist.
- The placement of detainment regions, e.g. shifting regions may need user-configured detainment regions.
- How to standardize your displays with consistent features. Keep in mind all of the following:
 - Function buttons reside in the same place on each screen
 - Color scheme
 - Font
 - Language for column heads, buttons, products, machinery, etc.
 - Screen size and geometry.

Display point examples

CIMPLICITY setpoints

CIMPLICITY setpoints can be configured to enhance your Tracking Model. There are numerous uses for setpoints.

You can configure setpoints to:

- Specify the status of the Region.

- Indicate the quantity of Items in the Region.
- Calculate the quantity of Items in the Region by Item Type.
- Update when an Item is automatically moved into a Region (Point-based move).
- Inform when an Item passes a Region transition indicator, to include the Item ID for serialized Items and the Item type for non-serialized Items.

Display Point Examples: PRT Planning

Here is an example of a point set to display at an operator's workstation in the appliance factory. When a refrigerator comes down the line on the conveyor belt, the operator reads the CimView screen that tells him if an electrical unit is required. This particular display uses text only.

Example 1 (page 163)	A point set to display at an operator workstation in production.
Example 2 (page 163)	A point is set to display graphically at an operator's workstation.
Example 3 (page 163)	Display point list.

Example 1. A point set to display at an operator workstation in production.

1	Sensor
2	Point set to display at an operator's workstation.
3	When the value is 0, the refrigerator needs an electrical unit added.

Example 2. A point is set to display graphically at an operator's workstation.

In this example, a Quality Control station graphically displays point values. As the dryers move through the station, the operator reads the interface that is color-coded to provide information during inspection. Here, the graphic display helps the operator spot problems quickly while inspecting the unit.

1	Sensor
2	Point set to display at an operator's workstation.
3	Values are graphically represented in a color chart for each item on the Quality Control list.

Example 3. Display point list.

Listed below is a sample of the factory display points.

Workstation ID	Tracking Information to be Displayed	Graphic or Text
----------------	--------------------------------------	-----------------

Main 01	Plant floor	G	Entire view of factory
Main 02	Plant floor	G	Regions by service
WK 01	Electric Unit installed Y/N	T	
WK 02	Coil installed Y/N	T	
WK 03	Shelving Type	G	Color chart
QC 01	Installed components	G	Color chart
QC 02	Final Inspection	G	Color chart
		T	Warnings
Shipping	Send to warehouse or loading	G	Graphic representation

Production Tracking Configuration

Production Tracking Configuration

It is strongly recommended that you design a tracking model that meets your company's requirements before you configure the PRT database using the Tracker Configuration User Interface (TrackerCfg_UI).

You can easily configure your database using the Tracking Model Wizard that walks you through each step of the configuration process. Properties for your regions, groups, items and the rest are all set using familiar dialog boxes in one contained structure.

You can also open any of the configuration dialog boxes at any time through the [PRT tree \(page 167\)](#).

Steps that describe PRT configuration using the Tracking Model Wizard include:

Step 1 (page 165)	Enable the Tracker Configuration User Interface for PRT.
Step 2 (page 168)	Start the Tracking Model Wizard.
Step 3 (page 168)	Configure regions.
Step 4 (page 173)	Configure routes.

Step 5 (page 176)	Configure item types. These items may: <ul style="list-style-type: none"> • Be Individual items. • Combine to a single location (Combine Items Tracking Region). • Disperse to reside at individual physical locations (Disperse Items Tracking Region). • Be associated items (both physically and logically connected).
Step 6 (page 177)	Configure PRT Services.
Step 7 (page 180)	Configure system definitions.
Step 8 (page 181)	Finish basic Tracking Model configuration.
Step 9 (page 181)	Do additional Tracking configuration.
Step 10 (page 197)	Use Tracker configuration tools.

! **Important:** A source PRT project must have a remote project login configured for the destination project through the prt_client.

i **Tip:** You can exit the Wizard anytime and pick up where you left off later. Upon exiting the Wizard you may receive a message box asking if you want points configured automatically for you. Clicking **Yes** will save you the time and effort of configuring a number of points for which default values will be provided.

☰ **Note:** The Tracker Configuration User Interface is a new feature developed to better help you configure and maintain your tracking data. However, for reference purposes the .idt files that some clients are accustomed to using are outlined in the topic Identify the Configuration Files.

Step 1. Enable the Tracker Configuration User Interface for PRT

Step 1. Enable the Tracker Configuration User Interface for PRT

! **Important:** You must have licensed and enabled the Tracker option to your project to enable access to the Tracker Configuration User Interface.

Step 1.1 (page 166)	Open the CIMPLICITY Tracker Configuration User Interface.
Step 1.2 (page 166)	Select a data source.

Step 1.3 (page 167)	View the Tracker Configuration User Interface: PRT Section.
Step 1.4 (page 167)	View the Tracker Configuration Tree: PRT Section.

Step 1.1. Open the CIMPLICITY Tracker Configuration User Interface

1. Select **Project>Tracker Configuration** in the Workbench left pane.
2. Select **Tracker Configuration** in the Workbench right pane.
3. Do one of the following.

A	Click Edit>Properties on the Workbench menu bar.	
B	Click the Properties button on the Workbench toolbar.	
C	In the Workbench left pane:	
	Either	Or
	Double-click Tracker Configuration .	a. Right-click Tracker Configuration . b. Select Properties on the Popup menu.
D	In the Workbench right pane:	
	Either	Or
	Double-click Tracker Configuration .	a. Right-click Tracker Configuration . b. Select Properties on the Popup menu.
E	Press Alt+Enter on the keyboard.	

4. Right-click **Tracker Configuration**.
5. Select Properties on the Popup menu.
6. Right-click **Tracker Configuration**.
7. Select Properties on the Popup menu.

Step 1.2. Select a Data Source

The first time you open the Tracker Configuration User Interface, the Options dialog box displays so you can set parameters for the data source to be used with the PRT database.

Associated configuration data is stored in a SQL Server database.


Fill in the fields as follows.

Field	Description
Datasource	Selected from the drop down list, must be defined as the system data source.
Login ID	ID that has access to the SQL Server data source.
Password	Entered with the Login ID for access to the SQL Server data source.
Drag and Drop	Checked enables drag and drop functionality.

Result: The data source is designated. All required tables are created in the database as specified by the designated data source. The Tracker Configuration User Interface displays.

 **Important:**

1. Click **ODBC Admin** to run the standard ODBC 32-bit administrator tool for ODBC configuration. Any change to the designated data source must be changed through the Tracker Configuration User Interface.
2. The RCO configuration database should be on the same node as the RCO project. If SQL Server is not on the same node, then MSDE (installed with base CIMPLICITY) should be used. If the RCO config DB is not accessible, then the RCO site cannot run.

 **Tip:** You can access the Options dialog box from the View menu.

Step 1.3. View the Tracker Configuration User Interface: PRT Section

The Tracker Configuration User Interface opens after the data source has been initially designated. All of the tools necessary for configuring items, regions and associated components are provided within the integrated structure. A Tracking Model Wizard is also available so you can quickly configure your tracking model in one session.

Once your Tracking Model is up and running, editing any of its components is easily accomplished using the menu bar options, toolbar buttons and popup menus. The integrated structure holds all of your PRT and RCO files in the left pane directory, and the related configuration dialogs in the right pane. This enables you to quickly edit configuration data for all of your Tracker files.

Step 1.4. View the Tracker Configuration Tree: PRT Section

The PRT directory is organized in a hierarchy with the Tracking Model Wizard in the Wizards folder, and PRT data in the Advanced folder. Each category is clearly defined in its own subfolder within the Advanced folder.

Click on a category to activate the associated dialog in the left pane.

1	RCO configuration folder.
---	---------------------------

2	PRT configuration folder.
3	Wizards folder
4	Tracking Model Wizard
5	Advanced PRT configuration
6	Tracking Model items

Step 2. Start the Tracking Model Wizard

1. Expand the PRT folder in the Tracker Configuration User Interface.
2. Expand the Wizards folder.
3. Double-click Tracking Model folder.

The Region dialog box displays to start the Tracking Model configuration.

Step 3. Configure Regions

Step 3. Configure Regions

- Tracking Model Wizard.
- Region configuration steps.

Tracking Model Wizard

Note: The Tracking Model Wizard facilitates setting up a Tracker project. You can also open the PRT Region Configuration dialog box by clicking **Advanced > Regions** in the TrackerCfg_UI left pane.

Once the Tracking Model Wizard is opened, configure regions in your Tracking Model using the PRT Region Configuration dialog box.

You must configure at least two regions to be able to continue to the next dialog box in the wizard.

1. Click to the right of the **Current Region** field in the Wizard's Region dialog box.
2. Select New.

The PRT Region Configuration dialog box opens.

Region configuration steps

Note: Use these steps whether or not you use the Tracking Model Wizard.

Step 3.1 <i>(page 169)</i>	Set general region parameters.
Step 3.2 <i>(page 169)</i>	Set advanced region parameters.
Step 3.3 <i>(page 170)</i>	Set out of sequence parameters.
Step 3.4 <i>(page 171)</i>	Set alarming options.
Step 3.5 <i>(page 172)</i>	Set logging options.

Step 3.1. Set General Region Parameters

1. Click to the right of the **Current Region** field in the Region dialog box.
2. Select **New** from the popup menu.

The PRT Region Configuration dialog box opens displaying the Region tab.

3. Begin configuring a region by filling in the following fields at the top of the PRT Region Configuration dialog box.

Field	Enter
ID	A unique name for the region Note: Tracker fills the Prefix field automatically as you type in the region ID.
Description	A brief description to help users identify the region.
Locations	The maximum number of available locations that are allowed in the region.
Items per Location	The maximum number of items that can reside at a single region location. Include serialized and non-serialized items
Group	Name for the group with which this region will be associated in the field. Note: If you do not have any groups already configured, it is recommended that you click at the right of the Group field to create a new group (page 191) in order to run the PRT User Interface after completing the Tracking Model wizard. You will also be able to create a group by double-clicking the Group icon in the TrackerCfg_Ui PRT tree.
Resource	Enter the CIMPLICITY resource with which alarms will be generated at the tracking region level in the field. Note: If a resource is not designated, you can click at the right of the Resource field to create a new resource. If you choose not to the CIMPLICITY \$SYSTEM resource will be used by default for alarm generation.

Step 3.2. Set Advanced Region Parameters


1. Click **Advanced**.

Advanced fields display in the middle of the PRT Region Configuration dialog box.

Note: Default points are automatically filled in based on region ID.

2. Configure the advanced fields, as follows.

Field	Description
Label	Enter text in the field that will display with the region and provide user recognition.
Service	Enter a name for the PRT Service that will manage this region. Note: PRT_DC is the default.
Tracking Type	Select a (primary region type) as follows: <ul style="list-style-type: none"> • Pool • Sequential • Detainment • Shifting
Region Type	Select a (secondary region type) as follows: <ul style="list-style-type: none"> • Normal • Combine • Disperse
Lockable	Check to lock the region. Guidelines: In order to make modifications to the region, and to be able to move items into the region, it must be locked. By checking the Lockable box, you enable this functionality.
Auto Associate	Check to enable parent-child association between a serialized and non-serialized item. Note: Must be a Combine region type.
Status Point	Enter the point ID in the field that will represent the region's status. Point must be a UDINT type. Note: A default point ID is inserted in the next 3 fields to help standardize point IDs in your database.
Auto Move Point	Enter the point ID in the field that will display when an item moves into the region due to a point-based move. Point must be a text point of no more than 55 characters. The point should contain the item ID and other optional information depending on the value of the num_setpt_param in the global parameter files.
Item Qty Point	Enter the point ID in the field that will display the total number of items contained in the region. Point must be of type INT or UINT.
Transition Region	(For a shifting region) Enter an ID for the Transition region. Note: When the items shift locations the item at the head location is moved to the shifting transition region specified here.

 **Tip:** The Status, Auto Move and Item Qty points will be automatically configured for you using default values upon exiting the wizard. Click **Yes** in the message box for automatic point configuration to occur.

Step 3.3. Set Out of Sequence Parameters

1. Click **Out of Seq**.

Out of Seq fields display at the bottom of the advanced PRT Region Configuration dialog box.

2. Configure the **Out of Sequence** fields, as follows.

Field	Description	
Detainment Region	ID for the region to which items will be sent when they exceed the out of sequence limit.	
OOS cycle	Maximum number of items that are allowed to exit out of sequence from the region.	
OOS Action	Does one of the following:	
	<ul style="list-style-type: none"> Selects the action to be taken when an item is out of sequence and exceeds the out of sequence limit. 	
	Detain	Moves item to detainment region.
	Scrap	Scraps (delete) item.
	Delay Detain	Allows item to be in out-of-sequence condition using the OOS cycle count, and then move to the detainment region.
	Delay Scrap	Allows item to be in out-of-sequence condition using the OOS cycle count, and then scrap (delete).
	Important: When the OOS cycle field is 0 with any the above options there is no tolerance in the region for out-of-sequence items.	
	<ul style="list-style-type: none"> Disables out-of-sequence. 	
	Blank (no entry)	Disable out-of-sequence. Used if the sequence of items entering a region is not important.
	Note: The OOS cycle field should be 0 for this option.	

 **Note:**


- A Tracking Region may have one and only one Detainment Region.
- Multiple regions may utilize the same Detainment Region.
- When moving an item in either a pooled or sequential region, if it is not the first item in that region, all the items in prior locations are deleted.

This is because an action must be chosen.

Step 3.4. Set Alarming Options

1. Select the Alarming tab in the PRT Configuration dialog box.
2. Check **Enable Alarms** to activate Alarm Switches.
3. Check boxes in the Alarm Switches group to enable alarming as needed for your Tracking Model:

Alarm Switch	Enables alarm generation when...
Alarm Detain	An item(s) is moved to the detainment region.
Alarm Region Not Empty	A Disperse region is not empty and new items have arrived.
Alarm Region Locked	An item enters an 'in-locked' region or exits an 'out-locked' region.
Alarm Location Capacity	The number of items at a single region location exceeds
Exceeded	The configured maximum.
Alarm Item Hold	An Item enters a region with an attribute that matches an item-hold specification.
Alarm Unknown Item	Information cannot be retrieved for a serialized item.
Alarm OOS Condition	An 'out-of-sequence' condition occurs.
Alarm Invalid Item Type	An invalid item type enters the region.
Alarm Unknown Item Type	An unknown item type enters the region.
Alarm Region Capacity	The region capacity has exceeded its configured maximum.

 **Note:** : You can also set alarming and logging options in the RCO_UI.

In the RCO_UI, alarming and logging are affected dynamically, but are not stored in the database.

Step 3.5. Set Logging Options

1. Click the Logging tab in the PRT Configuration dialog box.
2. Check **Master Logging Enable** to activate logging options.
3. In the logging options group, check boxes to enable logging as needed for your Tracking Model:

Logging Option	Enables logging when...
Log Detainment	Items are moved to the detainment region.
Log Region Not Empty	A Disperse region is not empty and new items have arrived.
Log Region Status	An Item enters an 'in-locked' region or exits an 'out-locked' region.
Log Capacity Exceeded	The number of items at a single region location exceeds the configured maximum.
Log Item Hold	An Item enters a region with an attribute that matches an item-hold specification.
Log Unknown Item	Information cannot be retrieved for a serialized item.
Log Item Scrapped	An item has been scrapped (deleted) from the region's queue.
Log Item Entry	An item is entered into the region's queue.
Log Item Exit	An item exits the region's queue.

Log Item Modify	An item's tracking data is modified.
Log Production Start	Item tracking starts in this region.
Log Production Stop	Items tracking stops in this region.
Log Invalid Item	An invalid item type enters the region.
Log Unknown Item Type	An unknown item type enters the region.
Log Region Capacity	The region capacity has exceeded its configured
Exceeded	Maximum.

4. Click **OK**.

The region, which is listed in the Region dialog box, is added to the Tracking Model configuration.

5. Repeat the tasks in Step 3 to configure at least two regions.

6. Click **Next** when you have finished configuring regions.

The Route dialog box displays.

In the RCO_UI, alarming and logging are affected dynamically, but are not stored in the database.


Be aware that you can also set alarming and logging options in the RCO_UI.

Step 4. Configure Routes

Step 4. Configure Routes

Tracking Model Wizard

Note: The Tracking Model Wizard facilitates setting up a Tracker project. You can also open the PRT Route Configuration dialog box by clicking **Advanced**> **Routes** in the TrackerCfg_UI left pane.

 **Important:** Make sure you have configured at least two regions before you begin to configure routes.

1. Click to the right of the **Current Route** field in the Wizard's Route dialog box.

2. Select New.

The PRT Region Configuration dialog box opens.

Route configuration steps

Step 4.1 (page 174)	Set general route parameters.
Step 4.2 (page 174)	Set advanced route parameters.

Step 4.1. Set General Route Parameters

1. Click in the Route dialog box.
2. Select **New** from the popup menu.

The PRT Route Configuration dialog box opens displaying the General tab.

3. Begin configuring a route by filling in the following fields at the top of the PRT Region Configuration dialog box.

Field	Description
Source Region	Region from which the item will exit.
Destination Region	Region that the item will enter.
Translation Type Code	Code, in the drop-down list, that will translate to an internal code. Limit Switch - Transitions any item type. Automatic Item ID giving Item ID - Transitions item type matching value type (Item ID) to text point. Automatic Item ID giving Reference ID - Transitions item type-matching value (Reference ID) to text point. Associated Reference ID to Item - Associates reference ID in setpoint (text) with Item. For example: Item ID = Red, Setpoint = Blue, Item ID = Red; Reference ID = Blue. Associated Item Type to Item - Transitions item type-matching value to an integer point. Uses an item type code configured in PRT Item Type Configuration dialog box. This route type will do the following if there is an item with an \$UNKNOWN item type in the source region. Take an item with an \$UNKNOWN item type from the source region, change the Item Type to the item type input by the user, and move the item to the destination. Otherwise, create an item in the destination region with the Item Type entered by the user. Note: The item type needs to be a non-serialized item type. This route type works like a Type Detector Route type once the Item with an \$UNKNOWN Item Type is changed to the input Item Type. Positive Edge Trigger - Transitions any item type when the setpoint value changes from zero to nonzero. Negative Edge Trigger - Transitions any item type when the setpoint value changes from nonzero to zero.
Transition Point ID	Enter the point ID that represents the transition between the two regions. There are two point types: Limit Switch - Boolean. An item is moved Yes or No. It doesn't matter what item is moved. Positive ID - Text or Numeric. Required when doing sequential tracking. The transition point value must match the value of the Item ID or the item will not be moved.



Step 4.2. Set Advanced Route Parameters

Step 4.2. Set Advanced Route Parameters

1. Click **Advanced**.

Advanced fields display at the bottom of the advanced PRT Route Configuration dialog box.

2. Configure the Advanced fields, as follows.

Field	Enter										
Item Type ID	The name of the item type that will most likely travel between the source and destination regions. Entries and resulting types allowed are as follows.										
	<table border="1"> <tr> <td>Entry in Item Type ID field</td> <td>Types allowed along the route</td> </tr> <tr> <td>Nothing. Field is blank.</td> <td>All item types.</td> </tr> <tr> <td>A specific type.</td> <td>Only the entered type.</td> </tr> <tr> <td>* with other characters</td> <td>Item types that match the pattern. Example A* allows all item types that begin with A.</td> </tr> <tr> <td>* only</td> <td>All item types. This is the same as leaving the field blank.</td> </tr> </table>	Entry in Item Type ID field	Types allowed along the route	Nothing. Field is blank.	All item types.	A specific type.	Only the entered type.	* with other characters	Item types that match the pattern. Example A* allows all item types that begin with A.	* only	All item types. This is the same as leaving the field blank.
Entry in Item Type ID field	Types allowed along the route										
Nothing. Field is blank.	All item types.										
A specific type.	Only the entered type.										
* with other characters	Item types that match the pattern. Example A* allows all item types that begin with A.										
* only	All item types. This is the same as leaving the field blank.										
Setpoint ID	<p>The point ID that will display when the item is transitioned.</p> <p> Note: Guidelines: The point must be:</p> <ul style="list-style-type: none"> • A text point and • No more than 256 characters. <p>The</p> <ul style="list-style-type: none"> • Item Type ID is displayed in the first 16 characters and • Item ID is displayed in the remaining characters (when the item type is serialized; otherwise, associated or parent item ID). 										
Translation ID	<p>An ID that will map to the translation ID in the PRT item Type Translation file. Note: The PRT Item Type translation file translates item type codes received from detection devices into item type IDs that can be recognized by Tracker. This file must be configured when the prt_route records have a translation type value of 2.</p> <p> Note: A route with a Type Detector translation type code requires some additional configuration. (page 175)</p>										
Process first point change	<p>Checked, makes Tracker accept any change, including a null value. Note: By default Tracker does not respond to null values. When a point changes from null to a value Tracker assumes that it was a loss in and a return to communications. If you are using unsolicited communications where a null may be valid; when the point value changes from null to a value you want that to be valid for the route. Process first point change makes the change valid.</p>										

3. Click **OK**.


The route, which is listed in the Route dialog box, is added to the Tracking Model configuration.

4. Click **Next** when you have finished configuring routes.

Configuration for a Route with a Type Detector Translation Type Code

1	Translation Type Code (page 196)	When the value is Type Detector, configure one or more entries in the PRT Type Translation Configuration dialog box.
2	Transition Point ID	Values change when an item type is detected.
3	Translation ID	Route maps to the Translation ID field of the PRT Type Translation Configuration. You can use this ID to group similar item Types together.
4	Input Code	Changes the Transition point to an integer value that matches the Input Code to determine the Item Type to move.
5	Item Type ID	Indicates the item that is to be moved when the input code matches the Transition point value.

Step 5. Configure Item Types


 **Important:** This topic describes item types for Tracker (Base)

Order Execution Mgt. configuration documentation describes item types, groups and attributes.

1 (page 176)	Tracker model wizard.
2 (page 176)	Item type configuration.

Tracking Model Wizard

Note: The Tracking Model Wizard facilitates setting up a Tracker project. You can also open the PRT Item Type Configuration dialog box by clicking **Advanced> Item Types** in the TrackerCfg_UI left pane.

 **Important:** When you create an item type the PRT backing files are re-created, since their configuration has been changed. Therefore, you must delete certain files called [backing files \(page 201\)](#) in order to re-start the Tracker project or re-open the PRT_UI. If you want to retain current information you can [export \(page 201\)](#) data before you delete the files.


1. Click to the right of the Item Type field in the Wizard's Item Types dialog box.

1. Select New.

A PRT Item Type Configuration dialog box opens.

Item Type Configuration

1. Configure the Item Type configuration fields, as follows.

Field	Enter
Item ID	A unique name for the item type.
Description	A brief description to help users identify the item type.
Item Type Code	An integer that will identify this item type.  guide: Guideline: This code is used internally by PRT to identify the Item Type. Each item type must have a unique code, for example COMPRESS = 4 and BASE = 5. You can also apply a unique code for the same item type to identify something unique about the item, for example, BASE = 5 and a painted BASE = 6.
Item Class	Class to which this item type belongs. Reminder: Item Classes are not required but are recommended. Feel free to create one Item Class for all of your item types.
Tracking Type	Click the appropriate radio button as follows: <ul style="list-style-type: none"> • Serialized (page 130) • Non-serialized (page 131)

1. Click **OK**.

The item type is added to your Tracking Model configuration.


1. Click **Next** when you have finished configuring item types.

Result: The Service dialog box displays.

Step 6. Configure PRT Services

By default PRT_DC is already configured as a service.

Usually only one service is configured per Tracking Model, which you are encouraged to use. There is an option for editing the pre-configured service to meet your requirements or you can configure additional services as needed.

 **Important:** A Tracker service name cannot be used for more than one Tracker project on the same node.

If you change the service name it is highly recommended that you [idtpop_selected Tracker service files \(page 318\)](#) to confirm that the name has changed.

1 (page 178)	Tracking model wizard.
------------------------------	------------------------

2 (page 178)	Service configuration.
---	------------------------

1. Tracking Model Wizard

Note: The Tracking Model Wizard facilitates setting up a Tracker project. You can also open the PRT Service Configuration dialog box by clicking **Services** in the TrackerCfg_UI left pane.

1. 1. Click the **Popup Menu** button to the right of the Service field in the Wizard's Service dialog box.

1. Select New.

The PRT Service Configuration dialog box opens.

1. Service Configuration

1. Select the appropriate menu option from the popup menu as follows:

Menu Item	Use to:
New	Create a new service.
Edit	Edit the service listed in the Service ID field.

The PRT Service Configuration dialog box displays.

rect 22, 64, 359, 85 [\(page 179\)](#)
 rect 26, 91, 364, 116 [\(page 179\)](#)
 rect 27, 118, 364, 144 [\(page 179\)](#)
 rect 30, 147, 366, 169 [\(page 179\)](#)
 rect 28, 176, 362, 201 [\(page 179\)](#)
 rect 28, 203, 365, 229 [\(page 179\)](#)
 rect 23, 227, 363, 254 [\(page 179\)](#)
 rect 26, 255, 366, 280 [\(page 179\)](#)
 rect 26, 283, 366, 308 [\(page 179\)](#)
 rect 29, 308, 367, 336 [\(page 179\)](#)
 rect 28, 338, 367, 364 [\(page 179\)](#)
 rect 27, 360, 365, 391 [\(page 179\)](#)
 rect 27, 394, 364, 419 [\(page 179\)](#)
 rect 29, 420, 363, 449 [\(page 179\)](#)
 rect 28, 449, 367, 478 [\(page 179\)](#)

1. Review the PRT service fields as follows.

Field	Description	
Service ID	Enter a new Service ID or use the default.	
	Default	PRT_DC
	A PRT Data Server is automatically created when you create the PRT Service.	
	Default	PRT_DS
Search Order	Enter the order in which this service will be searched for information static to the configured services.	
External item service ID	The service ID of the external process that will provide item tracking data. Note: In almost all cases, this field should be empty.	
External hold service ID	The service ID of the external process that will provide hold information. Note: In almost all cases, this field should be empty.	
Alarms	PRT Alarms	
Detainment region alarm	PRT_DET_REG_FULL	Items are sent to a detainment region.
Region lock alarm	PRT_ITM_THRU_LCK	An item enters an 'in-lock' region or exits an 'out-lock' region due to a point-based move.
Region capability exceeded	PRT_REG_CAP	The region has exceeded its configured maximum.
Location capacity alarm ID	PRT_REG_LOC_CAP	The number of items in a single region location have exceeded the configured maximum.
Sequence alarm ID	PRT_REG_OOS	An item is out of sequence at a region exit transition indicator.
Item hold alarm ID	PRT_ITM_WITH_HLD	An item arrives in a region and is determine to have an item-hold specification.
Invalid item type alarm ID	PRT_INVALID_TYPE	An invalid item type is detected in a region.
Unknown item type alarm ID	PRT_UNKNOWN_TYPE	An unknown item type is detected in a region.
Invalid item ID alarm ID	PRT_INVALID_ITEM	Information for a serialized item cannot be retrieved.
Region not empty alarm ID	PRT_DSP_NOT_MPTY	All items have not left a disperse region and new items have arrived in the region.
Batch que	Not currently used.	

1. Click **OK** to add modifications or a new PRT Service to your Tracking Model configuration.
2. Click **Next** to go to the System Definitions dialog box.

 **Note:** You can also open an existing or new PRT Service Configuration dialog box as follows.

1. Double-click the **Services** icon in the PRT>Advanced folder in the TrackerCfg_UI left pane.
2. Do the following.

Existing service	Double-click the service. The PRT Service Configuration dialog box for that service opens.
New service	Click the New button on the Service toolbar. A new PRT Service Configuration dialog box opens.

Step 7. Configure System Definitions


Tracking Model Wizard

Note: The Tracking Model Wizard facilitates setting up a Tracker project. You can also open the System Definitions dialog box by clicking **System Definitions** in the TrackerCfg_UI left pane.

Once you have configured, or made modifications to your service, proceed to the next dialog box of the wizard to configure the system definitions for your Tracking Model.

System Definitions configuration

The default for each port is a queue size of 10, which is the minimum recommended value.

 **Important:** It is recommended that each of the ports on this dialog box be set to a minimum value of 10.

Enter queue size integers, as follows:

Field	Port Involved to:
Interface Requested Port	Receive interested process updates from the PRT data server(s).
API Synchronous Port	Send requests to PRT data collector(s), and to send requests and receive responses from the PRT data server(s).
API Interested Process Port	Receive requests for item and item-hold data from the PRT data collector(s).
Data Collector Main Port	Collect data.
Data Server Throttle Port	
DS Interested Process Port	

Data Collector Aux Port

Step 8. Finish Basic Tracking Model Configuration

1. Click **Finish** on the Wizard's System Definitions dialog box.
2. Click on the TrackerCfg_UI toolbar to save configuration data.

Tracker adds all of the PRT data that you have entered in the configuration dialogs to the Tracking Model configuration.

3. Expand the Advanced folder.
4. Double-click categories to view configuration data in the left pane.

! **Important:** If you want to move items across project boundaries, you have to do [additional PRT startup configuration \(page 227\)](#) .

Step 9. Do Additional Tracking Configuration

Step 9. Do Additional Tracking Configuration

The Tracking Model Wizard enables you to configure all of the basic components needed to run Tracker. However, there are a number of other configuration options available to you so that you can fine-tune your tracking system and collect the most comprehensive, meaningful data for your operation.

- Basic components.
- Advanced components.

Basic Components

The basic components that are configured upon completion of the Tracking Model wizard are:

Wizard Configuration	
A	Regions
B	Routes
C	Item Types
D	Tracking Groups
E	Services
F	System Definitions

Advanced Components

The remaining configuration options include:

Option 9.1 (page 182)	PRT Graphics display.
Option 9.2 (page 188)	Item type display points.
Option 9.3 (page 189)	Region Attribute counters.
Option 9.4 (page 190)	Tracking groups.
Option 9.5 (page 191)	Group tracking regions.
Option 9.6 (page 191)	Status codes.
Option 9.7 (page 193)	Route invalid types.
Option 9.8 (page 194)	Region item associations.
Option 9.9 (page 195)	Interested processes.
Option 9.10 (page 196)	Type translations.

Option 9.1. PRT Graphics Display

Option 9.1. PRT Graphics Display

The PRT graphic display configuration enables you to perform various counts on region item data or to display item contents into array points.

i Tip: If you compiled a point display list, use it as a guide when configuring your graphics display.

☰ Note: PRT Graphics display ([prtcnt \(page 281\)](#)) will not work for non-serialized items; non-serialized items have no item ID, which is required for the counting process.

Option 9.1.1 (page 183)	Enter PRT Graphics general information.
Option 9.1.2 (page 188)	Enter PRT Graphics global information.

Option 9.1.1. Enter PRT Graphics General Information

1. Open a new PRT Graphic Display Configuration dialog box as follows:
 - a. Expand the PRT folder in the Tracker Configuration left pane.
 - b. Expand the Advanced folder.
 - c. Double-click **Graphics Display**.

A list box displays in the right pane.

- a. Click in the right pane.

The PRT Graphics Display Configuration dialog box opens.

2. Select the General tab.

- rect 0, 53, 285, 80 ([page 184](#))
- rect 1, 336, 286, 371 ([page 188](#))
- rect 1, 242, 286, 338 ([page 187](#))
- rect 1, 216, 286, 243 ([page 187](#))
- rect 1, 194, 286, 218 ([page 186](#))
- rect 1, 172, 286, 196 ([page 185](#))
- rect 1, 152, 286, 173 ([page 185](#))
- rect 1, 127, 286, 154 ([page 185](#))
- rect 1, 102, 286, 129 ([page 184](#))
- rect 1, 78, 286, 105 ([page 184](#))

A (page 184)	Region
------------------------------	--------

B <i>(page 184)</i>	Attribute
C <i>(page 184)</i>	Operator
D <i>(page 185)</i>	Value
E <i>(page 185)</i>	Compare Type
F <i>(page 185)</i>	Display Point
G <i>(page 186)</i>	Element Per Location
H <i>(page 187)</i>	Start Location
I <i>(page 187)</i>	End Location
J <i>(page 188)</i>	Item Class

A	Region
---	--------

Select or create a new [region \(page 168\)](#) using the standard Tracker region controls.

[Up \(page 183\)](#)

B	Attribute
---	-----------

Select the attribute you want to operate on.

[Up \(page 183\)](#)

C	Operator
---	----------

The currently defined operators are:

Count operators	<p>These operators will perform a count of attribute data matching the user specified value.</p> <ul style="list-style-type: none"> • Equals (=) • Not Equals (!=) • Greater Than (>) • Greater Than Equal To (>=) • Less Than (<) • Less Than Equal To (<=) • Less Than Equal To (<=) • Exists (%) • Contains (\$) • Between (@) • Not Between (!@)
Display operators	<p>A special operator that will simply copy the specified attribute for each item in the region to an array point.</p> <ul style="list-style-type: none"> • Display (*=)

[Up \(page 183\)](#)

D	Value
---	-------

Value entries depend upon the selected operator.

Count operators	Count up all attributes that meet the operator defined criteria.
Display operators	The value is not used.

[Up \(page 183\)](#)

E	Compare Type
---	--------------

Compare Type entries depend on the selected operator.

Count operators	The attribute value and the user value will be converted to this type and then the operator comparison will be performed.
Display operators	This field is not used.


[Up \(page 183\)](#)

F	Display Point
---	---------------

Display point entries depend on the selected operator.

Count operators	The Point may be numeric or text. The total of all items in the region matching the operator criteria will be entered into this field.
-----------------	--

Display operators	<p>This must be an array point. The point may be numeric or text. The data will be converted to match the point type. Each element in the array will represent a location in the region. The Item Class ID defined for this graphic display record will control filling of the array.</p> <ul style="list-style-type: none"> • If an Item Class is specified, the array will be filled front to back, starting at the first location, moving back to the last location or until the array limits are exceeded. There will be no holes in the array for empty locations. So if the user specified an item class of XXX and location 1, 2, 3, 5 each has had Item Type class, the array locations 1,2,3,4, will be filled. There will not be a blank location for location 4. • If no Item Class is specified there will be one array location for each region location. If each region can hold 2 items and there are 4 locations in the region then the array would look like this: 		
	Array Offset	Item Location filling	
	0	1	
	1	1	
	2	2	
	3	2	
	4	3	
	4	3	
	5	4	
	6	4	

 **Note:** If a location only contains 1 item, the array will contain an empty element for the missing location.

Guidelines for display points:

Attribute	Point Guideline
PRT_ASSOC_ITEM	Text point that must define a text size of at least 20.
PRT_ITEM_ID	Text point that must define a text size of at least 20.
PRT_ITEM_STATUS	Analog point that must be of type ANALOG_U32.
PRT_ITEM_TYPE_ID	Text point that must define a text size of at least 20.
PRT_REFERENCE_ID	Text point that must define a text size of at least 20.
USER DEFINED	Text point with the number of region locations covered by the point equaling the number of elements in the array. Calculate elements as follows: End location – Start Location +1.

[Up \(page 183\)](#)

G	Element per Location
---	----------------------

The same number as items per location.

The display views an array point that is designed to collect an element that is in a particular location.

For example a region has 10 locations. An array point has 10 elements. It can hold each location in an element. The point can hold 10 locations in the array.

[Up \(page 183\)](#)

H	Start Location
---	----------------

Starting location in the region where counting or displaying is to start.

Locations to send to a point	Enter	
Going in the direction of the Start to End of region		
All locations	Start location	0
	End location	0
Range of specific locations	Start location	Number of the Start location.
	End location	Number of the End location.
All locations after a start location	Start location	Number of the Start location
	End location	0
Example		
Start Location	1	
End Location	10	
Locations sent to a point	Locations 1 - 10.	
Going in the direction of the End to Start of region		
Range of specific locations	Start location	1 (End location in the region)
	End location	< x > Where X is the number of locations to be included going from end to start.

[Up \(page 183\)](#)

I	End Location
---	--------------

Ending location in the region where counting is displaying is to end.

Attribute characters

Start	Starting position in the attribute value to be displayed
-------	--

End	Ending position in the attribute value to be displayed.
-----	---

[Up \(page 183\)](#)

J	Item Class
---	------------

Leave blank for all. The item class to be included.

3. Click **OK**.

Option 9.1.2. Enter PRT Graphics Global Information

Select the Global tab in the PRT Graphic Display Configuration dialog box.

rect 3, 63, 322, 96 [\(page 188\)](#)

rect 3, 94, 322, 127 [\(page 188\)](#)

rect 4, 155, 323, 249 [\(page 188\)](#)

rect 4, 246, 323, 279 [\(page 188\)](#)

Field	Description
Reset	(Required for the prtcnt.exe configuration) Forces the counters to do a global update, when the reset point is updated. A reset point can be any type. When a user changes the reset point value, prtcnt recounts all items. Important: Do not use the Add, Modify or Delete points that you select on this tab as the reset point.
Refresh Rate	The interval between automatic counter updates. Minimum is 500 milliseconds.
Add, Modify and Delete	Interested process update points. These points get updated when a PRT item anywhere in the tracking model is modified, added or deleted. Each point should be a string. The point contents are: <Item id>, <Numeric Event code>, <String Event code>. When there is an add, modify or delete the specified point will be updated with the data, item ID, numeric event code, string event code. Tip: Add the point to the Point Control Panel and view its value when you perform the related action in the PRT_UI.
Status	

2. Click **OK**.

Option 9.2. Item Type Display Points

Item type display points enable you to specify a setpoint that can be used to identify the quantity of a particular item type in a region. The point can be used in graphic screens to supply you with runtime data about the quantity of a particular item at any given time keeping operators abreast of potential problems in a region.

1. Open the PRT Display Attribute Configuration dialog box as follows:
 - a. Expand the PRT folder in the Tracker Configuration left pane.
 - b. Expand the Advanced folder.
 - c. Double-click **Item Type Display Points**.

A list box displays in the right pane.

- a. Click in the right pane.
- b. Select New from the Popup menu.

A new PRT Display Attributes Configuration dialog box opens.

2. Configure the PRT display attribute quantity, as follows:

Field	Description
Region ID	The region in which data is to be collected.
Item Type	The item type (page 137) whose quantity is to be displayed.
Quantity	A setpoint ID that will be used to determine the item type quantity in the region.

3. Click **OK**.

The item display point configuration is added to your Tracking Model.

Option 9.3. Region Attribute Counters

Region attribute counters are used to maintain a count of the number of times an item has been through a region. The counter is tied to a specific attribute that is either increased or decreased by increments as the item re-enters the region. You can keep track of region attribute counters using the PRT User Interface or on a graphic screen. By specifying an alarm value, you can enter a number that will generate an alarm to avoid over-processing an item.

1. Open a new PRT Region Attribute Configuration dialog box as follows:
 - a. Expand the PRT folder in the Tracker Configuration left pane.
 - b. Expand the Advanced folder.
 - c. Double-click **Region Attribute Counters**.

A list box displays in the right pane.

- a. Click in the right pane.

A new PRT Region Attribute Configuration dialog box opens for configuration.

2. Configure the PRT region attribute counters, as follows:

Field	Description
Region	Region in which you want to maintain a count by attribute.
Attribute	Attribute that will have a counter in the region.
Start Byte	First byte of the attribute value that will be used for the counter in the region. Reminder: Attributes have a maximum of 16 bytes.

End Byte	Last byte of the attribute value that will be used for the counter in the region.	
Operation Code	Check one of the following:	
	Increment	Increase the counter by one whenever an item enters the region containing this attribute.
	Decrement	Decrease the counter by one whenever an item enters the region containing this attribute.
	Preset	Value is preset. Note: If using Preset , configure an additional counter to either increase or decrease by increments the value for alarming purposes.
Alarm Value		(Integer) An alarm will be generated when the counter reaches this value.

3. Click **OK**.

The region attribute counter configuration is added to your Tracking Model.

Option 9.4. Tracking Groups

It is recommended that you create your [groups \(page 154\)](#) during the configuration of your regions. You can, however, configure new or additional groups anytime using the PRT Group Configuration dialog box.

 **Important:** Every region in your Tracking Model **must** belong to a group.

1. Open the PRT Group Configuration dialog box as follows:
 - a. Expand the Advanced folder.
 - b. Double-click **Tracking Groups**.

A list box displays in the right pane.

- a. Click in the right pane.

The PRT Group Configuration dialog box opens.

2. Configure a PRT group, as follows:

Field	Description
Group ID	A unique name for the tracking group.
Label	Will display for the user as the group identifier. Note: This label is used in logging for the PRT logging tables.
Description	Displays in the list of groups to help a user identify the group.
Resource ID	The resource can either be selected or created for the group.

3. Click **OK**.

The group configuration is added to your Tracking Model.

Option 9.5. Group Tracking Regions

When a region is configured it must be assigned to a [group \(page 154\)](#), which automatically configures the Group Tracking Region. Configure additional Group Tracking Regions as explained below.

1. Open the PRT Region Group Configuration dialog box, as follows.
 - a. Expand the PRT folder in the Tracker Configuration left pane.
 - b. Expand the Advanced folder.
 - c. Double-click **Group Tracking Regions**.

A list box showing automatically configured group tracking regions displays in the right pane.

- a. Click in the right pane.

The PRT Region Group Configuration dialog box opens.

2. Assign a region to a group as follows:

Field	Description
Group	The tracking group to which the region is assigned.
Region	Region belonging to the group.
Sequence	The order of this region in the group in the field. Example There are 22 regions in a group and physically this is the twentieth region. Enter 20. Sequence numbering starts at 0.

3. Click **OK**.

The group tracking region configuration is added to your Tracking Model.

Option 9.6. Status Codes

Option 9.6. Status Codes

A status code must be assigned for each valid region and item status. For example, valid region statuses involve capacity, sequential error, invalid or unknown item type and so on. Tracker provides you with several predefined status codes for the most common statuses. You may, however, configure additional status codes to meet your criteria.

There are 31 status bits each for regions and items. For every defined status code (either predefined or user-defined), a status bit is used. The status bit is Boolean and is either on (1) or off (0). When

you define your own status codes, you must use the available bits. The lists below define the status code, description and bit used for the 31 status bits.

Option 9.6.1 (page 192)	Pre-defined status codes.
Option 9.6.2 (page 193)	Configure status codes.

Option 9.6.1. Pre-defined Status Codes

1. Expand the PRT folder in the Tracker Configuration left pane.
2. Expand the Advanced folder.
3. Double-click Status Codes.

A list box with predefined status codes displays in the right pane.

Predefined status codes for regions include:

Status Code	Description	Bit Used
CTRLWTRK	Control with track	22
RCAPEXC	Capacity exceeded	21
REGHDVLD	Region head valid	23
REHOLD	External hold	17
RFULL	Region full	16
RINLOCK	Region in lock	18
RNORML	Normal	31
ROUTLOCK	Region out lock	19
ROUTSEQ	Out of sequence error	20
WAIT4DEC	Wait for decision	24

Predefined status codes for items include:

Status Code	Description	Bit Used
DELYD	Delayed	16
EHOLD	External hold	18
HOLD	Internal hold	17
INORML	Normal	31

Reserved status codes include:

Status Code	Description	Bit Used
DETAIN	Immediate move to repair	0
DLY_DTN	Delay move to repair	1
DLY_SCRP	Delay scrap	2
SCRAP	Immediate scrap	3

Option 9.6.2. Configure Status Codes

1. Open the Status Configuration dialog box, as follows:
 - a. Expand the PRT folder in the Tracker Configuration left pane.
 - b. Expand the Advanced folder.
 - c. Double-click Status Codes.

A list box with predefined status codes displays in the right pane.

- a. Click in the right pane.

The Status Configuration dialog box opens.

2. Configure a status code, as follows:

Field	Description	
Status Code	A unique code, in the field, to identify the status of the region or item. The code can be no more than 16 characters.	
Description	Helps a user identify the status code.	
Label	Will display for the user as the status code identifier.	
Bit Set	The bit to be set by PRT to represent this status code.	
Code Type	Region Status	Applies the status code to regions
	Item Status	Applies the status code to items.

3. Click **OK**.

Option 9.7. Route Invalid Types

The route invalid type specifies item types that are not valid in a particular region.

When you configure your routes you have the option of validating all item types by placing an asterisk (*) in the **Item Type** field.

You can also filter out item types. Specify the item types that are invalid for a route and enable alarm generation so that operators can react promptly.

1. Display the Route Invalid Type Configuration dialog box, as follows:
 - a. Expand the PRT folder in the Tracker Configuration left pane.
 - b. Expand the Advanced folder.
 - c. Double-click **Route Invalid Types**.

A list box displays in the left pane.

- a. Click in the right pane.

The Route Invalid Types Configuration dialog box opens.

2. Configure a route invalid type, as follows:

Field	Description
Source Region	Region the item exits.
Destination Region	Region the item will enter.
Item Type	Type that is invalid through this particular route.
Transition Point	Transitions the item along the route.

3. Click **OK**.

The route invalid type configuration is added to your Tracking Model.

Option 9.8. Region Item Associations

In a basic PRT parent-child association, you must identify a serialized and non-serialized item type for the region to identify the region item association.

If there will be one or more items processed in the region that do not require the parent-child association, you must identify each as a bystander using the Region Item Association dialog box.


1. Open the Region Item Association dialog box, as follows.
 - a. Expand the PRT folder in the Tracker Configuration left pane.
 - b. Expand the Advanced folder.
 - c. Double-click **Region Item Associations**.

A list box displays in the left pane.

- a. Click in the right pane.

The Region Item Associations dialog box opens.

2. Configure an item's update type, as follows:

Field	Description	
Region ID	Region in which the parent-child association must be made for tracking purposes.  guide: Guideline: The parent-child association involves a serialized and non-serialized item. When the two are associated in the Tracking system, they become linked through the rest of the production process. For example, if you have two items through which holes are drilled so that they can be attached and the items are separated in production for processing, applying the parent-child association ensures that they remain logically linked in the system.	
Item Type	Serialized item	Enter one of the following. Parent item in the association.
	Non-Serialized item	Child item in the association.
	Bystander	Item that does not require the parent-child association, but will go through this region to be processed. Important: Entries for these items are also required.
Update Type	Check...	When the item type is...
	Parent	Serialized.
	Child	Non-serialized.
	Bystander	Either serialized or non-serialized but is not participating in a parent-child association.

3. Click **OK**.

The region item association configuration is added to your Tracking Model

Option 9.9. Interested Processes

In order to supply interested processes with regular updates from PRT Service, you must map the interested process to the PRT Service.

An interested process receives data from PRT and is updated to provide operators and other users with runtime information. An example is a graphical screen display (CimView), which would be "interested" in receiving data from PRT to display on screen for operators and other users.

1. Open the Interested Process Configuration dialog box, as follows:
 - a. Expand the PRT folder in the Tracker Configuration left pane.
 - b. Expand the Advanced folder.
 - c. Double-click **Interested Processes**.

A list box showing two pre-configured processes displays in the left pane. You may not be required to configure any other interested processes.

- a. Click in the right pane.

The Interested Processes Configuration dialog box opens.

2. Configured an interested process, as follows:

Field	Description	
PRT Service	The service that provides the interested process with updated information in the field. The process that receives data from PRT.	
Interested Process Service		
Update Type	Check...	When you want to be updated with...
	Brief	All manual transactions. No automatic transactions made within Tracker will be reported.
	Full	All manual and automatic transactions.

3. Click **OK**.

The interested process configuration is added to your Tracking Model.

Option 9.10. Type Translations

Type translations are configured when you have one or more routes with a Translation Type Code of Type Detector (internal code = 2). This means that when an item is moving along a route and is detected, the system will check the input code against the point value to determine if the item should be moved to the destination region. Translation types give you another method for maintaining control over the flow of items in your production facility.

1. Open the PRT Type Translation Configuration dialog box, as follows:
 - a. Expand the PRT folder in the Tracker Configuration left pane.
 - b. Expand the Advanced folder.
 - c. Double-click Type Translations.

A list box displays in the left pane.

- a. Click in the right pane.

The PRT Type Translation Configuration dialog box opens.

2. Configure a PRT type translation, as follows:

Field	Description
Translation ID	A unique name for the item type translation. The translating ID basically masks a value. For example, if you have a numeric value and you want to replace that with a text field, enter the text value for the Translation ID.
Input Code	The value will be matched against the value received from the detection equipment that is signaling the item type

Item Type ID	Item type that is to be translated using the input code
--------------	---

3. Click **OK**.

The type translation configuration is added to your Tracking Model

! **Important:** The **Translation ID** field maps the PRT Type Translation Configuration to the PRT Route Configuration. The Translation ID tells PRT what record to look at when an item type is detected in the route.

Step 10. Use Tracker Configuration Tools

Step 10. Use Tracker Configuration Tools

Once you configure all of your Tracking data, you can perform a number of functions within the TrackerCfg_UI. The menu options and toolbar buttons provide you with the flexibility to work with configuration data.

Options include:

Option 10.1 (page 197)	Use the TrackerCfg_UI menu (general)
Option 10.2 (page 200)	Modify PRT configuration.
Option 10.3 (page 202)	Use Browse to search for configuration data.
Option 10.4 (page 203)	Delete configuration data.
Option 10.5 (page 204)	Export Tracker configuration to a file.
Option 10.6 (page 205)	Import Tracker configuration to another project.

Option 10.1. Use the TrackerCfg_UI Menu (General)

Following are the menu items in the TrackerCfg_UI.

Menus are:

- File

- Edit
- Tree
- View
- Tools
- Help

If the feature applies to PRT or RCO, its column is checked.

File	PRT	RCO	Description	
New>Folder	X	X	PRT	Opens a Properties dialog box to create a new item when To enable the Folder menu item: <ul style="list-style-type: none"> • Select a PRT feature in the left pane. • Click in the right pane.
			RCO	Places a new folder at the end of the RCO branch.
New>Routing Control Object		X	Folder or Routing Control Object creates a new RCO.	
View	X	X		Displays an inactive feature that you select in the left pane.
Save	X	X		Saves any unsaved changes.
Rename	X	X	PRT	Opens (page 200) the Properties dialog box for a user created PRT item.
			RCO	In the TrackerCfg_UI left pane, makes a user created RCO feature writable.
Activate		X		Makes an RCO site available in the runtime environment.
Properties		X		Customize RCO folder properties.
Lock		X		Locks one or more RCO's.
Refresh	X	X		Refreshes the TrackerCfg_UI screen.
Export	X	X		Export Tracker configuration to a file.
Import	X	X		Import Tracker configuration from a file.
Exit	X	X		Close the TrackerCfg_UI.

Edit	PRT	RCO	Description	
Cut		X	Cut an entry in the following RCO fields in the TrackerCfg_UI right pane when the RCO is locked for configuration.	
			Feature	Fields
			Routing Control Object	All
			Decision	All
			Routing Logic	Name\Description

Copy		X	Copy an entry in the following RCO fields in the TrackerCfg_UI right pane when the RCO is locked for configuration.
			Feature
			Fields
			Routing Control Object
			All
			Decision
			All
			Routing Logic
			Name\Description
Paste		X	Paste an entry in any of the fields that you can cut or copy when the RCO is locked for configuration.
Delete		X	Delete any of the following user defined features in a locked RCO. <ul style="list-style-type: none"> • Trigger • Decision • Routing

Tree	PRT	RCO	Description	
Expand One Level	X	X	Expands any item in the TrackerCfg_UI tree to one level below its level. Example	
Expand One Branch	X	X	Expands the entire branch below a selected item in the TrackerCfg_UI tree. Example	
Expand All	X	X	Expands the entire TrackerCfg_UI tree.	
Collapse Branch	X	X	Collapses the selected branch in the TrackerCfg_UI tree. Example	
Indicate Expandable Branches	X	X	Checked	Displays + an - to left of tree items. Example
			Clear	Does not display + an - to left of tree items. Example

View	PRT	RCO	Description	
Toolbar	X	X	Checked	Displays the TrackerCfg_UI toolbar
			Clear	Hides the TrackerCfg_UI toolbar.
Status bar	X	X	Checked	Displays the TrackerCfg_UI status bar.
			Clear	Hides the TrackerCfg_UI status bar.
Split	X	X	Automatically positions the cursor between the left and right pane when Split is selected. Example	

Tools	PRT	RCO	Description	
New Point	X	X	Opens the New Point dialog box.	
Dynamic Configuration	X	X	Enables dynamic configuration.	
Attribute Maintenance	X	X	Opens the Attribute Maintenance (page 350) dialog box.	
Region Maintenance		X	Opens the Region Maintenance (page 351) dialog box.	

Error List		X	Opens the Error dialog (page 352) .
Verify		X	Check the RCO configuration data (page 352) .
Clean RCO directory		X	Cleans the RCO directory. This is a maintenance operation. When Clean RCO directory runs it: <ul style="list-style-type: none"> • Compares the records in the <code>rcocotr1loc</code> table of the database against the RCO directory for the current project. • Deletes any folders that exist in the directory that are no longer in the database.
RCO database		X	Opens the Options dialog box to select the RCO database.
TADB Configuration Update	X		(Order Execution Mgt.) Updates the TADB database
TADB Datasource	X		(Order Execution Mgt.) Opens the TADB Datasource Specification dialog box.
View TADB Revisions		X	(Order Execution Mgt.) Opens the Revisions browser.
TADB Historical Database		X	(Order Execution Mgt.) Opens the TADB Datasource Specification dialog box to specify the Historical database (page 838) .
Start/Stop Historical Database Activity		X	(Order Execution Mgt.) Starts or stops (page 840) the Historical Database.

Help	PRT	RCO	Description
Contents	X	X	Opens Tracker documentation first page and displays the Global documentation Table of Contents.
Search for help on	X	X	Opens Tracker documentation first page and displays the Global documentation Search tab.
About Tracker Configuration	X	X	Displays the CIMPLICITY Version and build number.

Option 10.2. Modify PRT Configuration

Option 10.2. Modify PRT Configuration

All of the Tracker configuration data is eligible for modification. If you make changes in your production facility, or you have simply made a mistake, you have the option of modifying your configuration data in its dialog box.

Option 10.2.1 (page 200)	Open an existing PRT Configuration dialog box.
Option 10.2.2 (page 201)	Follow a required procedure to restart the PRT User Interface

Option 10.2.1. Open an Existing PRT Configuration Dialog Box

1. Open an existing configuration dialog box, as follows:

Method 1. Menu option

- a. Expand the Advanced folder in the TrackerCfg_UI directory.
- b. Double-click the category that requires modification.

A list box displays in the right pane.

- a. Select the item in the list to be modified.
- b. Click File on the menu bar.
- c. Select **Rename**.

Method 2. Popup menu

- a. Expand the Advanced folder in the Tracker Configuration UI directory.
- b. Double-click the category that requires modification.

A list box displays in the left pane.

- a. Right-click the item in the list to be modified.
- b. Select **Edit** from the popup menu.

Method 3. Quickest

- a. Expand the Advanced folder in the Tracker Configuration UI directory.
- b. Double-click the category that requires modification.

A list box displays in the left pane.

- a. Double-click the item in the list to be modified.

2. Make the required modifications.

Option 10.2.2. Follow a Required Procedure to Restart the PRT User Interface

Option 10.2.2. Follow a Required Procedure to Restart the PRT User Interface

1. Run the Export and Import programs. This will put all of your item tracking data into an output file that can be imported back once the backing files have been deleted.
2. Delete the backing files.

Option 10.2.2.1. Save Tracking Data and Delete Files

1. Run the [Export \(page 295\)](#) program to save your tracking data to an output file.
2. Open the Tracker project in the Workbench.
3. Click Tools on the menu bar.
4. Select **Command Prompt**.

An MS Dos window opens.

5. On the command line type **cd log** .
6. Press **Enter**.
7. Type **del *.*prt_dc** .
8. Press **Enter**.

Backing files are deleted.

9. Run the [Import \(page 299\)](#) program to preserve tracking data.

Tracking data is preserved, backing files are deleted and the PRT_UI can be accessed to view tracking data.


Option 10.2.2.2. Delete Tracking Data

1. Open the Tracker project in the Workbench.
2. Click Tools on the menu bar.
3. Select Command Prompt.

An MS Dos window opens.

4. On the command line type **cd log** .
5. Press **Enter**.
6. Type **del *.*prt_dc** .
7. Press **Enter**.

Previous tracking data is lost, backing files are deleted and the PRT_UI can be accessed to view tracking data.

 **CAUTION:** If you choose to delete the backing files without running the Export and Import programs, you will lose all of your item tracking data.

Option 10.3. Use Browse to Search for Configuration Data

1. Do one of the following:

Method 1. Popup menu

- a. In the configuration dialog box, click next to the field for which you want to browse for Tracker configuration data.
- b. Select **Browse** from the popup menu.

A Browse list box opens with a list of related data.


Method 2. Quickest

In the configuration dialog box, click the related next to the field for which you want to browse for Tracker configuration data.

A PRT Browse opens with a list of related data.


2. Select an item from the Browse list box.
3. Click **OK**.

The ID for the selected item will display in the field, and/or the dialog box for the selected item will open.

 **Note:** Browse buttons for other CIMPLICITY data, e.g. Points, are also available for relevant data.

Option 10.4. Delete Configuration Data

You have the option of deleting configuration data from the TrackerCfg_UI. Configuration data is not deleted until you save the configuration.

 **CAUTION:** Use care when deleting configuration data. Once deleted this information is no longer available in the PRT_UI.

Method 1. Menu option

1. Expand the Advanced folder in the Tracker Configuration UI directory.
2. Double-click the category that contains the item to be deleted.

A list box displays in the left pane.

3. Select the item in the list to be deleted.
4. Click Edit on the menu bar.
5. Select **Delete**.

Method 2. Popup menu

6. Expand the Advanced folder in the Tracker Configuration UI directory.
7. Double-click the category that contains the item to be deleted.

A list box displays in the left pane.

8. Right-click the item in the list to be modified.
9. Select **Delete** from the popup menu.

Method 2. Quickest

10. Expand the Advanced folder in the Tracker Configuration UI directory.
11. Double-click the category that contains the item to be deleted.

A list box displays in the left pane.

12. Click .

Using any of the above methods, the font will change for the deleted item in the list. Once you save the configuration the item is deleted from the list and the Tracking Model configuration.

Option 10.5. Export Tracker Configuration to a File

1. Do one of the following?

Method 1. Menu option

- a. Expand the PRT folder in the Tracker Configuration UI directory.
- b. Select the Advanced folder.
- c. Click File on the menu bar.
- d. Select **Export**.

The Open dialog box opens.

Method 2. Popup menu

- a. Expand the PRT folder in the Tracker Configuration UI directory.
- b. Right-click on the Advanced folder.
- c. Select **Export** from the popup menu.

The Open dialog box opens.

2. Open the directory where the file is to be stored.
3. Type a name for the file in the **File name** field. The file must have a `.trk` extension.
4. Click **Open**.

The Tracker configuration data is saved to a file in the specified directory.

Option 10.6. Import Tracker Configuration from another Project

1. Do one of the following.

Method 1. Menu option

- a. Expand the PRT folder in the Tracker Configuration UI directory.
- b. Select the Advanced folder.
- c. Click File on the menu bar.
- d. Select **Import**.

The Open dialog box opens.

Method 2. Popup menu

- a. Expand the PRT folder in the Tracker Configuration UI directory.
- b. Right-click on the Advanced folder.
- c. Select **Import** from the Popup menu.

The Open dialog box opens.

2. Navigate to the directory where the Tracker configuration file is to stored.
3. Select the file.
4. Click **Open**.

The Tracker configuration data is imported to the project and can be viewed in the Tracker Configuration User Interface.

Production Tracking User Interface (PRT_UI)

Production Tracking User Interface (PRT_UI)

The Production Tracking User Interface (PRT_UI) enables you to view and manage production tracking data during runtime. The PRT_UI is designed to simplify management of your PRT data with an easy to use graphical interface.

When you use the PRT_UI, you are viewing and working with data that is in the computer and represents what happens in the tracking process.

Step 1 (page 206)	Open the PRT_UI.
Step 2 (page 212)	Display and browse for regions in the PRT_UI.
Step 3 (page 213)	Auto-lock regions.
Step 4 (page 214)	Configure items through the PRT_UI
Step 5 (page 225)	Update a region's status.
Step 6 (page 225)	Connect to multiple projects. Note: This step covers Tracker base and Order Execution Mgt. (page 772) functionality.
Step 7 (page 240)	Manipulate items.
Step 8 (page 246)	Print reports.

Step 1. Open the PRT_UI

Step 1. Open the PRT_UI

After you open the PRT_UI, it will be a good idea to become familiar with the features in the window before you start manipulating data.

You can also create a shortcut for users to easily open the PRT_UI.

Option 1.1 (page 207)	Open the PRT_UI through the Start menu or the Workbench.
Option 1.2 (page 208)	PRT_UI basic sections.
Option 1.3 (page 208)	PRT_UI toolbar buttons.

Option 1.4 (page 209)	Item Cache and Paging Bar
Option 1.5 (page 210)	Create a shortcut to open the PRT_UI.

Option 1.1. Open the PRT UI

1. Make sure the Tracker project is running.
2. Use one of the following.
 - Workbench
 - Start menu

Workbench

- a. Select **Runtime>Production Tracking>Production Tracking UI** in the Workbench left pane.
- b. Select **Production Tracking UI** in the Workbench right pane.
- c. Do one of the following.

1	Click Edit>Properties on the Workbench menu bar.	
2	Click the Properties button on the Workbench toolbar.	
3	In the Workbench left pane:	
	Either	Or
	Double-click Production Tracking UI .	a. Right-click Production Tracking UI . b. Select Properties on the Popup menu.
4	In the Workbench right pane:	
	Either	Or
	Double-click Production Tracking UI .	a. Right-click Production Tracking UI . b. Select Properties on the Popup menu.
5	Press Alt+Enter on the keyboard.	

Start menu

- a. Click Start on the Windows task bar.
 - b. Select (All) Programs>Proficy HMI SCADA - CIMPLICITY version>PRT User Interface.
3. Right-click **Production Tracking UI**.
 4. Select Properties on the Popup menu.
 5. Right-click **Production Tracking UI**.

6. Select Properties on the Popup menu.

Option 1.2. PRT UI Basic Sections

The PRT_UI can be broken into the following broad sections:

rect 0, 18, 168, 38 ([page 208](#))

rect 0, 36, 256, 63 ([page 208](#))

rect 0, 62, 256, 95 ([page 208](#))

rect 0, 62, 256, 95 ([page 208](#))

rect 254, 62, 492, 95 ([page 208](#))

rect 0, 93, 435, 126 ([page 208](#))

rect 5, 149, 491, 274 ([page 208](#))

rect 53, 279, 488, 301 ([page 208](#))

1	PRT_UI menu bar	(Items are described with related actions.)
2	Toolbar (page 208)	Provides one-click access to the most commonly used functions.
3	Region Id field	Displays and enables the user to select the active region in the PRT_UI. The browse button opens the Regions browser.
4	View field	Drop down list of the configured views for the PRT_UI.
		Note: The views are defined in the PRT_WINDOW.cfg file.
5	Item Cache and Paging Bar (page 209)	Allows attribute cache and paging through the PRT_UI.
6	PRT_UI body	Lists the items, with details, in each of the region's locations. The list reflects the status on the factory floor as recorded in the PRT database.
7	Locations:, Items:	On the status bar, displays the number of locations and items in the active region.

Option 1.3. PRT UI Toolbar Buttons

The buttons on the PRT toolbar provide one-click access to the most commonly used functions.

1	Adds a new item location to the selected item.
2	Inserts a new item "in front" of the selected item.
3	Modifies the selected item's properties.
4	Deletes the selected item from inventory.
5	Moves the selected item from the current region to another region.
6	Finds the region where the item is located.
7	Searches the PRT database for an item.

8	Refreshes PRT DV regions.
9	Auto locks the status bits in the region, when configured as such.
10	Disables automatic updates of the region list when a PRT region changes.
	When automatic updates are disabled PRT_UI displays N Messages Missed Where N is the number of missed updates.

Option 1.4. Item Cache and Paging Bar

PRT_UI provides:

- An attribute cache to enhance the response time speed.
- Paging controls to rapidly move through listed items.

rect 1, 1, 196, 54 ([page 209](#))

rect 194, 1, 291, 54 ([page 209](#))

rect 289, 0, 487, 53 ([page 209](#))

1	Go to Location	Finds the first item in a specified location (for the displayed region).	
		<ol style="list-style-type: none"> 1. Enter a location in the Starting Location field. 2. Click the Go button. 	
2	Paging controls	PRT_UI provides four paging control buttons.	
		Buttons select items as follows.	
		First item in the region.	
		Up one display screen full of items.	
		Down one display screen full of items.	
		Last item in the region.	
3	Attribute cache size	Controls the number of items that are fetched from the <code>Trkcollector</code> in each update.	
		Minimum size	100
		Default	100

 **guide: Cache and Paging Guidelines**

- When automatic updates are disabled, all items are downloaded from the `Trkcollector`.

There will be a slight delay.

- Do not scroll down more than one page using the scroll bar; use the page up or page down buttons instead.
- For regions with more than 1000 items, the automatic updates are automatically turned off.

Option 1.5. Create a Shortcut to Open the PRT UI

Option 1.5. Create a Shortcut to Open the PRT UI

Option 1.5.1 (page 210)	Place the PRT_UI shortcut on the Windows desktop.
Option 1.5.2 (page 210)	Set parameters for the PRT User Interface

Option 1.5.1. Place the PRT_UI Shortcut on the Windows Desktop

1. Open Windows Explorer.
2. Expand the Windows folder (e.g. WINNT).
3. Go to the ...\\Proficy CIMPLICITY\\exe directory.
4. Right-click prt_ui.exe.
5. Select Create shortcut from the popup menu.
6. Place the **prt_ui** shortcut icon onto your Windows desktop.

The PRT User Interface shortcut displays on your Windows desktop.

Option 1.5.2. Set Parameters for the PRT UI

1. Right-click the PRT User Interface shortcut on the Windows desktop.
A Popup menu opens.
2. Select Properties.

The PRT User Interface Properties dialog box displays.

3. Select the Shortcut tab.
4. Enter the switch and parameter to customize the PRT User Interface when opened In the **Target** field. Use the following switch options:

Switch	Parameter
<code>=R</code> <code><region name></code>	<code><region name></code> is a valid region name. Displays information for a specific region.
<code>=S</code> <code><prtsystem></code>	<code><prtsystem></code> is the PRT server to connect to in the project. Enables use of a particular PRT service. If none is specified, all regions on any PRT system are displayed.
<code>=N</code> <code><node/project></code>	<code><node/project></code> specifies the node or project name with which to connect. Enables connection to the named node or project. If none is specified, the connection is made to the local project.
<code>/L</code>	Displays Region Label (drop down) browse support. Note: <code>/l</code> can be used in upper or lower case.
<code>/LID</code>	Displays both: <ul style="list-style-type: none"> • Region Label (drop down) browse support. • Region Id (drop down) browse support.
<code>=P</code> <code><path></code>	<code><path></code> is the path name where the configuration files are loaded. This option must be used if accessing a remote project.
<code>=I</code> <code><file path></code>	<code><file path></code> is the name of the file used to connect to multiple projects.

Example	
<code>"C:\Program Files\Proficy\Proficy CIMPLICITY\EXE\prt_ui.exe" -R schedule (page 211)</code>	
Where	=
<code>"C:\Program Files\Proficy\Proficy CIMPLICITY\EXE\prt_ui.exe"</code>	exe target location
<code>-R</code>	Switch for region
<code>schedule</code>	The PRT_UI will open in the Schedule region

5. Click **OK**.

Parameters that are indicated in the Target field will be triggered when the PRT User Interface is accessed.

 **Note:** If you are viewing the PRT User Interface from a Viewer, you do not have to use the

–N switch on the command line to connect to a remote project. When the prt_ui is initially launched, a dialog box will display allowing connection with a remote project that has been enabled for broadcast.

Step 2. Display and Browse for Regions in the PRT_UI

Step 2. Display and Browse for Regions in the PRT UI

Option 2.1 (page 212)	Display a region in the local project.
Option 2.2 (page 212)	Browse regions.
Option 2.3 (page 212)	Filter the list of regions.

Option 2.1. Display a Region in the Local Project

1. Click the down arrow in the **Region ID** field to display the list of configured Region IDs.
2. Select the region from the drop-down list.

The window displays the list of items for the selected region and updates the item count on the status bar.

 **Note:** If you are connected to multiple projects you can also display regions in those projects.

Option 2.2. Browse all Regions


1. On the PRT_UI, click the browse (...) button to the right of the **Region ID** field to open the Browse Regions browser.
2. Enter * in the Region Id field.
3. Click Browse.
4. Select a region from the list.

Note: You can [filter \(page 212\)](#) the list to facilitate finding the region the region you want.

5. Click **OK** to view the region in the PRT window.

Option 2.3. Filter the List of Regions

1. Click to the right of the **Region ID** field to open the Browse Regions browser in the PRT_UI.
2. Enter criteria in the **Region Id** field to filter the list of Region ID's.


 **Note:** Use the wildcard character (*) to increase the search result.

Example	Description
*R	Displays all regions ending in R.
S*	Displays all regions beginning with an S.
T	Displays all region with a T in the name.
*	Displays all available regions.

3. Click **Browse** to display the search results.
4. Select a region from the group.
5. Click **OK** to view the region in the PRT window.

Step 3. Auto-lock Regions

Step 3. Auto-lock Regions

 **Note:** In order to make modifications to the region, and to be able to move items into the region, it must be locked.

Regions that are configured to be lockable in the PRT Region Configuration dialog box must have certain status bits set when performing operations on items. The **Auto Lock** feature will automatically lock and retain the status bit settings for you whenever you perform operations on items within the region. You must turn on the auto lock feature to enable its functionality.

Method 1. Menu option

1. Click Operation on the menu bar.
2. Select Auto Lock.

Method 2. Quickest

Click .

A check mark displays next to Auto Lock on the Operation drop-down menu. The check mark indicates the feature is turned on.

Operations and Setting Status Bits

The operations affected by locked regions, and the status bit settings are

Operation	Source Region Status Bit	Destination Region Status Bit
Add Item	in_lock	N/A
Insert Item	in_lock	N/A
Delete Item	out_lock	in_lock
Fetch Item	out_lock	in_lock
Move Item	out_lock	in_lock
Advance Item	out_lock	in_lock

Step 4. Configure Items through the PRT_UI

Step 4. Configure Items through the PRT UI

In order to add, insert or modify an item, ensure that the [active region \(page 212\)](#) in the PRT window is the region in which you want to make changes.

Option 4.1 (page 214)	Add, or insert an item in the PRT_UI.
Option 4.2 (page 220)	Modify an item in the PRT_UI.

Option 4.1. Insert or Add an Item in the PRT UI

Option 4.1. Insert or Add an Item in the PRT UI

1. All items that were in the location in which the item is inserted move back one location to the end of the region.
2. Items that were in the location to which items were moved move back one location to the end of the region.
3. The items in each involved location continue to move back one until each is in one location closer to the end of the region.
4. (If the last location in the region currently has items) Tracker creates an additional location, which is now the last location and moves the last items into the new location.
5. Each item that was in Location 5, moves to Location 6.
6. Each item that was in Location 6, moves to Location 7.
7. Each item that was in Location 7, moves to Location 8.
8. Each item that was in Location 8, moves to Location 9.
9. Each item that was in Location 9, moves to Location 10.

10. Tracker creates Location 11.
11. Each item that was in Location 10, moves to Location 11.
12. The item is inserted into the next location.
13. Tracker treats the add as an insert command.

Option 4.1.1 Open an Add or Insert Item Dialog Box (except in Shifting Regions)

- Open an Add Item dialog box.
- Open an Insert Item dialog box.
- First and last location entries.

Open an Add Item dialog box

1. Either:
 - Select an item in the region.
 - Do not select an item in the region.
2. Do one of the following to open the Add Item dialog box.
 - Select File>Add on the PRT_UI menu bar or
 - Click on the toolbar or
 - Right-click a selected item and select Add from the Popup menu.

 **Note:** If the region is empty, use any of these methods to add the first item.

Result: The Add Item dialog box opens. When the item is created (after you fill in the fields on the General tab and click OK) the new item will be added to the region.


Example: Add item

3. An item in Location 3 is selected in the PRT_UI.
4. The Add Item dialog box opens.

The default location is 3. An [Attributes tab \(page 221\)](#) is available in the Modify Item dialog box.

Open an Insert Item dialog box

5. Either:
 - Select an item in the region.
 - Do not select an item in the region.
6. Do one of the following to open the Insert Item dialog box.
 - Select File>Insert on the PRT_UI window menu bar or
 - Click on the toolbar or
 - Right-click a selected item and select Insert from the Popup menu.

 **Note:** If the region is empty, use any of these methods to add the first item.

Result: The Insert Item dialog box opens. When the item is created (after you fill in the fields on the General tab and click OK) the new item will be inserted in the region as described above.

Example: Insert item

7. An item in Location 3 is selected in the PRT_UI.
8. The Insert Item dialog box is opened.


The default location is 3.

Note: An [Attributes tab \(page 221\)](#) is available in the Modify Item dialog box.

First and Last Location Entries

You can select to add or insert the item in the first or last location by entering the following values in the **Location** field.

Region Location	Enter
First	-2
Last	-1

 **Note:** If no location is selected when you open the Add Item or Insert Item dialog box, the default is -1. How Tracker adds or inserts an item into the last location depends on several factors .

Option 4.1.3. Define Item Status


1. Select the Item Status tab in the Add or Insert dialog box.

The status list displays.

2. Select a status item in the list.
3. Click one of the following to set the status of the item:

	Button	YES/NO	Description
A	Set Active	YES	Activates item status
B	Clear Active	NO	De-activates item status.

4. Click **OK** to save Item Status change(s) or click **Cancel** to cancel the operation.

 **Note:** The status list is configurable.

Option 4.1.2. Enter General Information for an Item

Select the General Information tab in the Add Item or Insert Item dialog box.

Fields are the following.

rect 4, 61, 205, 81 ([page 218](#))
 rect -4, 225, 399, 259 ([page 219](#))
 rect -4, 186, 399, 220 ([page 219](#))
 rect -2, 142, 229, 167 ([page 219](#))
 rect 207, 113, 394, 138 ([page 219](#))
 rect -2, 114, 209, 139 ([page 218](#))
 rect 203, 85, 388, 110 ([page 218](#))
 rect -1, 87, 206, 112 ([page 218](#))
 rect 203, 61, 389, 81 ([page 218](#))

1 (page 218)	Region ID
2 (page 218)	Location
3 (page 218)	Item ID
4 (page 218)	Item Class ID
5 (page 218)	Reference ID
6 (page 219)	Item Type
7 (page 219)	Parent Item ID
8 (page 219)	Int Hold/Reason
9 (page 219)	Ext Hold/Reason

1	Region ID
---	-----------

Read-only

The **Region ID** field displays the active region to which you are adding, inserting or modifying this item.

[Up \(page 217\)](#)

2	Location
---	----------

(Add and Insert Item dialog boxes)

The item's location in the region.

The default location is based on whether you selected:

- To add or insert the new item.
- A location in which to add or insert the new item.

[Up \(page 217\)](#)

3	Item ID
---	---------

A unique ID to identify a serialized item.

Important: If you do not enter an item ID for an item, Tracker will assign one. As a result, all items, serialized and non-serialized, will have item ID's. They will be disregarded when not needed.

Do not use the # sign as the first character in an item ID. # is reserved as the first character for an automatically generated item ID.

[Up \(page 217\)](#)

4	Item Class ID
---	---------------

An item class that was created in the TrackerCfg_UI.

[Up \(page 217\)](#)

5	Reference ID
---	--------------

Unique reference ID for this item.

Important: If you do not enter a reference ID for an item, Tracker will assign one. As a result, all items, serialized and non-serialized, will have reference ID's. They will be disregarded when not needed.

Do not use the # sign as the first character in a reference ID. # is reserved as the first character for an automatically generated reference ID.

[Up \(page 217\)](#)

6	Item Type
---	-----------

An item type that was created in the TrackerCfg_UI.

[Up \(page 217\)](#)

7	Parent Item ID
---	----------------

Non-serialized items

The ID of the serialized parent item that the selected non-serialized item should be associated with.

[Up \(page 217\)](#)

8	Int Hold
---	----------

Check if you want to place an internal hold on the item.

An internal hold can be set or cleared by the user using this check box or a user script.

Important: Internal hold is an informational state to inform users or processes of items that are to be held.

It is up to the user and processes to determine proper action based on this status. An item will still move through the tracker system if internal hold is set. It is up to the processes and users to route based on this status.

Reason	Enter the reason for the hold.
--------	--------------------------------

[Up \(page 217\)](#)

9	Ext Hold
---	----------

Check if there is an external hold placed on the item by an external process in the region.

An external hold is set or cleared by an external process. This status is used by the external process to notify Tracker that an item is to be held.

Tracker can take action based on this status.

Tracker can route the item depending on the status of external hold.

Important: An item will still move through the tracker system if external hold is set.

Reason	Enter an explanation for the hold.
--------	------------------------------------

Guidelines: Add, Insert or Modify an Item in Shifting Regions

1. The last item in the region is moved to the configured **detainment** region (if not configured, it is deleted).
2. All the items from the added location to the end location are shifted one location towards the end location.
 - If the last item was a blank item, it gets dropped; it does not move into the detainment region.
 - If the configured detainment region is a shifting region, the item at the last location is not moved to the detainment region; it is deleted. A configured detainment region should never be a shifting region.
3. The head item (first location) in the region is moved to the configured **transition** region (if not configured, it is deleted) and
4. All items are shifted one location towards the head location.
5. The last item in the region is moved to the **detainment** region (if configured, otherwise it is deleted)
6. All the items from the inserted location to the end location are shifted one location towards the end location.
If the last item was a blank item, then it gets dropped instead of moving into the detainment region.
7. The head item (first location) in the region is moved to the configured **transition** region (if not configured, it is deleted)
8. All items are shifted one location towards the head location.

Option 4.2. Modify an Item in the PRT_UI

Option 4.2. Modify an Item in the PRT_UI

Any item in the PRT_UI can be modified.

 **Note:**

- Standard and extended attributes are available only in the Modify Item dialog box.
- Item ID's and reference ID's that have been automatically entered begin with a # sign.

Option 4.2.1 (page 221)	Open a Modify Item dialog box.
Option 4.2.2 (page 221)	Set attributes for an item.

Option 4.2.1. Open a Modify Item Dialog Box

Do one of the following to open an item's Modify Item dialog box.

- Select File>Modify on the PRT_UI menu bar or
- Click on the toolbar or
- Right-click the selected item and select Modify from the popup menu.

Note: Item ID's and reference ID's that have been automatically entered begin with a # sign.

The Modify Item dialog box opens displaying current specifications for the item.

Option 4.2.2. Set Attributes for an Item

Option 4.2.2. Set Attributes for an Item

Step 4.2.2.1 (page 221)	Set Item Attributes (except ASSOCIATE attributes).
Step 4.2.2.2 (page 223)	Set ASSOCIATE Attributes for an item.

Option 4.2.2.1. Set Item Attributes (except ASSOCIATE attributes)

Select the Attributes tab in an item's Modify Item dialog box.

The attributes that are currently assigned to the selected item are listed.

Attribute types and function buttons are as follows.

A	Standard Attributes
---	---------------------

Stored in the project's backing files.

There can be a maximum of:

- 40 attributes that are logged in the Tracker log.
- 16 characters used in the name.

- 16 characters for the value.

B	Associated Attributes
---	-----------------------

- Stored either in an:
 - ODBC database or
 - File on in the ...bsm_Root\Tracker directory on the server.
- There can be a maximum of:
 - Unlimited attributes.
 - 32 characters in the name in the name.
 - 255 characters in the value.

C	Function buttons
---	------------------

Add

An Add Attribute dialog box opens when you click Add.

Field descriptions are as follows.

Field	Attribute Type	Description
Name	Standard	<ul style="list-style-type: none"> • The maximum size is 16 bytes. • May not duplicate within a single item. • Different items may have duplicate attribute names.
	Extended	<ul style="list-style-type: none"> • The maximum size is 32 bytes. • May not duplicate within a single item. • Different items may have duplicate attribute names.
Value	Standard	<ul style="list-style-type: none"> • Stored as a string in the PRT database. • New standard attributes cannot be created from extended attribute interfaces.
	Extended	<ul style="list-style-type: none"> • Stored in the extended attributes database. • A variant, a dynamic typing object. • Any user defined structure, hold structure, script, interface pointer, etc.
OK	Saves the new name and value and closes the Add Attribute dialog box.	
Cancel	Cancels the Add command.	

Delete

The Delete Attribute dialog box opens when you select an attribute and click Delete.

Options are:


OK	Deletes the attribute.
Cancel	Closes the dialog box without deleting the attribute.

Modify

A Modify Attribute dialog box opens when you click Modify.

You can change the value within the guidelines described for entered values.

Name	The name can be changed within the guidelines described for entered names.
Value	The value can be changed within the guidelines described for entered values.
OK	Saves the modifications and closes the Modify Attribute dialog box.
Cancel	Cancels the modifications.

 **Important:** Tracker attributes are designed for all machines that are in the system and in the same domain.

If the machines are not in the same domain an Administrator password needs to be set up to be the same as the remote system's password in order to communicate.

If the password is not set up and you transition an item to a project that is not in the domain, you will not be able to see its attributes.

Option 4.2.2.2. Set ASSOCIATE Attributes for an Item

- Associate two items.
- ITEM 1's associated attributes.
- ITEM 2's associated attributes.
- Notes about associated attributes.

Associate two items

1. Enter the following in the Add Attribute dialog box to create an ASSOCIATE attribute.

Field	Description
Name	Enter ASSOCIATE.
Value	Item ID of the item, <ITEM2>, that is being associated with the selected item.

2. Click OK.

ITEM 1's associated attributes

The extended and standard attributes for <ITEM2> are listed in <ITEM1>'s Modify Item dialog box as follows.

Extended Attributes

The extended attributes list in <ITEM1>'s Modify Item dialog box includes associate extended attributes for <ITEM2>.

Standard Attributes

The standard attributes list in <ITEM1>'s Modify Item dialog box includes associate standard attributes for <ITEM2>.

ITEM 2's associated attributes.

Open the Modify Items dialog box for the associated item, ITEM 2.

The extended and standard attributes for ITEM 1 are listed in ITEM 2's Modify Item dialog box as follows.

Extended Attributes

The extended attributes list in <ITEM2>'s Modify Item dialog box includes associate extended attributes for <ITEM1>.

Standard Attributes

The standard attributes list in <ITEM2>'s Modify Item dialog box includes associate standard attributes for <ITEM1>.

Notes about associated attributes

- You cannot specify the @ attributes in a condition of a query.

The attributes with an @ sign do not belong to the item where they are listed. They are virtual attributes that are listed in the item's attribute lists, but, in reality, belong to another item, the ASSOCIATE item.

As a result, no queries can be run specifically on an item with an @ attribute in the condition.

However, when you query an item for all its attributes, you will get all attributes, including the attributes of the associated item (if any).

- It is strongly recommended that you do not create any actual attributes on an item with an @ sign as the first character.

Although the system will not prevent a user from creating an attribute that begins with an @ sign, it will be a cause for ambiguity when the item has actual ASSOCIATE attributes. It will be difficult to distinguish a real attribute with an @ sign and the virtual associated item's attributes.

However, if you choose to create an attribute for an item with an @ sign as the first character, you can specify this attribute in a query and expect results.

- Add, Modify and deleting @ attributes is not supported in the PRT UI.

Step 5. Update a Region's Status

1. Click Operation on the menu bar.
2. Select the Region Status.

The Region Status dialog box opens.

A	Region ID	Selected region.
B	Total Items	Item count in the selected region.
C	Item Information	Active items in the region.
D	Region Status	Current settings for items in the region.
E	Set Active	Changes NO to YES (active).
F	Clear Active	Changes YES to NO (inactive).

3. Select the status bit from the **Region Status** list to change the settings for a region,
4. Click the appropriate button to set the status as needed:

Set Active	Activate status
Clear Active	Deactivates status

5. Click **OK** to save the changes to the region status or click **Cancel** to cancel the operation.

Step 6. Connect to Multiple Projects

Step 6. Connect to Multiple Projects

- Overview.
- Configuration steps to connect to multiple projects.

Overview

You can run multiple tracking projects with a high degree of integration. This integration lets you create different models that are specific to a particular process that work together in a coordinated fashion. For example, you may want to:

- Model different stages in a manufacturing process in separate projects, but still be able to track an item from stage to stage across project boundaries. This provides the flexibility to stop one project without bringing all of your production to a halt.
- Model parallel processes in separate models while preserving a high degree of coordination between them.
- Divide the workload among servers to reduce resource use on any single server.

 **Note:** Tracker Base provides powerful multi-project support; Order Execution Mgt. adds to this functionality. This step describes the multiple project feature for both.


Possible multiple project configurations include:

rect 5, 2, 151, 337 ([page 226](#))

rect 172, 24, 324, 357 ([page 226](#))

rect 338, 4, 506, 336 ([page 226](#))

	Configuration	Tracker Base	Order Execution Mgt.
A	Multiple projects on the same Server node	X	X
B	Multiple projects on different Server nodes	X	X
C	Viewer nodes (in addition to Server nodes)		
	Application		
	PRT UI	X	X
	BQM		X

 **Important:** You must create a client record for the remote project for the local project to connect properly.

1. Open the Workbench.
2. Select Security>Advanced>**Remote Projects**.
3. Create a new client.

If you want to view region data from an off-node project in the PRT_UI,

[Up \(page 226\)](#)

Configuration Steps to Connect to Multiple Projects

Step 6.1 (page 227)	Important connect to multiple project technical notes.
Step 6.2 (page 231)	Configure TrkCollector
Step 6.3 (page 234)	Configure DCOM default security settings
Step 6.4 (page 237)	Configure connections to multiple projects.

Step 6.1. Review Connect to Multiple Projects Technical Notes

Step 6.1. Review Connect to Multiple Projects Technical Notes

Requirements for:	
	Multiple project tracking support.
	Running more than one tracking project on the same node.
	Multiple Viewer support.
	Moving items across project boundaries
	Order Execution Mgt. storage.
and	
	Extended attribute storage options.
	Multiple project connection architecture.
	Multiple project software transactions.

Multiple project tracking support requirements

- Each tracking project must run in broadcast mode. Broadcast mode makes PRT data from one project available to other projects running on-node or off-node across a local area network.
- Item and Reference ID's must be unique across all projects.

You cannot have two item ID's or reference ID's with the same name, even if they are defined in different projects.

- Region names must be unique across all projects.

You cannot have two regions with the same name, even if the regions are in different projects.

Running more than one tracking project on the same node requirements

- If you plan on running more than one tracking project on the same node, the name of the PRT services for each project must be unique.

The default service names PRT_DC, PRT_DS and TRK_RP must be modified to be unique on a given node. Before you run more than one project on a node, [configure their PRT services uniquely \(page 177\)](#).

It is suggested that you add a prefix to the name of each service that indicates the project it belongs to, e.g. MYPROJECT1PRT_DC.

- Whenever you add another project to your multi-project configuration, you must delete the backing files for extended attributes. If you keep the extended attributes in backing files, do the following:

1. Before deleting these files, use the [PRT Export utility \(page 295\)](#) to export PRT data that includes extended attribute data.

Important: Make a file copy back up of the backing files as an extra precaution. Backing files for extended attributes are located in the ...\\Proficy CIMPLICITY\\Tracker folder.

2. Delete the current backing files from project log directory.
3. Start the project. This will recreate extended attribute backing files.
4. Import the extended attributes using the [PRT Import utility \(page 299\)](#).

Multiple Viewer support requirements

In order to view multiple projects:

- PRT Services must be running for each project.
- You must have access privileges to the projects you want to view.

Moving items across project boundaries requirements

If you want to move items across project boundaries, you have to do [additional PRT startup configuration \(page 227\)](#).

Order Execution Management central storage requirements

Central storage is:

- Required for:
 - Ranges/criteria sets.
 - Tracker Attribute database (TADB)

- Optional for extended attributes

Extended attribute storage options

Earlier versions stored extended attributes information in backing files. The new default is to store extended attributes in a database.

If you want to store Extended Attributes:	Then
In backing files	Edit the file TrkAttribute.cfg and change the Attribute Storage Bitmask value to 2.
In a database	Accept the default Attribute Storage Bitmask value. If this value has been changed, restore the value of this bitmask to its default value of 4. Example: 4 ALBTRK01 TrkEA sa sapw Note: See Tracker Extended Attributes Configuration (trkattribute.cfg) (page 314) for an explanation of the TrkAttribute.cfg format.

Multiple Project Connection Architecture

Multiple project architecture includes the following features.

 **Note:** Some features are available in Order Execution Mgt. only. [Availability \(page 229\)](#) is listed below.

rect 20, 47, 137, 95 [\(page 229\)](#)

rect 20, 127, 137, 220 [\(page 229\)](#)

rect 21, 217, 138, 285 [\(page 229\)](#)

rect 20, 298, 137, 359 [\(page 229\)](#)

rect 164, 41, 261, 103 [\(page 230\)](#)

rect 175, 112, 256, 193 [\(page 230\)](#)

rect 170, 205, 268, 265 [\(page 230\)](#)

rect 175, 286, 252, 370 [\(page 230\)](#)

rect 286, 122, 383, 184 [\(page 230\)](#)

rect 285, 202, 382, 264 [\(page 230\)](#)

rect 288, 295, 385, 357 [\(page 230\)](#)

Feature	Description	Tracker Base	Order Execution Mgt.
Agent Interface	Tracker.dll	X	X
	Tracker Agents Object Model is used to search region, item and attribute data across on-node and off-node projects.		
Tracker Collector	Windows Service	X	X
PRTSVR	In process .dll to support Tracker Collector	X	X
RCO1	RCO function blocks (page 376).	X	X

Event Server	Windows Service.	X	X
Extended Attributes (page 221) (Project)	Stored in Tracker Collector backing files.	X	X
PRT API	PRT Application Programming Interface.	X	X
Standard Backing File	Backing files on project (page 201) .	X	X
TRK1_RP	Tracker Resident Process.		X
PRT1_DC	PRT Data Collector (page 177) .	X	X
PRT1_DS	PRT Data Server (page 177) .	X	X
Ranges/Criteria Sets (Common)	Expands the concept of a Tracker source.		X
TADB Query (Common)	Tracker Query Engine (page 893) query to the TADB database.		X
Extended Attributes (Common)	Central storage across projects and nodes Configured in trkattribute.cfg (page 314) .		X

Multiple Project Software Transactions

You can do the following in Tracker when you run multiple projects.


Note: Available features are checked in the Tracker and Order Execution Mgt. columns.

Application	Transaction across projects and nodes	Tracker Base	Order Execution Mgt.
PRT_UI	View:		
	Items	X	X
	Attributes	X	X
	Named holds		X
	Update:		
	Items	X	X
	Named holds		X
	ASSOCIATE (page 136) items	X	X
	prt_ui /config -<profilename> (page 239)	X	X
PRT Client	Move:		
	Items with attributes	X	X
	Items with named holds		X

	PRT_Route destination region (16 characters)	X	X
Scripts	CimView and EMRP		X
	Object Model calls (i.e. Agent Interface)		X
Function Blocks	Ranges/Criteria sets		X
Queries	TADB queries		X
	Tracker attributes		X
Application	Transaction	Tracker	Order Execution Mgt.
Broadcast	<ul style="list-style-type: none"> Multiple BQM_ALR processes on: 		X
	Single node		
	<ul style="list-style-type: none"> BQM Web interface communicates with: 		X
	All broadcasts		
	<ul style="list-style-type: none"> RCO generates print data on: 		X
	Any node		
Event Server	Handles events on:		X
	<ul style="list-style-type: none"> Multiple on-node projects Off-node 		

Step 6.2. Configure TrkCollector

Step 6.2. Configure TrkCollector

 **Note:** In most cases, the TrkCollector will be accessed by off-node agents and PRT_UI's. In these cases, local and remote launch, activation and access will be required. However, review these settings based on your particular implementation and security requirements. It may be possible to run successfully with some of the suggested permissions turned off.

Steps to configure TrkCollector for Windows XP or Windows Server 2003 are as follows.

Step 6.2.1 <i>(page 232)</i>	Open the TrkCollector dialog box.
Step 6.2.2 <i>(page 232)</i>	Configure TrkCollector authentication.

Step 6.2.3 (page 232)	Configure TrkCollector security.
---	----------------------------------

Step 6.2.1. Open the TrkCollector Dialog Box

1. Click Start on the Windows task bar.
2. Select Run.

The Run dialog box opens.

3. Enter `DCOMCNFG` in the **Open** field.

The Component Services window opens.

Do the following.

Expand	
A	Expand Component Services.
B	Expand Computers.
C	Expand My Computer.
D	Select DCOM Config
E	Right-click TrkCollector in the Component Services right pane.
F	Select Properties.

The TrkCollector Properties dialog box opens.

Step 6.2.2. Configure TrkCollector Authentication

1. Select the General tab in the TrkCollector Properties dialog box.
2. Select Default in the **Authentication Level** field.
3. Click Apply.

Step 6.2.3. Configure TrkCollector Security

1. Select the Security tab in the TrkCollector Properties dialog box.
2. Configure permissions as follows.

rect 13, 49, 312, 131 ([page 233](#))

rect 10, 141, 309, 223 ([page 233](#))

rect 9, 235, 308, 317 ([page 234](#))

A (page 233)	Launch and Activation Permissions
B (page 233)	Access Permissions
C (page 234)	Configuration Permissions

A	Launch and Activation Permissions
---	-----------------------------------

3. Check Customize.

4. Click Edit....

The Launch Permissions dialog box opens.

5. Add the following.

A	If you are configuring local peer to peer/non-domain users:
	<ul style="list-style-type: none"> • Select the local node name (referred to as {node} in the List Names From: combo box. • If you are configuring a domain user Select the correct domain that has the user account you wish to use. The {node} prefix below would become a domain prefix now. <ul style="list-style-type: none"> • Make sure the following are selected. {node}\Administrators NETWORK SYSTEM
B	Check Allow for each of the following (for the selected group or user names). <ul style="list-style-type: none"> • Local Launch. • Remote Launch. • Local Activation. • Remote Activation.

6. Click OK.

Result: The TrkCollector Properties dialog box displays.

B	Access Permissions
---	--------------------

7. Check Customize.

8. Click Edit....

The Access Permissions dialog box opens.

9. Add the following.

A	<ul style="list-style-type: none"> If you are configuring local peer to peer/non-domain users:
	<p>Select the local node name (referred to as {node}) in the List Names From: combo box.</p> <ul style="list-style-type: none"> If you are configuring a domain user <p>Select the correct domain that has the user account you wish to use. The {node} prefix below would become a domain prefix now.</p> <ul style="list-style-type: none"> Make sure the following are selected. <p>{node}\Administrators NETWORK SYSTEM</p>
B	<p>Check Allow for each of the following (for the selected group or user names).</p> <ul style="list-style-type: none"> Local Access. Remote Access.

10. Click OK.

The TrkCollector Properties dialog box displays.

C	Configuration permissions
---	---------------------------

Retain default configuration permissions for TrkCollector.

The TrkCollector part of the configuration is complete.

Step 6.3. Configure the DCOM Default Security Settings

Step 6.3. Configure the DCOM Default Security Settings

! **Important:** These settings are suggested for PRT_UI multiple projects only. Your system may require other settings that are not included here.

Review Microsoft documentation for details about DCOM configuration.

Step 6.3.1 (page 234)	Open the My Computer Properties dialog box.
Step 6.3.2 (page 235)	Configure the DCOM default security settings.

Step 6.3.1. Open the My Computer Properties Dialog Box

1. Do one of the following.

Method 1. MS_DOS window

- Open an MS_DOS window.
- Type `dcomcnfg` at the prompt.

Method 2. Windows Start menu

- a. Click Start on the Windows task bar.
- b. Select Run on the Start menu.

The Run dialog box opens.

- a. Type `dcomcnfg` in the **Open** field.

Method 3: Windows Control Panel

- a. Open the Windows Control Panel.
- b. Select Administrative Tools>Component Services.


Result: The Component Services window opens when you use any method.

2. Double-click **Component Services** in the Component Services left pane.

Component Services expands.

3. Double-click **Computers**.
4. Right-click **My Computer**.
5. Select Properties on the Popup menu.

The My Computer Properties dialog box opens.

 **Note:** COM objects that programmatically initialize DCOM security will cause the DCOMCNFG settings to be ignored.

Step 6.3.2. Configure the DCOM Default Security Settings

1. Select the COM Security tab in the My Computer Properties dialog box.
2. Configure permissions as follows.

rect 6, 62, 334, 157 [\(page 236\)](#)

rect 6, 166, 334, 261 [\(page 236\)](#)

A (page 236)	Access Permissions
B (page 236)	Launch and Activation Permissions

A	Access Permissions
---	--------------------

3. Click Edit Limits....

The Access Permissions dialog box opens.

4. Add the following.

A	<ul style="list-style-type: none"> If you are configuring local peer to peer/non-domain users:
	<p>Select the local node name (referred to as {node} in the List Names From: combo box.</p> <ul style="list-style-type: none"> If you are configuring a domain user <p>Select the correct domain that has the user account you wish to use. The {node} prefix below would become a domain prefix now.</p> <ul style="list-style-type: none"> Make sure the following are selected. <p>{node}\Administrators NETWORK SYSTEM</p>
B	<p>Check Allow for each of the following (for the selected group or user names).</p> <ul style="list-style-type: none"> Local Access. Remote Access.

5. Click OK.

Result: The My Computer Properties dialog box displays.

B	Launch and Activation Permissions
---	-----------------------------------

6. Check Customize.

7. Click Edit....

The Launch Permissions dialog box opens.

8. Add the following.

A	<p>If you are configuring local peer to peer/non-domain users:</p>
	<ul style="list-style-type: none"> Select the local node name (referred to as {node} in the List Names From: combo box. If you are configuring a domain user <p>Select the correct domain that has the user account you wish to use. The {node} prefix below would become a domain prefix now.</p> <ul style="list-style-type: none"> Make sure the following are selected. <p>{node}\Administrators NETWORK SYSTEM</p>
B	<p>Check Allow for each of the following (for the selected group or user names).</p> <ul style="list-style-type: none"> Local Launch. Remote Launch. Local Activation. Remote Activation.

9. Click OK.

The My Computer Properties dialog box displays.

Step 6.4. Configure Connections to Multiple Projects

Step 6.4. Configure Connections to Multiple Projects

Step 6.4.1 <i>(page 237)</i>	Enable the PRT_UI Projects Option.
Step 6.4.2 <i>(page 238)</i>	Add projects.
Step 6.4.3 <i>(page 238)</i>	Save a PRT_UI .cfg File
Step 6.4.4 <i>(page 239)</i>	Open multiple projects using the saved .cfg file.

Step 6.4.1. Enable the PRT_UI Projects Option

1. Close the PRT_UI window, if it is open.
2. Click Start on the Windows task bar.
3. Select Run... from the Start menu.

The Run dialog box opens.

4. Enter `prt_ui /config` in the **Open** field.

Where

`prt_ui /config` is the command required to enable the Project option in the PRT_UI window.

5. Click OK.

If:	Then a:
No project is running.	Select CIMPLICITY Project dialog box opens for you to start a project. The PRT_UI opens for that project.
More than one project is running locally.	Select project to connect dialog box opens for you to select the project you want to work with first. The PRT_UI opens for that project.
One project is running locally.	The PRT_UI opens for that project.

Step 6.4.2. Add Projects to the PRT_UI Window Configuration

1. Select Operation>Projects on the PRT_UI menu bar to open

A Projects list box opens.

2. Click Add.

The Add Project dialog box opens.

3. Select a project from the drop-down list.

If you have authorization to work with the project, it is added to the list.

4. Click Close when you have added all of the projects you need.


A message displays asking you if you want to save the changes.

5. Click OK.

A PRT_UI message box asks you if you want to save the changes.

6. Do one of the following.

Click:	If you want to:
Yes	Save the changes and save a .cfg file.
No	use the configuration only this time.

 **Note:** You can remove a project from the project list by clicking Delete.

Step 6.4.3. Save a PRT_UI.cfg File

You can save a .cfg file in the project's directory if you want to save the list of projects you created when you added projects to the Projects dialog box.

You can then use the file in the future to open the projects in the PRT_UI window without having to re-list them in the Projects dialog box.

If, in [Step 6.2 \(page 238\)](#), you clicked Yes in the PRT_UI message box, a Save As dialog box opens.

1. Find the folder for the project you want to use as the primary project in when you open the configuration.
2. Name the .cfg file.
3. Click Save.

You will be able to re-open the listed projects that are running at any time by simply opening the saved .cfg file.

Step 6.4.4. Open Multiple Projects using a Saved .cfg File

1. Click Start on the Windows task bar.
2. Select Run... from the Start menu.

The Run dialog box opens.

3. Enter `prt_ui /config -I C:\projects\PRJ1\ui_cfg.cfg` in the **Open** field.

Where

`prt_ui /config -I` is the command

`C:\projects\PRJ1\ui_cfg.cfg` is the path to the .cfg file.

4. Click OK.

A Select project to connect dialog box opens.

5. Select a project from the list.
6. Click Connect.


Note: If a CIMPLICITY® Login dialog box opens, enter your **User ID** and **Password**.

The PRT_UI window opens. The regions from all of the projects listed in the .cfg file are available in the Region Id list.

Regions in two or more projects.

If the projects have region names that are the same, each instance of the name displays in the list.
--

You can view multiple projects and [move items \(page 242\)](#) from one to the other.

 **Important:** In order to view multiple projects:

- PRT Services must be running for each project
- You must have access privileges to the projects you want to view.

Step 7. Manipulate Items

Step 7. Manipulate Items

Option 7.1 (page 240)	Find an item.
Option 7.2 (page 241)	Advance an item.
Option 7.3 (page 241)	Move an item.
Option 7.4 (page 243)	Fetch an item.
Option 7.5 (page 244)	Reorder items.
Option 7.6 (page 244)	Filter by item class.
Option 7.7 (page 245)	Delete an item.

Option 7.1. Find an Item

1. Click **Find** .

The Find Item dialog box opens.

2. Provide information in one or both of the following fields:

Field	Enter
Item ID	ID of the item you want to find.
Reference ID	Reference ID of the item you want to find.

3. Click **OK** to locate the item, or click **Cancel** to cancel the operation.

Field	Displays
Item ID	ID of the item that is found.
Reference ID	Reference ID of the item that is found.
Region	Region in which the item is located
Location	Region's location in which the item is found.

Note: If the item cannot be located, a message box displays indicating the error. If the item is found, the Item Location box opens showing read-only information about the item.

4. Click **OK** to exit the window, or **Go To** to display the item in the region in the PRT window.

Option 7.2. Advance an Item

1. Select the item or items to be advanced.
2. Do one of the following.


Press **Ctrl+V**.

Select File>Advance on the PRT_UI menu bar.

 **Tip:** Use the **Ctrl** key to select multiple items.

The Advance Model dialog box opens.

3. Select the **Destination Region** from the drop-down list, or type in the region ID.
4. (Optional) Type any additional notes for the PRT Log file in the **Comment** box.
5. Click **Advance** to advance the item(s) to the entrance location of the destination region. The number of items successfully advanced will display in the **Advances** field.
6. Click **Cancel** to close the dialog box.

 **Note:** If there are no items in the Source region, or no routes configured for the Source region, a Warning message will display.

Option 7.3. Move an Item

Option 7.3. Move an Item

You move an item from one region to another region within the same project, or [across projects \(page 242\)](#) .

1. Select the item to be moved.
2. Click **Move** to open the Destination dialog box.

A	Check insert. Inserts item.
B	Clear insert. Adds item.

3. Enter the move specifications as follows:

Field	Enter or Select	
Region ID	Destination region	
Location	Location in which the item will be placed in the destination region.	
Comment	(Optional) additional notes for the PRT Log file	
Insert	Check	Inserts the item into the location
	Clear	Adds the item to the location.

4. Click **OK** to move the item to the destination region, or click **Cancel** to cancel the operation.

Guidelines: Move Items between Projects

When you are connected to multiple projects in the PRT_UI, you can move items between projects. In the PRT_UI the active region is displayed in the viewing window and all of the corresponding information is loaded into the appropriate fields.

You can move an item from the viewing window into another region within another project.

There are, however, some issues to take into consideration.

Item Status	The item status must be identical for items on different projects. The item status is a 32-bit number for all projects; however, if the same bits have different meanings from project to project, problems will arise. Modify the status using the Status Configuration dialog box in the TrackerCfg_UI on all projects to have identical statuses in order to move items effectively.
Item Types	In order to move an item between projects, the item type must be allowed in the region of the project receiving the item. The item class will be as it is configured in the PRT Item Type Configuration dialog box. If the item type is not allowed in the recipient project, a warning message will display the move will be disallowed.

Guidelines: Move Items into a Shifting Region

1. The last item in the region is moved to the configured **detainment** region (if not configured, it is deleted) and

2. All the items from the added location to the end location are shifted one location towards the end location.
If the last item was a blank item, then it gets dropped instead of moving into the detainment region.
3. The head item (first location) in the region is moved to the configured **transition** region (if configured, otherwise it is deleted) and
4. All items are shifted one location towards head.
A move by the user is not allowed for a blank item.

Option 7.4. Fetch an Item

1. Click File on the menu bar.
2. Select Fetch.

The Fetch Item dialog box opens.


A	Region location where the item is to be placed
B	Places item in the next consecutive region location

3. Enter fetch item specifications as follows:

Field	Enter or Select:	
Item ID	ID of the item that you want to retrieve in the field.	
Reference ID	Reference of the item you want to retrieve.	
Location	Region location where the item is to be placed.	
Fetch to next location	(Optional) Places the item in the next consecutive location.	
	Note: If you selected an item available to the PRT_UI window, the location will be automatically filled in. Example Active Region has the following occupied locations: L1, L2 and L3. Each location is capable of holding 2 items. Originally the region looks like the following:	
	Location	Item
	L1	Item1
	1.2	Item2
	1.3	Item3
	Retrieving a new item (Item4) into L3 in the region with the Fetch to next location box unchecked would result in the following:	
	Location	Item

	1.1	Item1
	1.2	Item2
	1.3	Item3
	1.3	Item4 (*Item is retrieved into the location to reside with the other item)
Alternately, retrieving a new item (Item4) into the region with the Fetch to next location box checked, would result in the following:		
	Location	Item
	1.1	Item1
	1.2	Item2
	1.3	Item3
	1.4	Item4 (*Item is retrieved into the next consecutive location)
Comments	(Optional) additional notes for the PRT Log file.	

- Click **OK** to move the item into the active region, or click **Cancel** to cancel the operation.

 **Note:** PRT will locate the item in its current region; you are not required to know the Region ID.

Option 7.5. Reorder Items in a Region

- Click Operation on the menu bar.
- Select Reorder to open the Reorder Region dialog box.
- Select the item in the list that you want to move.
- Click a button to move the item up or down the list as follows.

Move-Up	Moves the item up once per mouse click.
Move Down	Moves the item down once per mouse click.

- Specify a new location for an item, as follows.
 - Select the item.
 - Click Edit Loc to open the Edit Location dialog box.
 - Enter the location in the list to where you want the item moved.
 - Click OK.

The item is moved to the location you specified.

Option 7.6. Filter by Item Class

1. Click Operation on the menu bar.
2. Select Filter by Item Class....

The Filter by Item Class dialog box opens.

3. Select the classes, as follows.

Available Item Class box	Check the classes that should be included in the display.
Item Classes	(Read only) Displays the number of item classes that have been configured for the projects.
Display items without a configured class	Check to display items that have not been included in a class
Select All	Displays items in all the classes.
Clear all	Either: <ul style="list-style-type: none"> • Displays only items without a configured class, if the checkbox is checked. • Clears the region display.

4. Click OK to accept the filter or Cancel to cancel your changes.

Option 7.7. Delete an Item

Option 7.7. Delete an Item

1. Select the item in the list to be deleted.
2. Do one of the following.
 - Select File>Delete on the PRT_UI menu bar.
 - Click .
 - Press Delete on the keyboard.

The Delete Item dialog box opens.

3. Check one of the following:

Checkbox	Description	
Delete	Deletes the item from PRT, TADB and the Historical database. Deleting the item is essentially administrative. e.g. An item may be deleted because it does not belong there. It was entered due to a clerical error.	
	Log string	Deleted
Scrap	Deletes the item, but identifies it as scrap. Scrapping occurs usually because the item is damaged or cannot be used for some reason; the occurrence needs to be recorded..	
	Log string	Scrapped

Archive	Deletes the item from PRT and the TADB database; retains the item in the Historical database. Retaining item data enables it to be retrieved if the item needs to be re-entered somewhere or its data needs to be reviewed.	
	Log string	Production stopped

4. Click one of the following.

Button	Description
OK	Confirms the action.
Cancel	Cancels the action.

Guidelines: Delete or Move an Item from a Shifting Region


1. All items from the end location to that location are shifted by one location towards the head location and
2. A blank item is added at the end location.
A Delete by the user is not allowed for a blank item.
3. All items are shifted by one location towards the head location and
4. A blank item is added at the end location.

Step 8. Print Reports

Step 8. Print Reports

Based on the current view in the PRT window, you can print a report with the same columns of information. The View drop-down list contains the different views configured you configured using the **prt_window.cfg** file. You can select the view that has the information you want to print, and then print out a report for your records. Use the print options to set up, preview and print out your report.

Step 8.1. <i>(page 247)</i>	Set up a printed report.
Step 8.2. <i>(page 247)</i>	Preview the printed report.
Step 8.3. <i>(page 248)</i>	Print the report.

 **Note:** Production Tracking also provides the capability to generate reports using logged data. A common method for logging data is by using Microsoft's ODBC SQL Server, and then generating reports using a report writer.

Step 8.1. Set up a Printed Report

1. Open the PRT_UI.
2. Select the Region for which data will be printed.
3. Select the view that displays the information you want to print from the **View** drop-down list.
4. Do one of the following.

Option 1

Make sure that no items are selected.

The printed report will contain all of the items in the region.

Option 2

Select items by holding down the

- Ctrl key to select individual items or
- Shift key to select a block of items.

The printed report will contain the selected items in the region.

5. Click File on the menu bar.
6. Select Print Setup.
The Print Setup dialog box opens.
7. Make any required changes to the print specifications, including Portrait or Landscape orientation.
8. Click **OK** to save the changes, or click **Cancel** to cancel the operation.


Step 8.2. Preview a Printed Report

1. Click File on the menu bar.
2. Select the **Print Preview** option to see how the printed report will look.
3. The button bar at the top of the window performs the following functions:

	Click	To
A	Print	Print out the report as it is displayed in view.
B	Next Page	Go to the next page of the report.
C	Prev Page	Go to the previous page in the report.
D	Two Page	View two pages in the viewing area.
E	Zoom In	Enlarge the view on screen.
F	Zoom Out	Reduce the view on screen.
G	Close	Exit print preview and return to the PRT window.

4. The status bar at the bottom of the window displays the:

- Page number
- Number of locations the report contains.
- Number of items the report contains.

 **Tip:** If there is too much or not enough information in the report, use the View drop-down list to select another file configuration. If the information is not displayed properly or the way you want it, go to Print Setup to adjust the settings.

Step 8.3. Print a Report

1. Click File on the menu bar.
2. Select **Print** option to open the Print dialog box.
3. Use the default printer or select another printer from the **Name** drop-down list.
4. Specify what should be printed, number of copies, etc.
5. Click **OK** to print or **Cancel** to cancel the operation.

PRT Technical Reference


PRT Technical Reference

Before the implementation of the Tracker Configuration User Interface, the Tracking Model was configured using ASCII-text files.

The need for this has been eliminated; however, for the convenience of our clients who are accustomed to editing these files, all of the information you need is included in this section.

1 (page 249)	PRT Configuration file overview.
---	----------------------------------

2 (page 271)	PRT Configuration file programs and utilities.
3 (page 315)	Startup configuration for PRT Client.
4 (page 318)	PRT Service and Resident Process name change configuration.
5 (page 318)	PRT validation program, verifiers and error codes.

 **Note:** Acronyms that are used include:

Acronym	Meaning
ISAM	Indexed sequential access method.
SCPOP	System configuration population.
ASCII	American Standard Code for Information Interchange.

1. PRT Configuration File Overview

1. PRT Configuration File Overview

Identify the PRT configuration files.
Use the IDTPOP utility.
Observe configuration file reserved characters and restrictions.
PRT configuration file definitions.

Identify the PRT Configuration Files

Identify the PRT Configuration Files

There are several required ISAM files to configure to run your tracking model, as well as several optional files for additional functionality.

Tracker primarily uses binary coded files to run PRT.

You can readily identify the files you want to work with by their **.prt** prefix and **.dat** or **.cfg** extension.

PRT configuration files include:

Required configuration files.

Optional configuration files.

Required PRT Configuration Files

The required configuration files **must** be configured in order to run your tracking model.

All of these files have a **.dat** extension, with the exception of the last one, which has a **.cfg** extension.

Files with a **.dat** extension are binary; files with a **.cfg** extension are in text format.

File Name	File Definition	Description
prt_sysdef (page 268)	PRT System Definitions	Sets the queue sizes for the ports used by the Data Collector, Data Service and Application Interface subsystems. Minimum recommended size is 10 for each port.
prt_service (page 266)	PRT Service Record	Identifies each PRT Service in the tracking model.
prt_intproc (page 258)	PRT Interested Process/Data Provider Mapping	Enables related processes to receive updates from the tracking model. <code>PRT_GRD</code> requires this file to be configured in order to function properly.
prt_group (page 258)	PRT Group	Identifies the Groups in your tracking model. Because every region must belong to a group, this file must be configured.
prt_region (page 260)	PRT Region	Identifies the regions in your tracking model, their primary and secondary types and other important information.
prt_grp_reg (page 258)	PRT Group Tracking Region	Ties the tracking regions to their tracking groups.
prt_item_typ (page 259)	PRT Item Type	Defines the item types, classes and parent-child relationships between items.
prt_route (page 264)	PRT Route	Identifies routes between regions and associations with transfer points.
prt_attributes (page 256)	PRT Attribute Labels	Configures a set of attributes for all items or for items in a specific class. This is not a required file, but is recommended.

Optional PRT Configuration Files

The optional configuration files are not required to run a basic tracking model. However the configuration of your tracking model may require some or all of these files in order to provide you with the kind of information you want to collect.

All of these files have a **. dat** extension, with the exception of the last two, which have a **. cfg** extension.

Files with a **. dat** extension are binary; files with a **. cfg** extension are in text format.

File Name	File Definition	Description
prt_type_trn (page 268)	PRT Item Type Translation	Translates item type codes received from detection devices into item type IDs that can be recognized by Tracker. This file must be configured when the prt_route records have a translation type value of 2
prt_inv_type (page 259)	PRT Route Invalid Type	Identifies an item type(s) that is invalid for a specified route.
prt_assoc (page 254)	PRT Region Item Association	Defines the item type roles for parent/child associations, of which there is one record configured for each item type.
prt_status (page 267)	PRT Status Code	Determines the status bits in the track model for both regions and items.
prt_disp_pt (page 257)	PRT Region Item Type Display Point	Configures the item type quantity in a region that is to be displayed through a setpoint. Note: Must be configured when prt_grd is configured.
prt_att_pt (page 255)	PRT Region Item Attribute Display Point	Identifies the setpoint ID that will link item attributes by region. Note: Must be configured when prt_grd is configured.
prt_reg_at (page 260)	PRT Region Attribute Counter Definitions	Indicates the counters in item attributes. This will in turn trigger the action taken when an item containing the defined attribute enters the region.
prt_window.cfg (page 269)	PRT Field Labels	Defines the column names for the prt_ui (PRT User Interface) in place of attribute names. A default record is provided with Tracker and must be present to run the prt_ui.

Use the IDTPOP Utility

1. Select Tools>Command Prompt on the CIMPLICITY Workbench menu bar.

The Command window opens in the project's directory.

2. Enter the following.
 - a. Type `cd master`
 - b. Press ENTER to switch to the main directory.
 - c. Type `idtpop <filename>`, e.g. `idtpop prt_region` .

Important: Do not include the file extension in the file name.

- a. Press **ENTER**.

Production Tracker process the records in the file and displays the number processed.

- a. Type <text editor> <filename.idt>, e.g . **notepad prt_region.idt** .
- b. Press **ENTER**.

The text editor will open with the ASCII-text version of the file displayed.

Observe Configuration File Reserved Characters and Restrictions

Observe Configuration File Reserved Characters and Restrictions

Each configuration file has a set of reserved characters and restrictions that you must observe. Once you understand how the files are set up, you will have no trouble configuring your tracking model.

Details about configuration file setup include:

	PRT configuration file reserved characters.
	PRT configuration file restrictions.
	PRT configuration file configuration file format.

PRT Configuration File Reserved Characters

The first line in the . **idt** file must define three reserved characters. All other characters in the line are ignored. You may select characters other than the ones used here; however, avoid using characters that will be used in the data fields.

- The first character defines the **delimiter**. This character will be used to separate fields in the record. In the example below, the vertical bar (|) is used for a delimiter.
- The second character is used to show a **continuation** of a record. When a record exceeds the length of one line, this character is used to append the next line to the record. In the example below, the hyphen (-) is used for a continuation character.
- The third character identifies the **comment** character. When the comment character is used before text, it is not recognized as part of the record, but rather additional information for the user. In the example below, the asterisk (*) is used for a comment character.

-*	Reserved characters generated by IDTPOP the utility.
*	Text is comment only; not part of the record.
	Separates the fields.
-	Continuation of the record; appends the next line to the record.

PRT Configuration File Restrictions

The following restrictions must be observed when editing the configuration files.

Delimiter	All fields must be separated by a delimiter, which is the first character defined in the first line of the file.
Field length	Must be less than or equal to its configured maximum. Cannot exceed 250 characters.
Field Order	The fields must be in the same order in the record as they appear in the input file.
Field Type	Vary depending on the file being edited; enter values appropriate for the file.
Field Values	Cannot exceed the specified size of the field. Do not support embedded spaces; use underscore (_) character instead. Exception: Text or description fields support embedded spaces.
Fields	Empty fields are supported, but must be delimited in the record.
Logical line length	Must be less than or equal to 2000 characters. Includes all continued lines in the record.
Padding	Character fields are blank-padded automatically when the value is less than the field size. Numeric fields are zero-filled automatically when the value is less than the field size.
Physical line length	Must be less than or equal to 250 characters.

PRT Configuration File Format

When the ASCII-text version of the file is viewed in a text editor, they all have the same basic format.

A	Reserved characters, delimiter, continuation and comment.
B	Name of the record.
C	Comments that describe each field in sequential order in the record.
D	Records with values in each field as they correspond to the above comments.

Example 99 11 Assembly Model 25 1 BLOCK			
Where			
99	=	0	ITEM_TYPE_ID
11	=	1	item_type_code
Assembly Model 25	=	2	item_type_desc
1	=	3	item_tracking_type
BLOCK	=	4	item_class_id

PRT Configuration File Definitions

PRT Configuration File Definitions

A definition is provided for each configuration file to include field name, description, data type and an entry example. Use the definitions to edit configuration files as needed for your Tracking Model.

The following configuration file definitions are covered:

PRT_Assoc
PRT_Att_Pt
PRT_Attributes.cfg
PRT_Disg_Pt
PRT_Group
PRT_Grp_Reg
PRT_Intproc
PRT_Inv_Type
PRT_Item_Typ
PRT_Reg_At
PRT_Region
PRT_Route
PRT_Service
PRT_Status
PRT_Sysdef
PRT_Type_Trn
PRT_Window.cfg

PRT ASSOC

The PRT Item Association (**prt_assoc**) configuration file is used to define a parent-child association between a serialized and non-serialized item. Configure one record for every parent-child association to be made during production.

OPTIONAL FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	region_id	Name of the region in which the parent-child association will take place.	16 characters	PROCLNE2_COM
1	item_type_id	Unique name of the item type that will enter the above region (can be serialized or non-serialized).	16 characters	WSHR_FRAME

2	role-code	Code indicating the role of the item type as follows:		One byte	1
		1	Parent item (serialized)		
		2	Child item (non-serialized item),		
		3	Bystander (no association will take place)		

PRT ATT PT

The PRT Region Item Attribute Display Point (**prt_att_pt**) configuration file is used to define the setpoint ID that will hold the value for an item attribute in a region location. This file must be configured when **prt_grd** is configured.

OPTIONAL FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	gd_service_id	Unique name of the Process Service ID.	32 characters	PRTGRD
1	region_id	Unique name for the region.	16 characters	OVEN1
2	item_att_name	Textual description of the item attribute, as follows: PRT_ITEM_ID PRT_REFERENCE_ID PRT_ITEM_TYPE_ID PRT_ITEM_STATUS PRT_ASSOC_ITEM Note: When using prt_item_status in this field, the values in att_start_loc and att_end_loc must both be 0.	16 characters	ITEM_ID
3	display_ptid	Unique name of the array point. The definition of the point depends on the item_att_name field, as follows: PRT_ITEM_ID – Point class must be Text; Point type must define a text size of at least 20. PRT_REFERENCE_ID – Point class must be Text; Point type must define a text size of at least 20. PRT_ITEM_TYPE_ID – Point class must be Text; Point type must define a text size of at least 20. PRT_ITEM_STATUS – Point class must be Analog; Point type must be ANALOG_U32. PRT_ASSOC – Point class must be Text; Point type must define a text size of at least 20. User defined – Point class must be Text. The number of region locations covered by the Point determines the number of elements needed. Use this formula to calculate the number of elements: (end_loc) – (start_loc) + 1	55 characters	OVEN1_INDEX
4	start_loc	First location in the region covered by this Point. If this value is '-2', then the Point displays data from the last <n> items in the region, where <n> is the number of elements in the Point.	Integer	1
5	end_loc	Last location in the region covered by this Point. Due to a restriction on Point size, one Point may not be able to cover an entire region; in which case, the region must be broken into subsets. Each subset as defined by a starting and ending location, is covered by a single Point.	Integer	25
6	att_start_loc	First location in the attribute to parse when setting value of display_ptid . The default value is 0. See Note.	Integer	0

7	att_end_loc	Last location in the attribute to parse when setting value of display_ptid . The default value is 0. See Note.	Integer	0
8	item_class_id	Name of the item class that can provide values for this Point. If a class is not specified, any item type will provide data. If more than one item matches the item filtering, the value of the Point is indeterminate.	16 characters	DRYER

 **Note:** The value of (**att_end_loc** – **att_start_loc** + 1) can be no greater than the data length of **display_ptid** .

When using the default value (0) in either the **att_end_loc** or the **att_start_loc** field, the entire attribute will be transferred to **display_ptid** and no parsing will take place.


PRT_ATTRIBUTES.CFG

The PRT Attribute Labels (**prt_attributes.cfg**) configuration file is used to assign attributes to your items. Attributes can be assigned to all item types or can be assigned to a specific item class.

When assigning attributes to an item class, use the keyword **ITEMCLASS**.

For each PRT Service that you have designated in your Tracking Model, you will need a separate configuration file named for the Service.

OPTIONAL FILE

 **Note:** This file is recommended in order to gather comprehensive data on the items tracked through production.

1. Open a `prt_attributes.cfg` file as follows.
 - a. Select Tools>Command prompt on the Workbench menu bar to open the Command prompt window.
 - b. Type `cd master`.
 - c. Press Enter.
 - d. Type `notepad prt_attributes.cfg`.

(`notepad` represents the text editor.)

A `prt_attributes.cfg` file opens in the text editor.

The following lines are filled in:

Line 1	- *
Line 2	*
Line 3	List all attribute names in a single record (separated by a *)
Line 4	*

2. Define an item class beginning with Line 5, as follows:

Line 5	<code>ITEMCLASS Item Class Name</code> Where <code>ITEMCLASS</code> is the keyword <code>Class Name</code> is the name of the item class
Line 6	<code>Attribute, Attribute Name Attribute, Attribute Name Attribute, Attribute Name ...</code> Where <code>Attribute</code> is the keyword, e.g. <code>Color</code> <code>Attribute Name</code> is the selected attribute, e.g. <code>white</code> Be sure to use the delimiter character to separate attributes.
Line 7 and 8+	Repeat lines 5 and 6 until you have entered all of the attributes for all of the involved item classes.

3. Click File on the menu bar once you have assigned attributes for your items and proceed as follows:

Do	When
Select Save	The <code>prt_ui</code> is configured to view all PRT Services.
Select Save As	The <code>prt_ui</code> is configured to view a specific service Then a. A. Enter <code>aprt_window.cfg</code> in the File Name field, Where <code>aprt</code> is the name of the PRT Service a. B. Click Save .

A	All items will have these attributes.
B	Keyword <code>ITEMCLASS</code> assigns attributes to the named item class.
C	Attributes apply to the <code>WSHR ITEM CLASS</code> only.

3. Close the text editor.

PRT_DISP_PT

The PRT Display Point (`prt_disp_pt`) configuration file is used to define the Item Type quantity in a region to be displayed through a setpoint. This file must be configured when `prt_grd` is configured.

OPTIONAL FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	<code>region_id</code>	Unique name for the tracking region.	16 characters	<code>RINSE1_NOR</code>
1	<code>item_type_id</code>	Unique name for the item type.	16 characters	<code>WSH_FRAME</code>
2	<code>qty_ptid</code>	Unique name for the setpoint to display the item type quantity in the region.	55 characters	<code>R1_CTQ</code>

PRT_GROUP

The PRT Group (**prt_group**) configuration file is used to define each of the Tracking Groups.

REQUIRED FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	<code>group_id</code>	Unique name for the tracking group.	16 characters	EASTAREA_B
1	<code>group_label</code>	Text to appear with the tracking region ID for user recognition.	16 characters	EAST AREA B
2	<code>Group_desc</code>	Description of the tracking group.	40 characters	EAST AREA B Tracking
3	<code>resource_id</code>	Unique name of the resource for the group.	16 characters	TRACKING
4	<code>Log_grp_mod</code>	Flag for logging tracking group modifications, i.e. change of hold status on group.	One byte	1

PRT_GRP_REG

The PRT Group Tracking Region (**prt_grp_reg**) configuration file is used to tie Regions to Groups.

REQUIRED FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	<code>group_id</code>	Unique name for the Group responsible for the corresponding region (see Field 1).	16 characters	EntAreaA
1	<code>region_id</code>	Unique name for the region in the Group.	16 characters	AsmbyLn1
2	<code>region_seq</code>	Sequence number of the region in this Group. Optionally, use this field to represent the physical order of the regions in a production line or in the group (used for display purposes only).	Integer	1

Prt Intproc

```
|-* IDT file generated by IDTPOP utility v1.0
* RECORD: PRT_INTPROC Interested Process/Data provider mapping
*
* 0 prt_svc_id PRT Service ID providing
info. to above
```

```

* 1 INTPROC_SVC_ID Interested Process Service ID
* 2 update_mask 1=Full, 0=Brief(all but auto
move )
*
APRT_DC | PRTCNT | 1

```

PRT_INV_TYPE

The PRT Invalid Type (**prt_inv_type**) configuration file is used to specify when a configured item type is not valid for a tracking route. Recall that you can use the wildcard character in the **prt_route** configuration file to identify the item types. When the wildcard character is used, you can use this file to eliminate the invalid item types from the route.

OPTIONAL FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	orig_region_id	Unique name of the region from which the item type will exit.	16 characters	PROCLNE1_NOR
1	dest_region_id	Unique name of the region where the item type may enter.	16 characters	RINSE1_NOR
2	item_type_id	Unique name of the item type that is not allowed to travel between the regions; in other words, the invalid item type.	16 characters	WSH_MTR
3	transition_ptid	Point ID that represents the transition between the two regions.	55 characters	PL1_RS1_TRANS

PRT_ITEM_TYP

The PRT Item Type (prt_item_typ) configuration file is used to define all of the item types that the Tracking Model is responsible for monitoring.

OPTIONAL FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	<i>item_type_id</i>	Unique name for the item type.	16 characters	COMPRESS
1	<i>item_type_code</i>	Unique code identifying the item type to be sued internally by PRT.	Integer	4
2	<i>item_type_desc</i>	Textual description of the item type.	40 characters	Compressor Item Type
3	<i>item_tracking_type</i>	Code identifying the tracking type:	One byte	1

		1	Serialized		
		2	Non-serialized		
4	<code>item_class_id</code>	Name of the item class to which this item type belongs. Note: Item Classes are not required but are recommended. One item class can be created for all item types.		16 characters	Refrig

PRT REG AT

The PRT Region Attribute Counter Definitions (**prt_reg_at**) configuration file is used to define counters in item attributes. This will in turn trigger the action taken when an item containing the defined attribute enters the region.

OPTIONAL FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	region	Name of the region where the operation will take place	16 characters	OVEN1
1	reserved_fld	GE Intelligent Platforms reserved.	16 characters	Blank
2	att_name	Name of the attribute containing the counter.	16 characters	ITEM_COUNT
3	start_byte	Index of the first character of the count within the attribute, i.e. 1-16. A value of 0 indicates use of the entire attribute.	Long word	0
4	end_byte	Index of the last character of the count within the attribute, i.e. 1-16.	Long word	3
5	op_code	Operation type code that is to impact the counter, as follows:	One character	I
		I Increment		
		D Decrease		
		P Preset		
6	alarm_value	Enter the value that will generate an alarm based on the op_code field, as follows:	Long word	4
		I (Increment) Over value will generate alarm.		
		D (Decrease) Under value will generate alarm.		
		P (Preset) value is equal to the preset value.		

PRT REGION

The PRT Region (**prt_region**) configuration file is used to define every region in the Tracking Model.

REQUIRED FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	<code>prt_svc_id</code>	Specifies the PRT Data Collector Resident Process Service ID. This must be identical to the ID in service.dat .	32 characters	APRT_DC
1	<code>region_id</code>	Unique name for the tracking region.	16 characters	ASMBLY LNE1_NOR
2	<code>region_desc</code>	Textual description of the region.	40 characters	Assembly Line 1 Normal Region
3	<code>region_label</code>	Text to display with the region for user recognition.	16 characters	AssmbyLne1 Nor
4	<code>tracking_type_id</code>	Primary tracking region type: Pool, Detainment, Sequential or Shifting	16 characters	SEQUENTIAL
5	<code>region_cap</code>	Maximum number of available physical region locations.	Integer	15
6	<code>region_loc_cap</code>	Maximum number of items that can reside at a single region location. Be sure to include all serialized and non-serialized items.	Integer	3
7	<code>oos_cycle_count</code>	Maximum number of items allowed to exit out of sequence from a tracking region.	Integer	1
8	<code>oos_error_code</code>	Indicates the action to be taken when an item does not arrive at the region exit transition indicator as expected. Use one of the following:	8 characters	DLY_DTN
		DETAIN	Move to detainment region.	
		SCRAP	Scrap (delete) item.	
		DLY_DTN	Allow item to be out-of-sequence using the oos_cycle_count condition, then move to detainment region.	
		DLY_SCRP	Allow item to be out-of- oos_cycle_count sequence using the condition, then scrap (delete).	
		BLANK	(No item in the combo box) - Disable out-of-sequence. Used if the sequence of items entering a region is not important. This will scrap (delete) the item.	
9	<code>GEF_reserved_1</code>	GE Intelligent Platforms Reserved.	8 characters	Blank
10	<code>GEF_reserved_2</code>	GE Intelligent Platforms Reserved.	8 characters	Blank

11	GEF_reserved_3	GE Intelligent Platforms Reserved.	8 characters	Blank
12	GEF_reserved_4	GE Intelligent Platforms Reserved.	8 characters	Blank
13	det_region_id	Unique name of the detainment region where items are sent when out of sequence.	16 characters	DETAINMNT_A
14	region_lockable	Enable/disable (1/0) locked region, i.e. region must be locked to make modifications or to move items into the region.	One byte	0
15	alarm_det_region	Enable/disable (1/0) alarm generation when items are moved to the detainment region.	One byte	1
16	alarm_region_not_empty	Enable/disable (1/0) alarm generation when a disperse tracking region is not empty and new items have arrived in the region.	One byte	0
17	alarm_region_lock	Enable/disable (1/0) alarm generation when an item enters an IN-LOCKED region or exits an OUT-LOCKED region.	One byte	0
18	alarm_loc_cap_excd	Enable/disable (1/0) alarm generation when the number of items at a single region location exceeds the configured maximum.	One byte	1
19	alarm_item_hold	Enable/disable (1/0) alarm generation when an item enters a region with an attribute that matches an item-hold specification.	One byte	1
20	alarm_unknown_item	Enable/disable (1/0) alarm generation when information cannot be retrieved for a serialized item.	One byte	1
21	enable_master_alarm_flag	Enable/disable (1/0) alarming for this tracking region regardless of alarming enabled elsewhere.	One byte	1
22	enable_master_log_flag	Enable/disable (1/0) logging for this tracking region regardless of logging events enabled elsewhere.	One byte	1
23	log_det_region	Enable/disable (1/0) logging when items are moved to the detainment region.	One byte	1
24	log_region_not_empty	Enable/disable (1/0) logging when a disperse tracking region is not empty and new items have arrived in the region.	One byte	0
25	log_region_stat	Enable/disable (1/0) logging when an item enters an IN-LOCKED region or exits an OUT-LOCKED region.	One byte	0
26	log_loc_cap_excd	Enable/disable (1/0) logging when the number of items at a single region location exceeds the configured maximum.	One byte	1
27	log_item_hold	Enable/disable (1/0) logging when an item enters a region with an attribute that matches an item-hold specification.	One byte	1
28	log_unknown_item	Enable/disable (1/0) logging when information cannot be retrieved for a serialized item.	One byte	1
29	log_item_scrapped	Enable/disable (1/0) logging when an item has been scrapped (deleted) from the tracking region's queue.	One byte	1

30	log_item_entry	Enable/disable (1/0) logging whenever an item is entered into queue.	One byte	1
31	log_item_exit	Enable/disable (1/0) logging whenever an item exits the tracking queue.	One byte	1
32	log_item_mod	Enable/disable (1/0) logging whenever an item's tracking data is modified.	One byte	0
33	log_prod_start	Enable/disable (1/0) logging when tracking of an item starts in this region.	One byte	0
34	log_prod_stop	Enable/disable (1/0) logging when tracking of an item stops after this region.	One byte	0
35	item_qty_ptid	Point ID that displays the total number of items contained in the region; must be of type INT or UINT.	55 characters	ASMBLYLNE1_TOTAL
36	stat_ptid	Point ID that represents the region's status; must be of type UDINT.	55 characters	ASMBLYLNE1_STAT
37	region_type	Secondary tracking region type. Use the following codes:	Integer	1
		1	Normal	
		2	Combine	
		3	Disperse	
		Note: If a Combine or Disperse region, region_cap field must be set to 1, and region_loc_cap must be greater than 1.		
38	auto_assoc_region	Indicates if items are automatically associated in this region; must be of secondary region type Combine. Yes/No (1/0).	One byte	0
39	resource_id	Uniquely identifies the resource against which the alarm will be generated at the tracking region level. Note: If a resource is not designated here, the alarm will be generated against the default, "\$SYSTEM."	16 characters	OpWks_1
40	cfg_override	GE Intelligent Platforms Reserved.	One byte	Blank
41	alarm_oos_region	Enable/disable (1/0) alarm generation when an "out-of-sequence" condition occurs.	One byte	1
42	alarm_inval_type	Enable/disable (1/0) alarm generation when an invalid item type enters the region.	One byte	1
43	log_inval_type	Enable/disable (1/0) logging when an invalid item type enters the region.	One byte	1
44	alarm_unknown_type	Enable/disable (1/0) alarm generation when an unknown item type enters the region.	One byte	1
45	log_unknown_type	Enable/disable (1/0) logging when an unknown item type enters the region.	One byte	1
46	alarm_reg_cap	Enable/disable (1/0) alarm generation when the region's configured capacity has been exceeded.	One byte	0
47	log_reg_cap	Enable/disable (1/0) logging when the region's configured capacity has been exceeded.	One byte	0

48	<code>auto_move_point</code>	Text Point ID that displays when an item moved into the region due to a "Point-based" move. The Point contains the item ID and optional information depending on the value of the <code>num_setpt_param</code> set in the global parameters files.	55 characters	Blank
49	<code>tran_region_id</code>	Indicates the transition region ID for a shifting region.	16 characters	Blank

PRT ROUTEPRT ROUTE

The PRT Route (`prt_route.idt`) configuration file is used to define all of the routes that exist between regions. This enables automatic transition of items based on changes in CIMPLICITY Point values.

REQUIRED FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	<code>orig_region_id</code>	Name of the source region (where the item will exit).	16 characters	NULL
1	<code>dest_region_id</code>	Name of the destination region (where the item will enter).	16 characters	SRC1
2	<code>item_type_id</code>	Unique name for the item type that will travel between the two regions. Use the wild card character (*) to validate all item types through this route.	16 characters	NULL
3	<code>transition_ptid</code>	Point ID that represents the transition between the two regions.	55 characters	X_S1
4	<code>set_ptid</code>	Point ID that will display when the item is transitioned. Must be of class, Text and should be no more than 36 characters. Data is displayed as follows: 16 characters – Item Type ID. 20 characters – Item ID when item type is serialized; otherwise, associated or parent item ID.	55 characters	T_S1
5	<code>translation_id</code>	Depending on Field 6 (<code>translation_type_code</code>), configure as follows: <code>translation_type_code</code> = 2: translation to use to interpret this transition indicator. <code>translation_type_code</code> = 3, 4 or 5: Up to 3 characters that will be attached as a prefix to the Point value prior to processing. This allows items with the same ID to be identified in different parts of the factory. <code>translation_type_code</code> = 1, 6 or 7: NOT USED.	16 characters	NULL
6	<code>translation_type_code</code>	Unique code to represent the tracking indicator type as follows:	One byte	3
		1	Limit switch	
		2	Type detector (integer Point)	
		3	Automatic Item ID displaying Item ID (Text Point)	

		4	Automatic Item ID displaying Reference ID (Text Point)		
		5	Associated Reference ID to Item (Text Point)		
		6 (page 265)	Associated Item Type to Item (Integer Point)		
		7	Positive Edge Trigger (transition occurs when value changes from zero to nonzero)		
		8	Negative Edge Trigger (transition occurs when value changes from nonzero to zero)		
7	Process_first_ptchange		<ul style="list-style-type: none"> FALSE or undefined transition point ignores the first point change of a route. TRUE processes the first point change as a transition point. 	0=FALSE 1=TRUE	0

PRT_ROUTE (Type 6) Configuration Example

A user can create an \$UNKNOWN item type and enter items of this type in the source region. If the user changes the transition point id to an integer value of the item type that should be moved, Tracker searches for an item with an \$UNKNOWN type in the Route's source region.

If an \$UNKNOWN item type is:	
Found	The <code>item_type_is</code> of the item is changed to the matching one in the <code>prt_type_trn.idt</code> file.
Not found	An item, which is the type that the user entered, is created in the destination region.

 **Note:** An \$UNKNOWN item type is non-serialized.

File examples

prt_route.idt
prt_type_trn.idt

prt_route.idt

```
|-* IDT file generated by IDTPOP utility v1.0
* RECORD: PRT_ROUTE Production Tracking Route
*
* 0 orig_region_id          ID of region item exits
* 1 dest_region_id         ID of region item enters
* 2 item_type_id           ID of type traveling between regions
* 3 transition_ptid        ID of transition point between regions
* 4 set_ptid               ID of point to be set on transition
* 5 translation_id         ID to interp type trans indicator
* 6 transition_type_code   transition indicator type code (1-8)
```

```
* 7 process_first_ptchange          Should we process the first point
change
*
B_TP1|A_TP1||B1_X_A1||XLATE|6|1
* XLATE Above is the translation ID that will match a record in
prt_type_trn.idt
* The transition_ptid must be an integer. When the user enters this value
it must match one of the input codes in the prt_type_trn.idt file
```

prt_type_trn.idt

```
|-* IDT file generated by IDTPOP utility v1.0
* RECORD: PRT_TYPE_TRN Prod. Tracking Item Type Translation
*
* 0 TRANSLATION_ID          Item Type Translation Id
* 1 input_code              Code sent by Item type detection
equip.
* 2 item_type_id            Item Type ID for type detected
*
XLATE|1|NONSER
XLATE|2|NONSER_TYPE2
* There can be several records in this file
prt_item_typ.idt
|-* IDT file generated by IDTPOP utility v1.0
* RECORD: PRT_ITEM_TYP Production Tracking Item Type
*
* 0 ITEM_TYPE_ID            Unique ID for Item Type
* 1 item_type_code          Unique code for Item Type used by
PRT
* 2 item_type_desc          Item Type Description
* 3 item_tracking_type      Tracking type code (1=ser, 2=nonser)
* 4 item_class_id           Item Class Identifier
*
NONSER|3||2|
NONSER_TYPE2|2||2|
$UNKNOWN|4||2|
```

PRT SERVICE

The PRT Service (**prt_service**) configuration file is used to identify each Production Tracking Service.

REQUIRED FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	<code>prt_svc_id</code>	Specifies the PRT Data Collector Resident Process Service ID. This must be identical to the ID in service.dat .	32 characters	APRT_DC

1	<code>prt_search_order</code>	Indicates the order in which this service will be searched for information static to other configured PRT Services.	1 byte	1
2	<code>ext_item_svc_id</code>	Service ID of the external process providing item tracking data.	32 characters	Blank
3	<code>ext_hold_svc_id</code>	Service ID of the external process providing hold information	32 characters	Blank
4	<code>det_reg_alrmid</code>	Alarm ID for items placed in detainment region.	32 characters	PRT_DET_REF_FULL
5	<code>region_lock_alrmid</code>	Alarm ID generated when an item enters IN-LOCKED region, or exit OUT-LOCKED region via a point-based move.	32 characters	PRT_ITM_THRU_LCK
6	<code>region_cap_alrmid</code>	Alarm ID when the maximum number of item carriers in the region are exceeded.	32 characters	PRT_REG_CAP
7	<code>loc_cap_alrmid</code>	Alarm ID when the maximum number of items at a single region location / item carrier have been exceeded.	32 characters	PRT_REF_LOC_CAP
8	<code>seq_err_alrmid</code>	Alarm ID when an item is out of sequence at a region exit transition indicator.	32 characters	PRT_REG_OOS
9	<code>item_hold_alrmid</code>	Alarm ID when an item arrives in a tracking region and is determined to have an item-hold specification.	32 characters	PRT_ITM_WITH_HLD
10	<code>invalid_type_alrmid</code>	Alarm ID when an invalid item type is detected.	32 characters	PRT_INVALID_TYPE
11	<code>unknown_type_alrmid</code>	Alarm ID when an unknown item type is detected.	32 characters	PRT_UNKNOWN_TYPE
12	<code>invalid_id_alrmid</code>	Alarm ID when a serialized item enters a region and information cannot be retrieved for that item.	32 characters	PRT_INVALID_ITEM
13	<code>region_not_empty_alrmid</code>	Alarm ID when all items have not left a disperse tracking region and new items have arrived in the region.	32 characters	PRT_DSP_NOT_MPTY
14	<code>Batch_queue</code>	NOT USED	Not used	Blank

PRT STATUS

The PRT Status (`prt_status`) configuration file is used to assign a status code for each valid region status; for example, capacity, sequential error, invalid or unknown type, etc. Similarly, a record is configured for each valid item status; for example, hold, no-hold, delayed, etc. A number of statuses have predefined records, and you may configure additional records to define other status types.

OPTIONAL FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	status_code	Unique code to identify the status of the region or item.	8 characters	CTRLWTRK
1	code_type	Unique code representing what you are defining a status code for, as follows: 0 = Region Status Code 1 = Item Status Code 2 = Reserved	One byte	0
2	code_desc	Textual description of the status code.	40 characters	Blank
3	code_label	Text to display with the status code for user recognition. Must be unique.	16 characters	Control w/ track
4	bit_set	Bit to be set by PRT that will represent this status code.	Long word	22

PRT_SYSDEF

The PRT System Definitions (**prt_sysdef.idt**) configuration file is used to define the queue sizes for the ports used by the Data Collector, Data Server and Application Interface subsystems.

REQUIRED FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	<code>api_req_port</code>	Sets the queue size for the Application Interface's request port. This is used to receive requests for item and item hold data from the PRT data collector(s).	Integer	10
1	<code>api_sync_port</code>	Sets the queue size for the Application Interface's synchronous port. This is used to send requests to PRT data collector(s), and to send requests and receive responses from the PRT data server(s).	Integer	10
2	<code>api_int_port</code>	Sets the queue size for the Application Interface's interested process port. This is used to receive interested process updates from the PRT data server(s).	Integer	10
3	<code>dc_main_port</code>	Sets the queue size for the data collector's main port.	Integer	20
4	<code>ds_main_port</code>	Sets the queue size for the data server's main port.	Integer	20
5	<code>ds_throttle_port</code>	Sets the queue size for the data server's throttle port.	Integer	20
6	<code>ds_int_port</code>	Sets the queue size for the data server's interested process port.	Integer	50
7	<code>dc_aux_port</code>	Sets the queue size for the data collector's auxiliary port.	Integer	10

 **Important:** It is recommended that the ports in this record be set to a minimum value of 10.

PRT_TYPE_TRN

The PRT Item Type Translation (**prt_type_trn**) configuration file is used to translate item type codes received from detection equipment into item type IDs. Once translated, Tracker recognizes the IDs via the PRT database. Configure this file when your **prt_route** records have a translation type code of 2.

OPTIONAL FILE

The fields are:

No.	Field Name	Description	Data Type	Entry Example
0	translation_id	Unique name of the item type translation.	16 characters	NOCHANGE
1	input_code	Unique code received from detection equipment signaling the item type.	Integer	10
2	item_type_id	Unique name of the item type that is to be translated using the input code from the detection equipment.	16 characters	WSHR_MOTOR

Configuration File Example

PRT_WINDOW.CFG

The PRT Window (Field Labels) (**prt_window.cfg**) configuration file is used to customize the columns and their order in the PRT User Interface (prt_ui). This file has a DEFAULT record already configured. If you want to rearrange or limit the information on screen, you can do so by View or by Region. This feature comes in handy for printing reports and for customizing a screen to display only the information needed by a technician.

Further, this file can be used to configure different views for each of your designated PRT Services, or views to run across all PRT Services.

OPTIONAL FILE


A View record:

- Is common to all regions.
- Specifies what information displays.
- Arranges information in the display.
- Customizes labels (column headings).
- Uses the keyword DEFAULT in the first line.

A Region View record:

- Impacts a specific region.
- Specifies what information displays.

- Arranges information in the display.
- Customizes labels (column headings).
- Overrides other views.
- Uses the keyword REGION in the first line.

 **todo: To configure the prt_window.cfg file:**

1. Click Tools on the Workbench menu bar.
2. Select Command Prompt
A DOS window opens.
3. Type **cd master** on the Command Line .
4. Press **Enter** to switch to the Master directory.
5. Type **notepad prt_window.cfg** (where **Notepad** is the text editor).
6. Press **Enter**.

The default record for prt_window.cfg will display in the text editor.

7. Do one of the following:

Option 1

- a. Type REGION to create a common view for all of your regions.
- b. Type the delimiter, e.g. |.
- c. Type a name for the view, e.g. VIEW1.

Option 2

- a. Type REGION to create a common view for all of your regions.
- b. Type the delimiter, e.g. |.
- c. Type the name of the region, e.g. ROBOT1.

8. Press **Enter**.
9. Create your record(s) using the field descriptions as follows:

No.	Field Name	Description	Data Type
0	field_type	Code defining the field type as follows:	One character
		0	Internal
		1	Attribute
		2	Status

1	prt_field	PRT field for which the modified label will be used. Use the following keywords: PRT_ASSOC_ITEM PRT_EXT_HOLD PRT_GROUP_ID PRT_INT_HOLD PRT_ITEM_ID PRT_ITEM_CLASS_ID PRT_ITEM_TYPE PRT_REFERENCE_ID PRT_REGION_ID	16 characters
2	prt_label	Label or column head to be alternately used in the PRT window. Recommend title case for alternate labels.	16 characters
3	field_size	Width of the field by number of characters.	Integer

10. Click File on the menu bar after you have created all of your views and proceed as follows:

Do	When
Select Save	The prt_ui is configured to view all PRT Services.
Select Save As	The prt_ui is configured to view a specific service Then a. Enter aprt_window.cfg in the File Name field, Where aprt is the name of the PRT Service a. Click Save .

11. Close the text editor.

The records created in the prt_window.cfg file will impact the type and arrangement of the information displayed in the PRT window by PRT Service, by common view and by region.

A	DEFAULT	Already configured
B	DEFAULT_VIEW	Common view, record created.
C	REGION	Region view, record created.

2. PRT Configuration File Programs and Utilities

2. PRT Configuration File Programs and Utilities

Production tracking global section usage.
Production tracking processes.
Automated region validation.
Shifting region technical reference.
Attribute counts by region.
PRT synchronization and alarm move exceptions.
PRT interested process utility.
Production Tracking export program: PRT_EXPORT.

Production Tracking import program: PRT_IMPORT.
Tracker Extended Attributes Configuration.
Tracker data logging.

Production Tracking Global Section Usage

PRT global sections are:

- Created by the Data Collector process for each PRT service
- Mapped onto by the corresponding Data Server Process.

The backing files are located in **%SITE_ROOT%\LOG** .

The following types of sections are created:

<GROUPS>.<PRT Service Name>	This file is used to maintain Group hold status information.
<ITEM_XREF0>.<PRT Service Name>	This file is used to maintain cross- reference information about serialized items to allow fast access to items.
<REGION ID>0.<PRT Service Name>	This file is used to maintain item information within a particular region (there may be many of this section type per PRT service).

Where

<PRT Service Name> is the service name of the Data Collector process and

<REGION ID> is a configured region identifier for this service.

It is not necessary to manually delete any of these global sections, since data integrity will then become uncertain. The exception is if changes are made to any of the following configuration files:

- prt_service
- prt_region
- prt_group
- ptr_item_typ.

Global memory corruption may occur if changes are made to this configuration data and new backing files are not created when PRT is rebooted.

In order for Production Tracking to eliminate the corruption you must perform the following procedure.


1. Terminate both the PRT data collector and server.

2. Delete the appropriate region files.
3. Restart the PRT data collector. This ensures that the cross-reference file will be deleted then reconstructed.
4. Restart the PRT server.

When you have performed this procedure, at startup, Production Tracking.

5. Deletes the cross-reference file, ITEM_XREF0.<PRT Service Name>, then
6. Reconstructs it based on the contents of the region files that are available.


This allows the system manager to eliminate the contents of certain regions without affecting the other regions in the service, or performing an "export-edit-import" function.

 **Note:** The regions are pre-allocated to a configuration-based size. If items are to be added to a region that is full, the contents of the head location are moved to the full region's detainment region. If the full region is a detainment region, or there is no detainment region configured for the full region, the contents of the head location are deleted.

Production Tracking Processes

The processes within Production Tracking that implement the functions of the Production Tracking Subsystem are:

File	Process	Description
prtc.exe	Production Tracking Data Collector	Process ID must terminate with _DC for interprocess communication to work properly.
prtx.exe	Production Tracking Data Server	Process ID must terminate with _DS for interprocess communication to work properly.
prtv.exe	Production Tracking Configuration Validation Program	<ul style="list-style-type: none"> • The CIMPPLICITY project for which you are checking the configuration must be running. • This utility may be run interactively. • Reads PRT configuration files, outputting information on a file-by-file basis about errors and/or discrepancies in configuration.

 **Note:** PRT Export does not transfer region validation statistics data.

Automated Region Validation

Automated Region Validation

Automated PRT Region represents a combination of functions that address the need to effectively process positive identification data (such as, Smarteye, RF Tag or bar code readers).

This functionality is used to make sure that:

- The CIMPLICITY Tracking Image is accurate at the head of zones where readers are located and
- Decisions are not made on an item until it is at the head of a zone and the tracking system has been verified against the reader.

This can be used to reduce manual intervention when a read is missed or misread. It also creates an enhanced self-healing capability without the need for single-carrier virtual regions at the head of each region with a reader. (This setup was used to utilize the positive ID transitions that made sure the tail of the destination region was corrected).

The components of the automated system are:

[\(page 274\)](#)

An overview of the system and its components is shown above. In general terms the following is a list of major components and their roles.

Component	Role
Reader	(Triangle) The carrier IDs are read by the reader and are put in to a reader point.
Carrier at head:	(Circle) This is a trigger that tells PRT that a carrier has just come to the head of the region and the reader point should contain the ID of the carrier. This trigger is usually a limit switch at the head of the region indicating that there is a carrier present. The Reader point should not be used for this trigger if the user wants no-read cases to be detected by this system.
Manual Entry 1 Correct Reader Value):	<p>If the reader</p> <ul style="list-style-type: none"> • Has a no read or the PRT does not match the read and • The region cannot be validated. <p>Then the solution should provide a way of manually entering the correct read.</p>
Manual Entry 2 (Re-triggering the validation process):	In case the region validation process fails (this will probably be as a result of bad read or no read), the solution should provide a way of forcing the system to validate itself again. This is usually needed in combination with the Manual Entry 1. The mechanism can either be automatic (via a heartbeat) or manual.

If the users' confidence in readers is high or they are planning a different behavior in case of a no-read, users can set up the system so that reader value can play the role of validation trigger as well. In this case it is users' responsibility make sure that no-read cases are handled.

Sequence of Events for Region Validation

An assumption for this system is that there can only be one carrier between the reader and the head of the region when PRT is triggered for the existence of a carrier at the head of that region. This will eliminate the timing problems that can arise in different circumstances.

A carrier passes by the reader at the head of a region.

1	The value of the read (if there is any) is copied to a device point that is configured for that reader
2	This value is then copied into a an Equation with override derived point for the reader
	At this point the equation with override derived point types are only available for analog points. This design allows users to configure analog points for the reader values. These points are then converted to ASCII for internal processing using a region specific C formatting string supplied in the configuration. This derived point will be the mechanism to manually enter the correct reader values if necessary. After giving the system enough time to get the complete read in to the device point, a trigger notifies the PRT subsystem about the new item at the head of the region
3	The trigger is usually a limit switch at the head of the region, which acts as a carrier at head indicator.
	The trigger value is copied into a derived point similar to the reader point
4	The derived point provides a mechanism to re-trigger the PRT validation process.

rect 58, 73, 136, 113 ([page 275](#))

rect 39, 133, 115, 157 ([page 275](#))

rect 62, 182, 120, 208 ([page 275](#))

rect 170, 3, 217, 34 ([page 275](#))

rect 183, 84, 226, 122 ([page 275](#))

rect 191, 130, 222, 156 ([page 275](#))

rect 195, 160, 226, 195 ([page 275](#))

rect 269, 20, 304, 47 ([page 275](#))

rect 187, 221, 232, 250 ([page 275](#))

rect 224, 302, 254, 323 ([page 275](#))

5	Changes in the derived trigger point trigger PRT.
6 and 7	PRT retrieves the latest value of the reader and (7) the configuration for the region on how to proceed with the validation.
8, 9 and 11	After the validation process is completed PRT updates the results of the process (8)(9) and updates some runtime statistics (11).
12	After the validation process is completed, the PRT subsystem should provide a mechanism for letting other subsystems know about the completion of the validation process.
	This would help eliminate the timing issues that might arise between the PRT validation process and the processes (such as RCO s or an event manager driven routing mechanism) that may act on the validity of regions after the validation process has completed. Since validation statistics are going to be updated after every validation, this point will be used to trigger other processes.

*Different Scenarios for Flow of Control**Different Scenarios for Flow of Control*

In this section different plant setup scenarios that will result in different operational behavior are illustrated.

	Setup 1
	Setup 2
	Setup 3
	Setup 4

Setup 1. Flow of Control

PRT validation is triggered by the 'carrier at stop' point. PRT does its validation and then sets the statistics point, which in turn triggers the RCO.

Setup 2. Flow of Control

PRT validation is triggered with the reader point. In this case, PRT cannot catch no-read cases. The systems relies on the processing performed by the RCO when the carrier sets the carrier at stop point. In the case of manual intervention (setting the correct reader value), the RCO can be triggered by the user or the RCO can be configured to re-trigger automatically.

Setup 3. Flow of Control

In this configuration, there is no carrier at stop point. The system relies on the readers being correct most of the time.

Setup 4. Flow of Control

This is a combination of Configuration 1 and 2. Handling of the No-Read cases for the regions are handled in PRT. The RCO is then triggered independently with a heartbeat. This automatic trigger can be combined with the validation triggers. Note that because of multiple regions, some of the items shown in the previous configurations are eliminated to make the figure more readable.

prt_reg_val.cfg

Region specific configuration is done with prt_reg_val.cfg.

A layout of the text is:

Field	Description	
Region_id	Region id	
Item_or_ref	Type of matching to perform. Valid values are:	
	1	match item_id
	2	match reference_id
Reader_point_id	Reader values are provided with this point. It can be a string or numeric point. The reason that we are providing a numeric point as reader point is that Equation with override derived point is only applicable to numeric points.	
Class_id	Class Id to use for matching (used if multiple item per location configured, and if only one item type is being identified. For example, PRT can be configured to contain 2 items per location, one carrier and one buck, and the readers in the system read the id's of carriers. In this case the user configures PRT so that carrier item types and buck item types are in different classes; and uses carrier class in this field)	
Validation_trigger	When received, depending on the next field, used to trigger the validation process. This field is ignored if the next field is Use_Reader .	
Validation_trigger_type	Validation trigger types are:	
	1	Transition_high
	2	Transition_low
	3	Limit_switch
	4	Use_Reader_point
Statistics_point_id	Point id to put the run time statistics. It should be a DWORD array of 9 elements, but if it is less than 9 only a portion of the statistics will be written.	
Reader_format	It is a C print formatting string. It should be in quotes. For string points use variations of %s, for analog points use variation of %ld. If empty "%s" (or "%ld" depending on the point type) is used. Reader values are formatted with this string before trying to match it to the head item's item_id or reference_id.	
Invalid_reset_value	When a region has a mismatch with reader value, region image is assumed invalid. Region's invalid count is set to this value. This count shows the number of consecutive good reads that is necessary to make the region valid again.	
Control_With_Track	This value is used when backing files are created for the first time.	
Use_OOS	If the region image does not match the read, and the item that should have been at the head of the region is not at the head, but it is in the same region; Automated Region Validation will move the items that are in front of the identified item to the detainment region of the region. The move is done as item moves if this flag is set to 0. Otherwise (if it is 1), out of sequence processing feature of PRT will be used to determine how those items will be treated.	

prt_svc_val.cfg

Service specific configuration is done with prt_svc_val.cfg.

Command_point and response_point will be used to expand the functionality of PRT without changing the interface to it.

Command point has the following general format:

< REFERENCE >,< COMMAND >,< PARAMETER-LIST >

Where

< REFERENCE > is copied back to the response point, and can be used by the clients to match the responses to the command that they have sent. The following is the list of commands supported and their formats:

<REFERENCE>,\$RESET_STATS,{<REGION_ID>|\$ALL},<INDEX>

<REFERENCE>,\$RESET_PREV_RD,{<REGION_ID>|\$ALL}

Note that \$ALL is used to refer to the all regions of the PRT service. If < INDEX > is 0, all the elements of the statistics point will be set to 0. Otherwise, the specified element will be set to 0.

The field descriptions are:

Field	Description
Service_id	Service id
Warning_alarm_id	Alarm Id to use for validation alarms that PRT was able to fix. If not configured, the alarms are logged to cor_status.log .
Failure_alarm_id	Alarm Id to use for validation alarms that PRT was not able to fix, and requires manual correction of the system. If not configured, the alarms are logged to cor_status.log .
Command_point	PRT command point
Response_point	PRT response point
keep_validation_flags	If it is 1, it means to keep the validity flags from the global memory when PRT starts up. If set to 0, reset the validity flags to FALSE regardless of their values in the global memory.

Runtime Data

Runtime Data

The validation process is also required to maintain some statistics about the performance. The following table gives the list of the elements of the statistics point.

Index	Field	Description
0	MODIFY_MASK	Collection of bits giving information about the validation process just completed. Details of the bits are given in the following table.
1	TOTAL_VALIDATIONS	Number of times the region validation is triggered.

2	INVALID_COUNT	Current 'Region Invalid' count. Region is Invalid if this count is greater than 0.
3	TOTAL_INVALID	Number of times the region has become invalid
4	TOTAL_DBL_TRIGGER	Number of times the region is double triggered (Triggered multiple times, with the same read while the head is validated)
5	TOTAL_MISMATCH	Number of times head of the region did not match the reader value
6	CONS_MISMATCH	Consecutive mismatch count
7	TOTAL_NOREAD	Number of times region validation is triggered with no new read.
8	CONS_NOREAD	Consecutive no read count

MODIFY_MASK Bits

The first element of the statistics point contains information about the validation process just performed. This information is in the form of bits that are described below.

Bit	Field	Description
0	HEAD_ITEM_VALID	The Head Item of the region was valid when the point was set.
1	TOTAL_VALIDATIONS_MOD	Statistics array point's value at index 1 has been modified
2	INVALID_COUNT_MOD	Statistics array point's value at index 2 has been modified
3	TOTAL_BECOMING_INVALID_MOD	Statistics array point's value at index 3 has been modified
4	TOTAL_DOUBLE_TRIGGERS_MOD	Statistics array point's value at index 4 has been modified
5	TOTAL_MISMATCH_MOD	Statistics array point's value at index 5 has been modified
6	CONSECUTIVE_MISMATCH_MOD	Statistics array point's value at index 6 has been modified
7	TOTAL_NOREAD_MOD	Statistics array point's value at index 7 has been modified
8	CONSECUTIVE_NOREAD_MOD	Statistics array point's value at index 8 has been modified
27	RESERVED_READ	Read is ignored (reserved read starting with a \$).
28	STATS_RESET	Statistics point has been reset
29	NEW_ITEM	Item not found in the current PRT Service, and a new item has been created
30	REORDER	Region has been validated by reordering the region.
31	MOVE	Region has been validated by moving an item from another region.

Shifting Region Technical Reference

Shifting region tracking configuration

To configure a shifting region, the corresponding record in the PRT_REGION file should have:

- **tracking_type_id** : SHIFTING
- **tran_region_id: region_id** of the transition region of shifting region.

When a new shifting region is configured, on the subsequent startup PRT will create blank items automatically for all the locations. These blank items are only placeholders; they cannot be deleted, added, or moved by the user.

! **Important:** While configuring item types, do not configure any as **item_type_id**, which is a GE Intelligent Platforms reserved word for **item_type_id**s of blank items for the Import/Export utilities.

Shifting region process behavior

Other process behavior regarding shifting regions includes:

Process	Description
PRT	PRT never returns blank items when a request for an item list is made. It will return only the valid items, and the RegionLoc field in the item structure will contain the actual location of that item.
GLB_VERIFY	GLB_VERIFY checks that the number of occupied locations in a shifting region is equal to the region capacity.
prt_ui	<code>prt_ui</code> displays a blank row for each blank location in a shifting region. A user cannot modify, delete, or move items at these locations.
PRT_EXPORT/ PRT_IMPORT	<ul style="list-style-type: none"> • While exporting a shifting region, <p><code>prt_export</code> generates one record for every blank location (even if the location can contain multiple items). This record contains the location number and the <code>item_type_id</code> as \$\$\$GEF@BLK@ITM\$\$.</p> <ul style="list-style-type: none"> • On import of these records with the reserved item_type_ids, <p><code>prt_import</code> generates blank item(s) at those locations.</p>
PRTCNT	lf:
	The <code>prt_att_pt</code> file is configured to display n to n+m items in a region and <ul style="list-style-type: none"> • Any of the locations in that range is blank, • Then no data is displayed for that location.
	lf:

	<ul style="list-style-type: none"> • The prt_att_pt file is configured to display the last n items in a shifting region and • The number of valid items are equal to or greater than n. <p>Then the last n valid items will be displayed.</p> <ul style="list-style-type: none"> • The prt_att_pt file is configured to display the last n items in a shifting region and • The number of valid items are is less than n. <p>Then those items will be displayed and the remaining locations will display blanks.</p>
--	--

Attribute Counts by Region

Attribute Counts by Region

Production Tracking counts PRT items based on:

- Region(s)
- Item class
- Various item attributes criteria

An item is included in a count if its attribute matches a configured set of criteria.

Components for attribute counts by region include:

Attribute counts by region: <code>PRTCNT</code>
Attribute counts fo region: <code>prtcnt.cfg</code>
Count items: <code>prtcnt.exe</code> (Upgrade to from CIMPPLICITY 6.0 or earlier)

Attribute Counts by Region: PRTCNT

In order for an item to be included in the count, the attribute value must be compared against a user-defined value. The comparison can use any of the following user-defined operators:

Operator	Description
<code>==</code>	Equals
<code>!=</code>	Not Equals
<code><</code>	Less Than
<code><=</code>	Less Than Or Equal
<code>></code>	Greater Than
<code>>=</code>	Greater Than Or Equal
<code>@</code>	Between
<code>!@</code>	Not Between

%	Exists
\$	Contains

The comparison may be:

- Numeric(N)
- Alphanumeric(A)
- Float(F)

Alphanumeric comparison values can specify the wildcard character ?.

Multiple region names can be specified by separating them with commas.

Example

Specify "Source Regions 1 and 2 and Distribution Regions 1 and 2":

SRC1, SRC2,DST1,DST2.

When using @ (**Between**) or !@ (**Not Between**), separate the high and low values with a comma.

Example

Specify "Between 2 and 3":

|@|2,3|

Counts are stored in CIMPLICITY analog non-array points. These points are initialized at startup and are updated as items enter and exit a region.

When the process first starts it:

1. Performs a count of all items in the configured regions
2. Initializes the points with the count values.
3. Utilizes a configured reset point; when the reset point updates, `PRTCNT` will automatically recount all regions. When an item is modified in a configured region, `PRTCNT` will recount all counters associated with that region.

Attribute Counts by Region: prtcnt.cfg

Tracker looks for the configuration file prtcnt.cfg in the project's \data.directory.

[Line 1 \(page 283\)](#) lists the reset point.

Lines 2+ data displays as follows.

RegionName	Name of the region(s) where the items reside; separate multiple regions by commas.
------------	--

Item Class	Item class of the items to be included.
AttribName	Name of the attribute whose value is to be compared.
StartChar	Starting character within the attribute value. (Allows comparison of substrings)
Length	Number of characters to include in the comparison.
Operator	The type of the comparison to be performed between the attribute and the user specified value (==,! =,>,>=,<,<=,%(exists),\$(contains),@(between),!@).
Value	Value to be compared against attribute value. For alphanumeric comparisons, this value can contain the wildcard character "?".For @, !@ operators, separate two (2) values with a comma.
Compare type	Type of Comparison, N (Numeric integer),A (Alphanumeric) and F (real). Attribute will be converted to this type before comparison.
Point Id	Id of CIMPLICITY analog non-array point which will hold the count.

Following is a listing of a sample configuration file.

Sample configuration file:

```
| - *
PRTCNT_RESET
*Multiple Region Counts
SRC1,SRC2,DST1,DST2|VEHICLE|COLOR|0|10|!=|BLUE|A|BIG_CNT1
SRC1,SRC2,DST1,DST2|CARRIER|CYCLE_CNT|0|10|!=|2|F|BIG_CNT2
* Region SRC2
SRC2|CARRIER|CYCLE_CNT|0|10|==|2|F|SRC2_CNT1
SRC2|CARRIER|CYCLE_CNT|0|10|!=|2|F|SRC2_CNT2
SRC2|CARRIER|CYCLE_CNT|0|10|>|2|F|SRC2_CNT3
SRC2|CARRIER|CYCLE_CNT|0|10|>=|2|F|SRC2_CNT4
SRC2|CARRIER|CYCLE_CNT|0|10|<|2|F|SRC2_CNT5
SRC2|CARRIER|CYCLE_CNT|0|10|<=|2|F|SRC2_CNT6
SRC2|CARRIER|CYCLE_CNT|0|10|$|2|F|SRC2_CNT7
SRC2|CARRIER|CYCLE_CNT|0|10|%||F|SRC2_CNT8
SRC2|CARRIER|CYCLE_CNT|0|10|@|2,3|F|SRC2_CNT9
SRC2|CARRIER|CYCLE_CNT|0|10|!@|2,3|F|SRC2_CNT10
* Region DST1
DST1|VEHICLE|COLOR|0|10|==|RED|A|DST1_CNT1
DST1|VEHICLE|COLOR|0|10|>|RED|A|DST1_CNT3
DST1|VEHICLE|COLOR|0|10|<|RED|A|DST1_CNT5
DST1|VEHICLE|COLOR|0|10|$|R|A|DST1_CNT7
DST1|VEHICLE|COLOR|0|10|%||A|DST1_CNT8
DST1|VEHICLE|COLOR|0|10|@|BLUE,RED,|A|DST1_CNT9
DST1|VEHICLE|COLOR|0|10|!@|BLUE,RED|A|DST1_CNT10
DST1|VEHICLE|COLOR|0|10|==|??D|A|DST1_CNT11
* Region DST2
DST2|VEHICLE|PRT_ITEM_ID|2|2|==|5|N|DST2_CNT1
DST2|VEHICLE|PRT_ITEM_ID|2|2|!=|5|N|DST2_CNT2
DST2|VEHICLE|PRT_ITEM_ID|2|2|>=|5|N|DST2_CNT4
DST2|VEHICLE|PRT_ITEM_ID|2|2|<|5|N|DST2_CNT5
DST2|VEHICLE|PRT_ITEM_ID|2|2|$|5|N|DST2_CNT7
DST2|VEHICLE|PRT_ITEM_ID|2|2|%|5|N|DST2_CNT8
```

```
DST2 |VEHICLE |PRT_ITEM_ID |2 |2 |@ |5 ,10 |N |DST2_CNT9
DST2 |VEHICLE |PRT_ITEM_ID |2 |2 |!@ |5 ,10 |N |DST2_CNT10
DST2 |VEHICLE |PRT_ITEM_ID |0 |4 |== |D?0? |A |DST2_CNT11
*
```

Count Items: prtcnt.exe

Count Items: prtcnt.exe (Upgrade from CIMPLICITY 6.0 or Earlier)

Prcnt.exe is a tool that enables you to use points to count items and display the count through the Point Control Panel or a CimEdit screen.

If you have upgraded from a CIMPLICITY 6.0 or earlier you may have to enable this new tool.

Two steps to enable prtcnt are:

Step 1 (page 284)	Modify five files in the Master directory.
Step 2 (page 288)	Add a reset point to a PRT Graphic display configuration.

Step 1. Modify Five Files in the Master Directory

Step 1. Modify Five Files in the Project's Master Directory

1. Open the project in the Workbench.
2. Select Tools>Command prompt on the Workbench menu bar.

The Command window opens in the project's root directory.

3. Type `cd master`

4. Press Enter.

The Command prompt is the project's Master sub-directory.

5. Modify the following five master files.

Step 1.1 (page 285)	Add prtcnt to physproc.idt.
Step 1.2 (page 285)	Add prtcnt to logproc.idt.

Step 1.3 (page 286)	Add prtcnt to node_logproc.idt.
Step 1.4 (page 286)	Add prtcnt to service.idt.
Step 1.5 (page 287)	Add prtcnt to prt_intproc.
Step 1.6 (page 287)	Add prtcnt to master.mcp.

Step 1.1. Add prtcnt to physproc.idt

1. Use the [idtpop utility \(page 251\)](#) to open the physproc.idt file in a text editor.
2. Add the following line to the file:

```
MASTER | PRTCNT | PRTCNT | 0
```

New	Add or replace the old line with the following.
	Master PRTCNT PRTCNT 0.
Old	Delete, comment out or edit to create the new line.
	MASTER PRTGRD PRTGRD 0

3. Close the text editor.
4. Type at the Master sub-directory command prompt:

```
scpop physproc
```

The master configuration data will be copied to the runtime data the next time you do a project configuration update in the Workbench.

Step 1.2. Add prtcnt to logproc.idt

1. Use the [idtpop utility \(page 251\)](#) to open the logproc.idt file in a text editor.
2. Add the following line to the file:

```
PRTCNT | RESIDENT | 1 | PRT Graphic Display
```

New	Add or replace the old line with the following.
-----	---

	PRTCNT RESIDENT 1 PRT Graphic Display.
Old	Delete, comment out or edit to create the new line.
	PRTGRD RESIDENT 1 PRT Graphic Display

3. Close the text editor.
4. Type at the Master sub-directory command prompt:

```
scpop logproc
```

The master configuration data will be copied to the runtime data the next time you do a project configuration update in the Workbench.

Step 1.3. Add prtcnt to node_logproc.idt

1. Use the [idtpop utility \(page 251\)](#) to open the node_logproc.idt file in a text editor.
2. Add the following line to the file:

```
MASTER|PRTCNT|BSM_ROOT:[exe]prtcnt.exe|20|1|1|80|50|50000|0|50|2000|100|
50000|-
10|50|99|999|9999|-1
```

New	Add or replace the old line with the following.
	MASTER PRTCNT BSM_ROOT:[exe]prtcnt.exe 20 1 1 80 50 50000 0 50 2000 100 50000 - 10 50 99 999 9999 -1.
Old	Delete, comment out or edit to create the new line.
	MASTER PRTGRD BSM_ROOT:[exe]prtcnt.exe 20 1 1 80 50 50000 0 50 2000 100 50000 - 10 50 99 999 9999 -1, if it is in the file.

3. Close the text editor.
4. Type at the Master sub-directory command prompt:

```
scpop node_logproc
```

The master configuration data will be copied to the runtime data the next time you do a project configuration update in the Workbench.

Step 1.4. Add prtcnt to service.idt

1. Use the [idtpop utility \(page 251\)](#) to open the service.idt file in a text editor.

2. Add the following line to the file:

```
PRTCNT | PRT | PRTCNT
```

New	Add or replace the old line with the following.
	PRTCNT PRT PRTCNT.
Old	Delete, comment out or edit to create the new line.
	PRTGRD PRT PRTGRD

3. Close the text editor.
4. Type at the Master sub-directory command prompt:

```
scpop service
```

The master configuration data will be copied to the runtime data the next time you do a project configuration update in the Workbench.

Step 1.5. Add prtcnt to prt_intproc

1. Use the [idtpop utility \(page 251\)](#) to open the prt_intproc.idt file in a text editor.
2. Add the following line to the file:

```
PRT_DC | PRTCNT | 1
```

New	Add or replace the old line with the following.
	PRT_DC PRTCNT 1.
Old	Delete, comment out or edit to create the new line.
	PRT_DC PRTGRD 1

3. Close the text editor.
4. Type at the Master sub-directory command prompt:

```
scpop prt_intproc
```

The master configuration data will be copied to the runtime data the next time you do a project configuration update in the Workbench.

Step 1.6. Add prtcnt to master.mcp

1. Use the [idtpop utility \(page 251\)](#) to open the master.mcp file in a text editor.
2. Add `prtcnt` to the list in the file.

New	Add <code>PRTCNT</code> .
Old	Delete <code>PRTGRD</code>

3. Close the text editor.
4. Exit the Command window.

Step 2. Add a Reset Point to a PRT Graphic Display Configuration

1. Open the Tracker window.
2. Double-click Graphics Display.
3. Do one of the following:

Option 1

Double-click an existing graphic display configuration in the right pane to open the configuration's PRT Graphic Display Configuration dialog box.

Option 2

Click the New button in the right pane to open a new PRT Graphic Display Configuration dialog box.

4. Fill in or edit the fields on the General tab.

The Display point will count the items according to the specifications entered on this tab.

5. Select the Global tab.
6. Specify a reset point.

Updating the reset point forces the counters to recount the number of items that contain the specified attributes in the selected region

PRT Synchronization and Alarm Move Exceptions

PRT Synchronization

The **auto_move_point** field in the PRT_REGION file supplies the ID of a text point to be set when an item moves into a tracking region due to a point-based automatic move.

This point is set according to the value of the num_setpt_param parameter in the global parameters file.

The format for the num_setpt_param global parameter is:

NUM_SETPT_PARAM | 3 | <number>

Where:

If the	the point will be set with the
Parameter is not present	Item ID.
Or	
< number > is set to	
1	Item ID.
2	Item ID and Type
3	Item ID, Type, and Reference ID.
4	Item ID, Type, Reference ID, and Source Region.

Alarm Move Exceptions

In order to generate the **Undefined Source Region** alarm, the global parameter, **PRT_ALARM_MOVE_EXCEPTION**, must be defined in the global parameters file as follows:

PRT_ALARM_MOVE_EXCEPTION|1|YES

PRT Interested Process Utility

PRT Interested Process Utility: prt_inttest.exe

prt_inttest.exe is a PRT Interested Process Utility that is used to ensure that the Tracker system is [generating interested processes \(page 289\)](#).

The [utility is configured \(page 290\)](#) using a combination of CIMPLICITY and PRT files.

The output generated by the PRT Interested Process Utility can be viewed in a Command window from the CIMPLICITY Workbench.

PRT Interested Process Utility Output

The following output can be generated for the PRT interested process.

All files are stored in the project's master directory and can be edited using the [IDTPOP Utility \(page 251\)](#) to convert records to ASCII-text format.

Item Update	
reg_entry_time	parent_item_id
event_code	item_status
modify_bitmask	int_hold_active
region_1_id	item_hold_reason
region_1_loc	ext_hold_active
region_2_id	group_id
region_2_loc	num_atts_valid
item_tracking_type	item_att_list
item_id	comment
reference_id	user_or_svc_id
item_type_id	action
Region Update	Error Response
event_code	status
region_id	err_code
region_status	err_ref
comment	err_msg
user_or_svc_id	
location	
action	

PRT Interested Process Utility Configuration

PRT Interested Process Utility Configuration

The six files needed to configure the PRT Interested Process Utility (**prt_inttest.exe**) can be found in the Master directory of CIMPLICITY. The files are in binary format and must be converted to ASCII text using the IDTPOP Utility. Also, files that have overlapping fields (for example, process_id) must have identical values in order to run the utility. Sample configuration files are provided for your reference in this section.

The required configuration files are:

- physproc.dat
- logproc.dat
- node_logproc.dat
- service.dat
- prt_intproc.dat

The following steps provide instruction on configuring prt_inttest.exe:

Step 1 (page 291)	Open Tracker project in Workbench.
Step 2 (page 292)	Configure physproc.
Step 3 (page 292)	Configure logproc.
Step 4 (page 293)	Configure node_logproc.
Step 5 (page 293)	Configure service.
Step 6 (page 294)	Configure prt_intproc.
Step 7 (page 294)	Run the interested process.

Step 1. Open a Tracker Project in the Workbench

1. Click **Start** on the Windows task bar.
2. Select (All) Programs>Proficy HMI SCADA - CIMPLICITY version>Workbench.

The Workbench window opens.

3. Click File on the Workbench menu bar.
4. Select Open.

The Open dialog box opens.

5. Find and select the Tracker project.

- Click **Open** to open the project in the Workbench window.

i Tip: To open a project quickly, create a shortcut for the project on your Windows desktop and double-click the icon.

Step 2. Configure Physproc

- Use the [IDTPOP utility \(page 251\)](#) to convert the file to ASCII-text using the text editor of your choice, e.g. Notepad.
- Add a line that contains the following fields.

```
node_id | process | id | object_name | order
```

Field No.	Field Name	Description	Example
0	node_id	Node name, always MASTER.	MASTER
1	process_id	The name of the process that the program will look to for data.	PRT_INT
2	object_name	Object name, same as Process ID.	PRT_INT
3	order	Use default value = 0.	0

```
MASTER | PRT_INT | PRT | INT | 0
```

- Save the file.
- Use the SCPOP Utility to convert file back to binary format.

Step 3. Configure Logproc

- Use the [IDTPOP Utility \(page 251\)](#) to convert the file to ASCII-text using the text editor of your choice, e.g. Notepad.
- Add a line that contains the following fields.

```
process_id | process_type_id | pm_flags | description
```

Field No.	Field Name	Description	Example
0	process_id	The name of the process that the program will look to for data.	DL_RP
1	process_type_id	Process type ID; usually RESIDENT process.	RESIDENT
2	pm_flags	Use default value = 1.	1
3	description	Description of the process	Data Logging

Example

```
DL_RP|RESIDENT|1|Data Logging
```

3. Save the file.
4. Use the SCPOP Utility to convert the file back to binary format.

Step 4. Configure Node logproc

1. Use the [IDTPOP Utility \(page 251\)](#) to convert the file to ASCII-text using the text editor of your choice, e.g. Notepad.
2. Add a line that contains the following fields.

```
node_id|process_id|image_name|20|1|1|80|50|50000|0|50|2000|100|50000|10|-  
50|99|999|9999|-1
```

Field No.	Field Name	Description	Entry Example
0	node_id	Node name, always MASTER.	MASTER
1	process_id	The name of the process that the program will look to for data.	AMRP
2	image_name	The name of the executable image. Configure as follows: Directory_name:[exe]prt_inttest.exe	BSM_ROOT:[EXE] AMRP.EXE

Use the default values for the remaining fields.	20 1 1 80 50 50000 0 50 2000 100 50000 10 -50 99 999 9999 -1
--	--

Example

```
MASTER|AMRP|BSM_ROOT:[exe]amrp.exe|20|1|1|80|50|50000|0|50|2000|100|50000|  
10|-50|99|999|9999|-1
```

3. Save the file.
4. Use the SCPOP Utility to convert file back to binary format.

Step 5. Configure Service

1. Use the [IDTPOP Utility \(page 251\)](#) to convert the file to ASCII-text using the text editor of your choice, e.g. Notepad.
2. Add a line that contains the following fields.

```
service_id|subsys_id|process_id
```

Field No.	Field Name	Description	Entry Example
0	service_id	The name given to this service.	PRT_INT
1	subsys_id	The ID of the subsystem, in this case PRT.	PRT
2	process_id	The name of the process that the program will look to for data.	PRT_INT

Example

```
PRT_INT|PRT|PRT_INT
```

3. Save the file.
4. Use the SCPOP Utility to convert file back to binary format.

Step 6. Configure Prt_intproc

1. Use the [IDTPOP Utility \(page 251\)](#) to convert the file to ASCII-text using the text editor of your choice, e.g. Notepad.
2. (If the line you need is not in the file) add a line that contains the following fields.

```
prt_svc_id|intproc_svc_id|update_mask
```

Field No.	Field Name	Description	Example
0	prt_svc_id	Unique identifier of a PRT service that will provide interested process updates.	PRT_DC
1	intproc_svc_id	The name given to this service. This must be identical to the ID in service.dat .	TRK_RP
2	update_mask	Specifies the type of updates to be supplied to the interested process. Values are 1=Full; 0=Brief	1

Example

```
PRT_DC|TRK_RP|1
```

3. Save the file.
4. Use the SCPOP Utility to convert file back to binary format.

Step 7. Run the PRT Interested Process Utility

Once all of the appropriate files have been configured, you can run the `prt_inttest.exe` and view the output in a Command window.

1. (If running) stop your project in the Workbench.
2. Click on the Workbench toolbar to perform a Configuration Update.
3. Click on the Workbench toolbar to start your project
4. Click Tools on the menu bar.
5. Select Command Prompt

The Command window opens.


6. Type **Set PRCNAM=PRT_INT** on the command line.

Where

PRT_INT is the process ID used in your configuration files.

7. Press **Enter**.
8. Type **PRT_INTTEST.EXE** on the command line..
9. Press **Enter**.

The interested process messages generated by PRT operations will output in the Command window.

 **Important:** In order to run the executable file, match the letter case of the arguments on the Command lines.

Production Tracking Export Program: PRT_EXPORT

Production Tracking Export Program: PRT_EXPORT

`prt_export`, a Production Tracking export program that has a set of export record types and corresponding record formats.

Review:

	Export input file format.
	Export output file format.
	Run the export program.

Export Input File Format: prtexp.prtx

The Export Program reads in one or more records from an input file. These records specify which groups and regions of the specified service are to be dumped (all information associated with the groups and/or regions specified, and complete information on all the items within the group or region, are written to the output file).

The first field in each record in the Export file specifies the command. These commands are described by denoting the literal characters to be placed in the first field of each input file record. Subsequent fields in the record are delimited by a field delimiter character (usually |) which is defined on the very first line of the Export file.

The delimiter, continuation, and comment character specified in prt_export's input file are the same ones used to form an output file.

Commands are:

Command	Function
ALL (page 296)	Dumps all tracking data.
DSP (page 296)	Prints to the terminal.
GRP (page 297)	Dumps all tracking data in a specified Group.
REG (page 297)	Dumps all tracking data in a specified Region.
SVC (page 297)	Dumps all tracking data for a specified service or services.

ALL

Specifies that everything is to be dumped for all services. This record should be the first (and only) record in the file. If there are any subsequent records in the file, they will be ignored, i.e. processing will terminate upon completion of the complete dump of everything. If the ALL appears after some other record type, it will be ignored.

DSP

Prints out the fields following the DSP field (one field per line). For example:

DSP|Production start for item XYZ

GRP

Specifies the group identifier of a group to be dumped. This record causes all information on all regions and items within the group to be dumped. The format for the record is:

```
GRP|group_identifier
```

or

```
GRP|group_id_1|group_id_2...
```

The limit on the number of group identifiers that can be specified within a single GRP record is limited by the IDT routines that read and parse the input records. The current limits are 100 fields per record and 1000 characters per record.

REG

Specifies the regions (by region identifier) which are to be dumped. The format is:

```
REG|region_identifier
```

Or

```
REG|region_id_1|region_id_2...
```

The limit on the number of region identifiers that be specified within a single REG record is limited by the IDT routines that read and parse the input records. The current limits are 100 fields per record and 1000 characters per record.

SVC

Specifies the service ID of the PRT service to be dumped. This record causes all information on all groups, regions, and items within the service to be dumped. The format for the record is:

```
SVC|service_identifier
```

Or

```
SVC|svc_id_1|svc_id_2...
```

The limit on the number of service identifiers that can be specified within a single SVC record is limited by the IDT routines that read and parse the input records. The current limits are 100 fields per record and 1000 characters per record.

Export Output File Format: prtexp.prtm

The format for this file is identical to the format of the Import program's [Input file \(page 301\)](#) .

Note: A region may be a member of more than one group. Thus, the dumping of two different groups could include the same region twice. However, this does not result in the region being dumped each time it is encountered (either as a member of a different group or because the region was explicitly dumped).

The first time the region is encountered, all information on it is dumped (the region status and information on all items within the region). Subsequent encounters with the same region result in a comment record being written into the output file specifying the region name and indicating that the region had previously been dumped and appears earlier in the output file.

The same holds true for all similar circumstances, i.e. a service is dumped and then a group or region within that service is explicitly dumped (or vice versa).

No groups or regions will be dumped more than once.

All subsequent appearances of a group or region will be indicated by the insertion of a comment record in the output file.

Run the Export Program

1. Open the project in the Workbench.
2. Make sure the project is running.
3. Open a Command window.
4. Type the following command.

```
notepad prtexp.prtx
```

Where

`notepad` is the text editor

`prtexp.prtx` is the name of the file you create.

Note:

You can:

- Give the file any name. However, you must give it a .prt extension.
- Open the Command window from a command prompt. If you do, make sure you copy the export file into the execution path.

A message box will open asking if you want to create a new file.

5. Click Yes.
6. Enter one or more [commands \(page 296\)](#) in the file.

Note: If the file is empty you will receive an error message when you try to run it.

7. Enter the following command to run the program:

PRT_EXPORT < filename >. **PRTX** < prt_system > < prt_node >

Where

< filename >. **PRTX** is an input file that specified what is to be dumped to the export file

< prt_system > is the prefix of the PRT data server you wish to export from. Specify **ALL** if all data servers should be connected.

< prt_node > is the node where the PRT servers are running. Specify **LOCAL** if they are on the same node.

If you do not specify any of the parameters on the command line, you will be prompted.

Example Input File

|-*

*

REG|R1_BOX

REG|R2_CAB

REG|R3_CAR

Result:

The program will create an output file named < filename >. **PRTM**

Where

< filename > is the same as that of the input file.

A sample output file is provided in the section Running the Import Program.

Production Tracking Import Program: PRT_IMPORT

Production Tracking Import Program: PRT_IMPORT

`prt_import` is a Production Tracking import program that has a set of Import record types and corresponding record formats.

	Program elements (Production Tracking Import program).
	Import file naming conventions.
	Import file format.
	Run the import program.

Program Elements: Production Tracking Import Program

The Production Tracking Import Program interprets a file to determine the commands to issue. It also provides information about the results of the commands issued.

- Process Registration
- Initialization of the Production Tracking Application Interface
- Determination of the name of the Import data file to be used. The name of the input file is specified on the command line, (for example, **prt_import** < filename >). The input file is assumed to have **. prtm** as a filename extension. If a filename is not specified on the command line, the program prompts for it.
- Opening of the Import data file
- Reading of Import file records
- Parsing of Import file records to determine applicable PRT Application Interface routine to call
- Further parsing of Import file record to obtain the arguments to be included to the PRT Application Interface call
- Call to appropriate PRT Application Interface routine
- Reporting of any errors returned by the Application Interface.
- Termination of the Application Interface
- Reporting of "command" totals for each type of command, total records processed, etc.

Import File Naming Conventions

The file naming conventions are:

Input Files	<filename>. PRTX
Output Files	<filename>. PRTM

Where

< filename > for input is used to create output.

*Import File Format**Import File Format*

The Import program reads a series of records from the Import file. Each record begins with a 3-character command.

Valid commands are as follow:

Command	Function
ADD <i>(page 302)</i>	Add item to be tracked.
ADV <i>(page 303)</i>	Advance item along specified route.
AEA <i>(page 303)</i>	Add item attribute (extended).
ATT <i>(page 304)</i>	Add item attribute (standard).
CLL <i>(page 304)</i>	Clear location.
DEL <i>(page 305)</i>	Delete tracked item
DSP <i>(page 305)</i>	Print message to terminal.
HLD <i>(page 306)</i>	Apply group hold to item.
INS <i>(page 306)</i>	Insert item to be tracked.
MAT <i>(page 307)</i>	Modify item attribute.
MOD <i>(page 307)</i>	Modify tracked item.
MOV <i>(page 308)</i>	Move item from source region to destination region.

PTS (page 309)	Enable/disable (1/0) production setpoints.
REG (page 309)	Modify region status.
STT (page 310)	Initiate a production start.
STP (page 310)	Initiate a production stop.

Subsequent fields in the record are delimited by a field delimiter character (usually |) defined on the very first line of the Import file. A blank line in the import input file causes the Import program to terminate processing of the input file at that point.

All fields in the file are of type character. The actual data passed to PRT is converted to the correct type by the Import program.

- For the DELete command the disposition can be specified as "SCRAP", "DEL", or "DELETE". These strings are translated to the manifest constants SCRAP and DELETE.
- For any API functions that require a region location argument (e.g. the ADD and INSert commands) the region location can be specified as "ALL", "LAST" ("ENTRY"), or "FIRST" ("EXIT"). These are translated to the manifest constants ALL_LOCATIONS, LAST and FIRST, respectively.
- For arguments that are required to be a Boolean value, the strings "T", "TRUE", "F", or "FALSE" can be specified, which translate to the manifest constants TRUE and FALSE.

ADD (Add an Item to be Tracked)

This command provides a mechanism to add an individual Item to a location in which another Item currently resides.

The fields in this record are:

Field	Description
char record_type[3]	ADD
COR_BOOLEAN ext_proc_flag	If TRUE, get all info from ext process for serialized item
char region_id[17]	Region where the item is to be added to the queue or production start is to begin
COR_I2 region_loc	Specific location in the region where the Item is to be added
char item_id[21]	Unique identifier of a serialized item
char reference_id[21]	Secondary identifier of a serialized item
char item_type_id[17]	Item type identifier

char parent_item_id[21]	Associated serialized item identifier
COR_U4 item_status	Item status
COR_BOOLEAN ext_hold_active	If TRUE, item held due to external hold.
char group_id[17]	Identifier of the Tracking Group providing the reason for the hold.
COR_BOOLEAN int_hold_active	If TRUE, item held due to internal hold.
char int_hold_reason[41]	Comment specifying reason for internal hold.
char comment[41]	Comment to be recorded in prt history log file.
char user_or_svc_id[33]	User or service identifier

If no Item Attribute information is to be specified for this item, the ADD record need not be preceded by any ATT records (for filling in elements within the Item Attribute array). To specify Item Attribute information, see the description of the ATT record type.

ADV (Advance an Item Along a Specified Route)

The Application Interface call for this record type provides a mechanism for advancing the next Item Carrier to leave a Source Tracking Region to a Destination Tracking Region. This call moves ALL items at the FIRST (EXIT) location in the source region to the LAST (ENTRY) location in the destination region.

The fields in this record are:

Field	Description
char record_type[3]	ADV
char src_region_id[17]	Region the item is moving out of
char dest_region_id[17]	Region the Item is moving into
char comment[41]	Comment to be recorded in PRT history log file
char user_or_svc_id[33]	User or service identifier

AEA (Add an Extended Item Attribute)

This record type provides a means of filling in the array of Item Attribute structures (one element at a time) the program maintains internally, for providing as an argument to the PRT Application Interface functions invoked by the STT, ADD, INS, and MAT record types.

Important : If [trkattribute.cfg \(page 314\)](#) is configured to store extended attributes in a database, instead of backing files, do not use this option. See Tracker Extended Attribute Configuration for more information on extended attribute storage options.

The fields in this record are:

Field	Description
char record_type[3]	AEA
char att_name[40]	Extended Attribute name
char att_value[255]	Extended Attribute value

ATT (Add an Item Attribute)

This record type provides a means of filling in the array of Item Attribute structures (one element at a time) the program maintains internally, for providing as an argument to the PRT Application Interface functions invoked by the STT, ADD, INS, and MAT record types.

The Item Attribute structure is:

```
typedef struct
{
    char att_name[17];
    char att_value[17];
} PRT_ITEM_ATT;
```

The fields in this record are:

Field	Description
char record_type[3]	ATT
char att_name[17]	Attribute name
char att_value[17]	Attribute value

The array of **PRT_ITEM_ATT** structures has **PRT_NUM_ATTRIBUTES** elements. This is the maximum number of ATT records that can appear in the Import file without an intervening **STT, ADD, INS, or MAT** record (which sends whatever Item Attribute data is currently in the array and then clears it out for the next Import command (API call) that uses it). More than **PRT_NUM_ATTRIBUTES** ATT records without any intervening STT, ADD, INS, or MAT records will result in an error being reported to the user.

CLL (Clear Location)

The Application Interface call for this record type provides a mechanism for deleting or scrapping all items at a given location in a specific Tracking Region.

The fields in this record are:

Field	Description
char record_type[3]	CLL
char region_id[17]	Identifier of the Tracking Region where the Item is to be deleted from the queue.

char region_loc	Identifier of the Tracking location where the Item is to be deleted from the queue.
COR_I1 disposition	Manifest constant specifying the disposition of the item (which can be specified as 'DEL', 'DELETE' or 'SCRAP').
char comment[41]	Comment to be recorded in prt history log file
char user_or_svc_id[33]	User or service identifier

DEL (Delete a Tracked Item)

The Application Interface call for this record type provides a mechanism for deleting or scrapping a single Item from all configured PRT services, a specific PRT service, a specific Tracking Group or a specific Tracking Region.

The fields in this record are:

Field	Description
char record_type[3]	DEL
char prt_svc_id[33]	PRT service identifier
char group_id[17]	Identifier of the Tracking Group
char region_id[17]	Identifier of the Tracking Region where the Item is to be deleted from the queue
COR_I2 region_loc	Region location where the item resides
char item_id[21]	Unique identifier of a serialized item
Reference_id[21]	Secondary identifier of a serialized item
char item_type_id[17]	Item type identifier
COR_I1 disposition	Manifest constant specifying the disposition of the item (which can be specified as 'DEL', 'DELETE' or 'SCRAP').
char comment[41]	Comment to be recorded in prt history log file
char user_or_svc_id[33]	User or service identifier

DSP (Print Message to Terminal)

This record type provides a means of printing a message out on the terminal. Messages can be inserted in the Import input file to notify the user as to the current point of processing within the file.

The fields in this record are:

Field	Description
char record_type[3]	DSP

char display_string	Message to be displayed on terminal. The message can be as long as necessary.
----------------------------	---

HLD (Apply a Group Hold to an Item)

The application interface call for this record type provides a mechanism to activate and deactivate a HOLD specification for a Tracking Group.

The fields in this record are:

Field	Description
char record_type[3]	HLD
char group_id[17]	Unique identifier of tracking group
char ext_hold_reason[41]	Text describing the reason for placing holds on Tracking Regions in this Tracking Group (optional for activate, does not apply to deactivate)
COR_BOOLEAN act_hold_flag	If set to TRUE, the Hold Specification is activated. If set to FALSE, the Hold Specification is Deactivated
char comment[41]	Comment to be recorded in PRT history log file
char user_or_svc_id[33]	User or service identifier

INS (Insert an Item to be Tracked)

This command provides a mechanism for inserting an Item into the Tracking Region queue. If an Item is to be inserted into the queue at a particular location (other than at the head or tail of the queue), the region location of the Item it is to be inserted ahead of is specified (i.e. specifying 'region_loc = 1', is equivalent to specifying 'region_loc = FIRST'). Defined constants are provided to indicate that the item is to be added either at the head or at the tail of the queue. The region location can be specified as FIRST (or also as EXIT) to insert the Item as the first (head) Item in the queue (i.e. this Item will be the next Item to transition out of the queue). LAST and ENTRY are provided to insert the Item as the last Item at the tail of the queue, BEHIND all other Items.

The fields in this record are:

Field	Description
char record_type[3]	INS
COR_BOOLEAN ext_proc_flag	If TRUE, get all info from ext process for serialized item
char region_id[17]	Region where the item is to be added to the queue or production start is to begin
COR_I2 region_loc	Location of the Item that the inserted item is to be inserted in front of (nearer to exiting the queue).
char item_id[21]	Unique identifier of a serialized item
char reference_id[21]	secondary identifier of a serialized item

char item_type_id[17]	Item type identifier
char parent_item_id[21]	associated serialized item identifier
COR_U4 item_status	Item status
COR_BOOLEAN ext_hold_active	If TRUE, item held due to external hold.
char group_id[17]	Identifier of the Tracking Group providing the reason for the hold.
COR_BOOLEAN int_hold_active	If TRUE, item held due internal hold.
char int_hold_reason[41]	Comment specifying reason for internal hold.
char comment[41]	Comment to be recorded in prt history log file.
char user_or_svc_id[33]	User or service identifier

If no Item Attribute information is to be specified for this Item, the INS record need not be preceded by any ATT records (for filling in elements within the Item Attribute array). To specify Item Attribute information, see the description of the ATT record type.

MAT (Modify an Item's Attributes)

The application interface call for this record type provides a mechanism for modifying Serialized Item Attribute data.

The fields in this record are:

Field	Description
char record_type[3]	MAT
char region_id[17]	Unique identifier of the tracking region where the item resides
char item_id	Unique identifier of a Serialized Item
char reference_id[21]	Secondary identifier of a Serialized Item
char comment[41]	Comment to be recorded in prt history log file.
char user_or_svc_id[33]	User or service identifier

The purpose of this call is to modify an item's attributes, the MAT record must be preceded by one or more ATT records (for filling in elements within the Item Attribute array). To specify item attribute information, see the description of the ATT record type.

MOV (Move an Item from a Source to a Destination Region)

The Application Interface call for this record type provides a mechanism to move an Item from the Tracking Region where it currently resides to a new Tracking Region. It also permits the movement of an individual Item from one location to another within the same Tracking Region.

The fields in this record are:

Field	Description
char record_type[3]	MOV
char src_region_id[17]	Region the item is moving out of
char dest_region_id[17]	Region the Item is moving into
COR_I2 src_region_loc	Sequence number of the item in the source tracking region
COR_I2 dest_region_loc	Sequence number of the item in the destination tracking region
COR_BOOLEAN dest_insert_flag	If TRUE, the moved Item is inserted in the destination Tracking Region queue ahead of the Item located at the specified destination region location.
char item_id[21]	Unique identifier of a Serialized Item
char reference_id[21]	secondary identifier of a Serialized Item
char item_type_id[17]	Item type identifier
char comment[41]	Comment to be recorded in PRT history log file
char user_or_svc_id[33]	User or service identifier

MOV (Move an Item from a Source to a Destination Region)

The Application Interface call for this record type provides a mechanism to move an Item from the Tracking Region where it currently resides to a new Tracking Region. It also permits the movement of an individual Item from one location to another within the same Tracking Region.

The fields in this record are:

Field	Description
char record_type[3]	MOV
char src_region_id[17]	Region the item is moving out of
char dest_region_id[17]	Region the Item is moving into
COR_I2 src_region_loc	Sequence number of the item in the source tracking region
COR_I2 dest_region_loc	Sequence number of the item in the destination tracking region

COR_BOOLEAN dest_insert_flag	If TRUE, the moved Item is inserted in the destination Tracking Region queue ahead of the Item located at the specified destination region location.
char item_id[21]	Unique identifier of a Serialized Item
char reference_id[21]	secondary identifier of a Serialized Item
char item_type_id[17]	Item type identifier
char comment[41]	Comment to be recorded in PRT history log file
char user_or_svc_id[33]	User or service identifier

PTS (Enable/Disable Region Setpoints)

The Application Interface call for this record type provides a mechanism to enable and disable the Item transition setpoints configured in the PRT Tracking Route records. These points are set by PRT when an Item transitions into the configured destination region.

The fields in this record are:

Field	Description
char record_type[3]	PTS
char region_id[17]	Tracking region identifier
COR_BOOLEAN enable_point_flag	If TRUE, enable region item transition setpoint. If FALSE, disable region item transition setpoint.
char comment[41]	Comment to be recorded in PRT history log file
char user_or_svc_id[33]	User or service identifier

REG (Modify Region Status)

The application interface call for this record type provides a mechanism for modifying Tracking Region Data; specifically, the region status.

The fields in this record are:

Field	Description
char record_type[3]	REG
COR_U4 status_bitmask	Indicates field that has been modified
char region_id[17]	Unique identifier of the tracking region
COR_U4 region_status	Tracking Region status code (e.g. IN-LOCKED, OUT-LOCKED, FULL, OUT-OF-SEQ)
char comment[41]	Comment to be recorded in prt history log file

char user_or_svc_id[33]	User or service identifier
--------------------------------	----------------------------

STP (Initiate a Production Stop)

The application interface call for this record type provides a mechanism for removing an Item from the Production Tracking system as the result of its exiting out of a region along a route that has no destination region. The specified Item must currently reside at a source region associated with such a route. If the Item cannot legitimately transition out of the system along such a route, an error is returned.

The fields in this record are:

Field	Description
char record_type[3]	STP
char region_id[17]	Identifier of the Tracking Region where the Item to exit the system is located
COR_I2 region_loc	Specific location in the region where the Item is to be added
char item_id[21]	Unique identifier of a serialized item
char reference_id[21]	Secondary identifier of a serialized item
char item_type_id[17]	item type identifier
char comment[41]	Comment to be recorded in prt history log file
char user_or_svc_id[33]	User or service identifier

STT (Initiate a Production Start)

Provides a mechanism to initiate a production start for an item in a tracking region.

The fields in this record are:

Field	Description
char record_type[3]	STT
BOOLEAN ext_proc_flag	If TRUE, get all info from ext process for serialized item
char region_id[17]	Region where the item is to be added to the queue or production start is to begin
char item_id[21]	Unique identifier of a serialized item
char reference_id[21]	Secondary identifier of a serialized item
char item_type_id[17]	Item type identifier
char parent_item_id[21]	Associated serialized item identifier
COR_U4 item_status	Hexadecimal representation of desired item status
COR_BOOLEAN ext_hold_active	If TRUE, item held due to external hold.

char group_id[17]	identifier of the Tracking Group providing the reason for the hold.
COR_BOOLEAN int_hold_active	If TRUE, item held due to internal hold.
char int_hold_reason[41]	Comment specifying reason for internal hold.
char comment[41]	Comment to be recorded in prt history log file.
char user_or_svc_id[33]	User or service identifier

If no Item Attribute information is to be specified in the production start for an item, the STT record need not be preceded by any ATT records (for filling in elements within the Item Attribute array). To specify item attribute information, see the description of the ATT record type.

Run the Import Program

1. Open the project in the Workbench.
2. Make sure the project is running.
3. Open a command window.
4. Enter the following command at the project's Master directory command prompt.

Note: Make sure the import file *.prtm (that was created when the [export program \(page 298\)](#) was run) is in the execution path.

5. Enter the following command to run the program:

```
PRT_IMPORT <filename>.PRTM <prt_system> <prt_node>
```

Where

<filename>.PRTM is the import file

<prt_system> is the prefix of the PRT data server you wish to import to. Specify **ALL** if all data servers should be connected.

<prt_node> is the node where the PRT servers are running. Specify **LOCAL** if they are on the same node.

If you do not specify any of the parameters on the command line, you will be prompted.

Example Import File

```
| - *
*
*   Region: R1_BOX
*
REG|FFFF|R1_BOX|0|PRT_EXPORT|
```

```


INS | F | R1_BOX | 1 | | | REARBOX | | 38 | F | | F | | PRT_EXPORT |
INS | F | R1_BOX | 2 | | | REARBOX | | 38 | F | | F | | PRT_EXPORT |
INS | F | R1_BOX | 3 | | | REARBOX | | 38 | F | | F | | PRT_EXPORT |
INS | F | R1_BOX | 4 | | | REARBOX | | 38 | F | | F | | PRT_EXPORT |
INS | F | R1_BOX | 5 | | | REARBOX | | 38 | F | | F | | PRT_EXPORT |
*
*   Region: R2_CAB
*
REG | FFFF | R2_CAB | 0 | PRT_EXPORT |
ATT | STYLE | LX
ATT | COLOR | ORANGE
ATT | A/C | YES
ATT | ENG TYPE | V6
ATT | BRAKES | POWER
INS | F | R2_CAB | 1 | C3294190 | CT7865 | CAB | | 35 | F | | F | | PRT_EXPORT |
ATT | STYLE | LX
ATT | COLOR | WHITE
ATT | A/C | YES
ATT | ENG TYPE | V6
ATT | BRAKES | MANUAL
INS | F | R2_CAB | 2 | C4322290 | CT8534 | CAB | | 35 | F | | F | | PRT_EXPORT |
ATT | STYLE | SX
ATT | COLOR | RED
ATT | A/C | NO
ATT | ENG TYPE | V8
ATT | BRAKES | POWER
INS | F | R2_CAB | 3 | C8629390 | CT32119 | CAB | | 35 | F | | F | | PRT_EXPORT |
ATT | STYLE | LX
ATT | COLOR | BLUE
ATT | A/C | YES
ATT | ENG TYPE | V6
ATT | BRAKES | MANUAL
INS | F | R2_CAB | 4 | C7738990 | CT7535 | CAB | | 35 | F | | F | | PRT_EXPORT |
ATT | STYLE | SX
ATT | COLOR | BLACK
ATT | A/C | NO
ATT | ENG TYPE | V8
ATT | BRAKES | MANUAL
INS | F | R2_CAB | 5 | C4296590 | CT7008 | CAB | | 35 | F | | F | | PRT_EXPORT |
ATT | STYLE | SX
ATT | COLOR | GREEN
ATT | A/C | YES
ATT | ENG TYPE | V8
ATT | BRAKES | POWER
INS | F | R2_CAB | 6 | C2986590 | CT0943 | CAB | | 35 | F | | F | | PRT_EXPORT |
ATT | STYLE | LX
ATT | COLOR | DK BLUE
ATT | A/C | YES
ATT | ENG TYPE | V6
ATT | BRAKES | MANUAL
INS | F | R2_CAB | 7 | C6398290 | CT5476 | CAB | | 35 | F | | F | | PRT_EXPORT |
ATT | STYLE | SX
ATT | COLOR | YELLOW

```

```

ATT | A/C | NO
ATT | ENG TYPE | V8
ATT | BRAKES | MANUAL
INS | F | R2_CAB | 8 | C6884990 | CT2312 | CAB | | 35 | F | | F | | PRT_EXPORT |
ATT | STYLE | LX
ATT | COLOR | RED/WHITE
ATT | A/C | YES
ATT | ENG TYPE | V6
ATT | BRAKES | POWER
INS | F | R2_CAB | 9 | C5185590 | CT5643 | CAB | | 35 | F | | F | | PRT_EXPORT |
ATT | STYLE | LX
ATT | COLOR | GREY
ATT | A/C | YES
ATT | ENG TYPE | V6
ATT | BRAKES | POWER
INS | F | R2_CAB | 10 | C3299890 | CT7885 | CAB | | 35 | F | | F | | PRT_EXPORT |
*
*   Region: R3_CAR
*
REG | FFFF | R3_CAR | 0 | PRT_EXPORT |
ATT | STYLE | SX
ATT | COLOR | RED
ATT | A/C | NO
ATT | ENG TYPE | V8
ATT | BRAKES | POWER
INS | F | R3_CAR | 1 | S2173490 | ST2993 | CAR | | 32 | F | | F | | PRT_EXPORT |
ATT | STYLE | LX
ATT | COLOR | BLUE
ATT | A/C | YES
ATT | ENG TYPE | V8
ATT | BRAKES | MANUAL
INS | F | R3_CAR | 2 | S6427390 | ST6991 | CAR | | 32 | F | | F | | PRT_EXPORT |
ATT | STYLE | SX
ATT | COLOR | GREEN
ATT | A/C | YES
ATT | ENG TYPE | V8
ATT | BRAKES | POWER
INS | F | R3_CAR | 3 | S4883690 | ST4999 | CAR | | 32 | F | | F | | PRT_EXPORT |
ATT | STYLE | SX
ATT | COLOR | YELLOW
ATT | A/C | NO
ATT | ENG TYPE | V6
ATT | BRAKES | POWER
INS | F | R3_CAR | 4 | S1429890 | ST1996 | CAR | | 32 | F | | F | | PRT_EXPORT |
ATT | STYLE | GTI
ATT | COLOR | GREEN
ATT | A/C | NO
ATT | ENG TYPE | TURBO
ATT | BRAKES | SELFADJUST
INS | F | R3_CAR | 5 | S9812390 | ST9998 | CAR | | 32 | F | | F | | PRT_EXPORT |

```

 **Note:** `prt_import` does not transfer region validation statistics data.

Tracker Extended Attributes Configuration (*trkattribute.cfg*)

Important:

- Save a copy of this file. If you install a product upgrade you will have to copy the file with your configuration into the ...\`Proficy CIMPLICITY\RCO\trkattribute.cfg` file.
- All projects must point to the same database for storing extended attributes, ranges and criteria sets.

1. Open ...\`Proficy CIMPLICITY\RCO\trkattribute.cfg` in a text editor.

The default file, with an explanation of the fields, is as follows.

```
|-*
```

```
* Tracker extended attribute configuration
```

```
*
```

```
* 1. Attribute storage bitmask. 2 - backing files, 4 - Database.
```

```
* 2. Database Server Name
```

```
* 3. Database name
```

```
* 4. Database Username
```


```
* 5. Database Password
```

```
* 6. Maximum number of attributes in the backing file
```

```
* Example
```

```
*4|ALBTRK01|TrkEA|sa|sapw|100000
```

2. Enter the correct criteria for your system in the file.
3. Save and close the file.

 **Note:** If you run more than one project on a server, by default they will all use the same Extended Attribute database.

3. Startup Configuration for PRT Client

3. Startup Configuration for PRT Client

In order to have the PRT Client application start when CIMPLICITY starts the following configuration files must be modified.

Each of the following files need to be changed in the Master directory.

Required files are:

Physproc.idt
Logproc.idt
Node_Logproc.idt
Service.idt
Master.mcp.idt

An optional configuration file enables you to use long project\region names (in the form \\Project \Regionname).

The optional files are:

prtclnt_route.cfg
prtclnt_inv_type.cfg

If these files are configured and located in the SITE_ROOT:\data directory you will be able to use the long names.

If these files do not exist PRT will use [prt_route.idt \(page 264\)](#) and [prt_inv_type.idt \(page 259\)](#) files; using these files limits the user to a 16 character field for the project/region combination.

PRT Client Startup: physproc.idt

Add the following line to physproc.idt when you configure the PRT client startup.

```
|-* IDT file generated by IDTPOP utility v1.0
* RECORD: PHYSPROC PROCESS NAMES ON SPECIFIC NODES
*
* 0 NODE_ID          Decnet Node Name
* 1 PROCESS_ID      Logical Process Name
* 2 object_name     Decnet Object Name
* 3 order           Redundant or Allocated Process Order
*
MASTER | PRT_CLNT | PRT_CLNT | 0
```

PRT Client Startup: logproc.idt

Add the following line to logproc.idt when you configure the PRT client startup.

```
|-* IDT file generated by IDTPOP utility v1.0
* RECORD: LOGPROC LOGICAL PROCESS IDENTIFIERS
*
* 0 PROCESS_ID      Process Identifier
* 1 process_type_id Process Type Identifier
* 2 pm_flags        Bits(last 5):(AP)(RSTAL)(LD)(NOAL)(WT)
* 3 description     Process Description
*
```


```
PRT_CLNT|RESIDENT|1|PRT Client
```

PRT Client Startup: node_Logproc.idt

Add the following line to node_logproc.idt when you configure the PRT client startup.

```
|-* IDT file generated by IDTPOP utility v1.0
* RECORD: NODE_LOGPROC PROCESSES ON SPECIFIC NODES
*
* 0 NODE_ID        Decnet Node Name
* 1 PROCESS_ID     Process Identifier
* 2 image_name     Executable Image name
* 3 priority       Process Priority
* 4 base_quan      Base quantity
* 5 max_quan       Max Quantity
* 6 astlm          AST Limit
* 7 biolm          BIO Limit
* 8 bytlim         BYTE Limit
* 9 cpulm          CPU Limit
* 10 diolm         DIO Limit
* 11 enqlm         ENQ Limit
* 12 fillm         Fil Limit
* 13 pgflquota     Page file quota
* 14 prcml         PRCLM (field name is a typo)
* 15 tqelm         TQELM
* 16 wsdefault     Working set extet
* 18 wsquota       Working set quota
* 19 sys_priv_mask System wide process privilege mask
* 20 check_health  checkHealth / NoHealthParmEdit (-1)
* 21 response_period responsePeriod
* 22 restart       restart
* 23 threshold     threshold
* 24 retry_period  retryPeriod
* 25 kill_process  killProcess
* 26 fail_resource failResource
*
```

```
MASTER | PRT_CLNT | BSM_ROOT: [exe]PRTClient.exe | 20 | 1 | 1 | 80 | 50 | 50000 | 0 | 50 | 2000 |
100 | 50000 | -
10 | 50 | 99 | 999 | 9999 | -1 | 0 | 30 | 0 | 3 | 900 | 0 | 0
```

 **Note:** The line above includes the default settings with the Process health configuration turned off.

PRT Client Startup: service.idt

Add the following line to service.idt when you configure the PRT client startup.

```
|-* IDT file generated by IDTPOP utility v1.0
* RECORD: SERVICE SERVICE IDENTIFIERS
*
* 0 SERVICE_ID      Service Identifier
* 1 SUBSYS_ID       Subsystem Identifier
* 2 PROCESS_ID      Process Identifier
*
PRT_CLNT | PRT | PRT_CLNT
```

PRT Client Startup: master.mcp

Add the following line to the master.mcp list when you configure the PRT client startup.

```
PRT_CLNT
```

PRT Client Startup: prtclnt_route.cfg

The layout of prtclnt_route.cfg that is shown here is the same as in the existing configuration files.

```
|-* IDT file generated by IDTPOP utility v1.0
* RECORD: PRT_ROUTE Production Tracking Route
*
* 0 orig_region_id  ID of region item exits
* 1 dest_region_id  ID of region item enters
* 2 item_type_id    ID of type traveling between regions
* 3 transition_ptid ID of transition point between regions
* 4 set_ptid        ID of point to be set on transition
* 5 translation_id  ID to interp type trans indicator
* 6 transition_type_code transition indicator type code (1-8)
* 7 process_first_ptchange Should we process the first point change
*
*A_TP1 | \\PA3\LONGREGIONNAME_T | A1_A3_ITEMID | | 3 | 1
*A_TP1 | \\PA3\LONGREGIONNAME_T | A1_A3_ITEMID | | 4 | 1
*A_TP1 | \\PA3\LONGREGIONNAME_T | A1_A3_ITEMID | | 1 | 1
*A_TP1 | \\PA3\LONGREGIONNAME_T | A1_A3_ITEMID | | 7 | 1
A_TP1 | \\PA3\LONGREGIONNAME_T | A1_A3_ITEMID | | 8 | 1
A_TP1 | \\PA3\B_TP3 | A1_X_B3 | | 5 | 1
```

```
B_TP1 | \\PA2\A_TP2 | B1_A2_ITEMID | | 3 | 1
B_TP1 | \\PA3\A_TP3 | B1_X_A3 | XLATE1 | 6 | 1
```

PRT Client Startup: prtclnt_inv_type.cfg

The layout of prtclnt_inv_type.cfg that is shown here is the same as in the existing configuration files.

```
|-* IDT file generated by IDTPOP utility v1.0
* RECORD: PRT_INV_TYPE Production Tracking Route Invalid Type
*
* 0 orig_region_id   ID of region item exits
* 1 dest_region_id  ID of region item enters
* 2 item_type_id    ID of type invalid between these regions
* 3 transition_ptid ID of transition point between regions
*
A_TP1 | \\PA3\LONGREGIONNAME_T | BODY COMPONENT | A1_A3_ITEMID
```

4. PRT Service and Resident Process Name Change Configuration

! **Important:** The same name cannot be used for more than one Tracker project on the same node for the following.

	Applies to:
Service name	Tracker Base only or Tracker Base/Tracker projects.
Tracker resident process	Tracker Base/Order Execution Mgt. projects

In this section the following names are being used for example purposes. You can use any name that complies with your system guidelines.

Default	Example new name
PRT_DC	ALPRT1_DC
PRT_DS	ALPRT1_DS
TRK_RP	ALTRK1_RP. Order Execution Mgt. only

5. PRT Validation Program, Verifiers and Error Codes

5. PRT Validation Program, Verifiers and Error Codes

PRT provides the following validation.

Configuration validation program.
PRT Global verifier: prt_glb_vfy.exe

Error codes returned by Production Tracking.
--

Configuration Validation Program

The Configuration Validation Program can be run to produce a report if errors are found in the configuration. In order to run the program:

Open a DOS window and change directory to the data directory for your project and then execute the program by typing the following command at the prompt:

PRTV < filename >

Where

< filename > is the error report file name.

You will receive one of the following two messages:

- No errors found
- Errors found - report in file < filename > .

Sample reports are found below (an asterisk in the error line means that the error is fatal):

```

Jan 22 1991 14:41:51
PRT_SERVICE Error Report
Production Tracking Service Identification File
-----
PRT_SVC_ID Value | Error found
-----|-----
PRT3_DC          | xlat - no known translation srv>ext_hold_svc_id node>
obj>
PRT3_DC          | xlat - no known translation srv>ext_item_svc_id node>
obj>
PRT3_DC          | xlat - no known translation srv>PRT3_DC node> obj>
Jan 22 1991 14:41:51
PRT_GROUP Error Report
Production Tracking Group File
-----
GROUP_ID Value | Error found
-----|-----
GROUP X        | Specified Resource does not exist XX_FR
Jan 22 1991 14:41:51
PRT_REGION Error Report
Production Tracking Region File
-----
REGION_ID Value | Error found
-----|-----
TEST_REGION     | POINT not configured: PTP06_Y
TEST_REGION     | POINT not configured: PTP06_X
TEST_REGION     | Specified Resource does not exist FRX

```

```
TEST_REGION | *SERVICE not configured: PRT4_DC
```

PRT Global Verifier: prt_glb_vfy.exe

PRT Global Verifier: prt_glb_vfy.exe

`prt_glb_vfy.exe` is a utility that validates the integrity of the global sections for a given Production Tracking service.

`prt_glb_vfy.exe` is located in `%BSM_ROOT%\exe`.

The utility requires three inputs from the user:

1. The service name of the PRT data collector that manages the global sections of interest.
2. A value that indicates the type of verification.

1	Analyzes the global sections
2	Attempts repairs.

3. The name of a file in which to deposit the output.

The program:

- Is run from the command line as a utility.
- Requires that the corresponding PRT data collector and data server be terminated to prevent erroneous messages due to the global section changing during the verification.
- Maps onto the global sections, which means that it must be run on the node where the global section backing files exist.

PRT Global Verifier: Validations and Repairs

Below is a list of the validations that are performed by the global verifier and the repairs that can be made:

Problem	Repair Made
Cross-reference element count larger than slots.	Make number of elements be the number of slots.
Cross-reference element count larger than number of slots in use.	Make number of elements to be the number of slots in use.
More locations are configured than allocated.	None.
More locations are occupied than allocated.	None.
Fewer items in the region than locations are occupied.	None.
More items in the region than possible for the occupied locations.	None.

More items in the region than item slots are allocated.	None.
The number of items in the region is not consistent with the start and/or end of the location list.	Set the number of items to zero and the location list to empty.
Count of an item type is negative.	Make the count zero.
Count of all item types does not match the total of items in the region.	None.
A region location index is out of range.	None.
Location list is circular.	None.
An entry of the location list is marked as not being used.	None.
An item index is out of range.	None.
The count of serialized items does not match the number of items in the serialized item list.	Make the count be the number of items in the serialized item list.
The count of non-serialized items does not match the number of items in the non-serialized item list.	Make the count be the number of items in the non-serialized item list.
Location list end is at a different spot than is pointed to.	Make the pointer point to the end of the location list.
An item is linked into the region twice.	None.
An entry on the item list is marked as not being used.	None.
An item is on the wrong type of list.	None.
An item is in the cross reference but not in the region.	None.

Error codes Returned by Production Tracking

The following error codes are returned by the Production Tracking Application Interface:

Number	Defined Constant	Description
11401	PRTI_FILE_READ_ERR	Error reading configuration file:
11402	PRTI_FILE_OPEN_ERR	Error opening one of the configuration files
11403	PRTI_SVC_DEF_MISSING	Undefined Service in file: <filename>
11404	PRTI_REG_DEF_MISSING	Undefined Region in file: <filename>
11405	PRTI_GRP_DEF_MISSING	Undefined Group in file: <filename>
11406	PRTI_XLATE_ERR	Error in ipc_xlate getting physical addr of: <address>
11407	PRTI_AUX_NAM_ERR	Error in ipc_aux_nam
11408	PRTI_ADD_PORT_ERR	Error in ipc_add_port
11409	PRTI_READ_PORT_ERR	Error in ipc_read_port
11410	PRTI_WRITE_PORT_ERR	Error in ipc_write_port
11411	PRTI_MF_INIT_ERR	Error initializing segment

11412	PRTI_CRESEG_ERR	Error creating segment
11413	PRTI_BAD_MSG	Error in segment
11414	PRTI_NOT_STATUS_SEG	Expected STATUS_SEG as response, rcvd:
11415	PRTI_EXTRA_SEGS	Ignored extra segments after STATUS_SEG or END_SEG
11416	PRTI_UNEXPECTED_SEG	Unexpected segment type: svc: <filename>
11417	PRTI_UNEXPECTED_SEG_TYPE	Unexpected segment type from PRT_DS:
11418	PRTI_ITEM_TYPE_NDEF	Item type identifier not valid:
11419	PRTI_ITEM_TYPE_NULL	Item type identifier NULL
11420	PRTI_NUM_ATTS_INVALID	Invalid number of item attributes specified
11421	PRTI_REG_ID_NDEF	Region identifier not valid:
11422	PRTI_REG_ID_NULL	Region identifier NULL
11423	PRTI_GRP_ID_NDEF	Group identifier not valid:
11424	PRTI_GRP_ID_NULL	Group identifier NULL
11425	PRTI_SVC_ID_NDEF	PRT service identifier not valid:
11426	PRTI_SVC_ID_NULL	PRT service identifier NULL
11427	PRTI_NS_EXT_PROC_INVALID	ext_proc_flag set for non-serialized item
11428	PRTI_SER_NO_ID	Serialized item missing both item id and reference id
11429	PRTI_NON_SER_ID	Non-serialized item has item id or reference id
11430	PRTI_INVALID_TRK_TYPE	Invalid Tracking Type (not SERIALIZED/NON_SERIALIZED)
11431	PRTI_INVALID_REG_LOC	Invalid region location
11432	PRTI_EXT_HOLD_NO_GRP	External hold specified without Group ID
11433	PRTI_EXT_HOLD_NON_SER	External hold specified for non-serialized item
11434	PRTI_INT_HOLD_NO_REASON	Internal hold specified without reason
11435	PRTI_REG_NOT_IN_GRP	Region not in group specified
11436	PRTI_REG_NOT_IN_SVC	Region not managed by service specified
11437	PRTI_GRP_NOT_IN_SVC	Group ID inconsistent with service specified
11438	PRTI_NO_DEL_SUCCESSES	No deletes successful to any PRT services
11439	PRTI_INVALID_DEST	Destination region is invalid for item of this type
11440	PRTI_NO_SUCH_ROUTE	No route exists between src: <region> dest: <region>
11441	PRTI_INVALID_ROUTE	Invalid route (%.15s->%.15s) for item type <item type>
11442	PRTI_SER_PARENT	Parent ID specified for serialized item
11443	PRTI_EXT_HOLD_NO_REASON	Group hold specified without reason

11444	PRTI_ROUTE_NDEF	Non-exist. route in PRT_INV_TYPE, src: <region> dest: <region>
11445	PRTI_INV_EXPLICIT_ROUTE	Can't deny explicit item-route, src: <region> dest: <region>
11446	PRTI_NO_TYPE_FOR_ROUTE	Denied type not on route, src: <region> dest: <region> type: <item type>
11447	PRTI_NOTHING_PENDING	No requests pending
11448	PRTI_CFG_NO_ROUTE	Route has no source and no destination region
11449	PRTI_EF_CHECK_ERR	Error checking event flag
11450	PRTI_ITEM_NOT_FOUND	Item not found
11451	PRTI_ITEM_NOT_FOUND_INC_SRCH	Item not found: some service(s) down
11452	PRTI_TOO_MANY_REQ	Total number in list less than number requested
11453	PRTI_UNKNOWN_INIT_SRC	MF_INIT segment received from unknown service
11454	PRTI_NO_SRC_OR_DEST	Route has no source and no destination region
11455	PRTI_NO_SVC IMPLIED	No service specified (via svc_ id/group_id/region_id)
11456	PRTI_BAD_SERVER_RESPONSE	PRT Data Server returned bad/error response
11457	PRTI_NO_MORE_ITEMS	No more items available
11458	PRTI_BAD_POINTER	Pointer to data structure is NULL
11459	PRTI_UNKNOWN_RESPONDER	Response received from unknown service
11460	PRTI_EMPTY_FILE	Configuration file has no records:
11461	PRTI_BAD_SVC_NAME	Bad service name (no underscore):
11462	PRTI_PARTNER_DEAD	Partner dead
11463	PRTI_DG_BUF_FULL	Datagram buffer full
11464	PRTI_NO_SEGS_RCVD	No segments received into datagram buffer
11465	PRTI_API_CANT_RCV	PRT Application not initialized to receive asynch messages
11466	PRTI_INIT_MSG_RCVD	Init segment received from Data Collector
11467	PRTI_PROC_NOT_CONFIGURED	Process is not configured as network process
11468	PRTI_NOT_ALL_SVCS_RESPONDED	Not all services responded
11469	PRTI_NON_SER_ATTS	Non-serialized item has attribute(s)
11470	PRTI_CANT_MOD_ITEM_REF_IDS	Can't modify both item & ref.ids
11471	PRTI_CANT_MOD_ITEM_WOUT_REF	Must specify ref. id to modify item id
11472	PRTI_REGION_NOT_IN_GROUP	Region <region> does not belong to any group
11473	PRTI_GROUP_HAS_NO_REGIONS	Group <group> does not 'own' any regions
11474	PRTI_REORD_NUM_TOO_BIG	element <element> bigger than
11475	PRTI_GRP_SPANS_SVCS	Can't put region %s in group %s (wrong service)

11476	PRTI_SVC_NOMATCH	No configured Services for the PRT System:
11477	PRTI_REGSVCS_NOMATCH	No configured Regions for the PRT System:
11478	PRTI_BAD_ALLOC	Error allocating memory!
11479	PRTI_NULL_ITEMTYPE_LIST	Bad item type list.

Routing Control Objects: RCO

About the Routing Control Objects Module

The Routing Control Objects (RCO) module works in tandem with PRT, using its data to perform enhanced production routing decisions at runtime.

Wherever a routing decision has to be made on the factory floor, an RCO will monitor the site and make decisions based on current production conditions. Each routing site monitored by an RCO is called a control site.

The following topics provide a detailed description of RCO and how it is configured.

Routing Control Object definitions.
Routing Control Object configuration.
Routing Control Object User Interface (RCO_UI).
Relocate RCO and TADB databases.
RCO function blocks.

Routing Control Object Definitions

Routing Control Object Definitions

It is important to have a basic understanding of the parts that make up a [routing control object sequence \(page 325\)](#) and how they relate to each other.

The following sections provide those definitions.

1 (page 325)	RCO control site.
2 (page 327)	Triggers.

3 <i>(page 327)</i>	Decisions.
4 <i>(page 328)</i>	Function blocks.
5 <i>(page 329)</i>	Routing logic modules.

RCO Execution Sequence

The logic used by RCO to execute decisions is illustrated in the flowchart below.

rect 71, 18, 623, 477 [Step 5. Configure Decisions \(page 343\)](#)

RCO Overview Graphic

RCO is an independent process that monitors control sites on the factory floor. Like a traffic cop directing traffic, RCO directs the flow of product through the manufacturing environment using 'decision-based logic'.

[\(page 325\)](#)

1	A site is triggered by an operator		
2	Routing Logic Modules eliminate decisions based on current production conditions		
	Decision 1	Send dryer to Region 2	Heating unit installation.
	Decision 2	Send dryer to Region 3	Temperature gauge installation.
	Decision 3	Send dryer to Region 4	Motor installation.
3	Output Module sets point values sending the dryer to Region 4		
4	The Control cycle ends and resets the control site to wait for the next trigger		

1. RCO Control Site

1. RCO Control Site: Defined

A control site is a location in the factory where a production routing decision must be made.

Example

1	Region 1
2	Control site for split route. A routing decision must be made at this juncture.

3	Split route into regions 2, 3 and 4.
---	--------------------------------------

Events at a control site have a domino-like effect as follows:

1	The site is triggered.
2	The Logic Module eliminates invalid decisions.
3	A Decision is made.
4	The Output Module = product routing.

Control site: Synchronous validation

Synchronous validation occurs during the control cycle to verify that a carrier has arrived at the stop position and allows the control site to stay in synchronization with the factory floor.

In the Tracker Config_UI, the following fields are configured to enable synchronous validation:

Dialog	Group Name	Field Name	Value
Control Site	Execution Sequence	Verification Point	Point ID
		Verification Value	Boolean, integer or string

1	Control site for a split route. A routing decision must be made at this juncture.
2	A control cycle is in progress moving Monitor A to Region 4.
3	Monitor B has moved into the Split control site.
4	RCO waits for synchronous validation before allowing the control site to trigger.

Control site: Asynchronous validation

Each asynchronous validation decision has two additional fields: **Validation point** and **Default validation value**.

1. The Validation Point must be set to the Validation value in order for the decision to be considered complete.
2. RCO will continue to execute control cycles while waiting for validation to occur.
3. When the logic module has selected a decision, the
 - a. Output modules will run and
 - b. Decision will appear in the RCO_UI as Pending.
4. If the decision has a Validation Point configured:
 - a. The site will wait for the validation, and process any additional triggers.
 - b. The decision waiting for validation will be marked as InProcess until the Verification.

5. RCO maintains a history of the last 32 decisions.
 - a. When a new decision is made, it is added to the decision history.
 - b. When the maximum number of configured decisions is exceeded, the oldest decision is removed from the list.
6. RCO triggers an alarm if a decision is removed and was not validated.
7. Once a decision has been removed from the list, it is considered canceled.
8. When Async Validation is configured for a decision and the default value is specified in the Decisions dialog box or in the Output Module via the Set Asynch Validation Value function block, the decision status reads, InProcess.
9. When RCO receives a Point update whose value is equal to the configured Validation Point, the decision is then marked 'Completed'.

RCODB_RP Process

The `RCODB_RP` Process is the database server for RCO clients.

`RCODB_RP` supplies the details of the RCO sites configured in a project to any requesting RCO Runtime UI process. The RCO configuration data is stored in a SQL server database to which this process will have an open connection in order to retrieve information and deliver to the requesting process.

2. Triggers: Defined

A trigger initiates the cycle of events at a control site. A trigger is commonly one or more configured points. When a group of points are configured to trigger a control site, they must be triggered in the correct sequence to start the cycle of events.

Example

1	Points A, B and C must trigger in sequence to send refrigerator to next region.
2	Points A, B and C. trigger in sequence starting the cycle of events at the control site.

3. Decisions: Defined

A decision is made at a control site in the factory based on current production conditions. RCO uses decision-based logic to execute production routing. This means that unlike traditional software where the outcome is determined after conditions have been evaluated decisions are determined in advance. Therefore, all possible outcomes are known in advance and eliminated based on the configured routing logic for the current production conditions.

Example

After the Logic Module ends, RCO determines which decision to execute as follows.

Number of Possible Remaining Decisions	RCO Executes
One	The remaining decision
None	No decision
Multiple	Multiple decisions

The control cycle is ended when RCO detects that the decision is either successfully or unsuccessfully completed.

1	All possible decisions are known in advance.	
2	Decision 1	Car = Blue; Send to Region 1
	Decision 2	Car = Green; Hold
	Decision 3	Car = Red; Send to Mach1 region.

3	Function blocks contain decisions.
4	Configured function blocks eliminate decisions based on current production conditions
5	The red car is sent down the line to the next region.

4. Function Blocks: Defined

Function blocks are standard logic routines from which you create logic modules for routing product, generating errors, disabling the control site, and so on. You set the parameters on each function block so that it meets your specific needs in the production environment. Further, two or more function blocks can be grouped together to meet your production criteria.

There are an abundant number of function block templates to choose from in various categories as follows:

Function block	Description
Conditional (page 382)	Evaluates 'If-Then-Else' conditions.
Core (page 398)	Performs all-purpose function.
Diagnostic (page 434)	Assists in resolving RCO issues.
Include (page 435)	Enables the inclusion of a file into the generated script.

Output (page 436)	Produces an output in the system, for example generating an alarm.
PRT (page 450)	Performs PRT functions.
Routing (page 462)	Performs routing functions.

5. Routing Logic Modules: Defined

A Routing Logic Module (RLM) is one or more function blocks configured to achieve a particular routing result. When using the function blocks provided with the Tracker software option, you set the parameters using dialogs in the Tracker Configuration User Interface. There is no need to write code to achieve routing results; the function blocks automate the entire process for you making this a highly effective feature of the RCO subsystem.

Example

1	A site is triggered and a Routing Logic Module is run with a single function block.		
2	Function block	Decision	Region 1 to Region 2
		Point ID	Pnt_1_Val
		Value	Not Green
		Comparison Operator	Equal to
		Comparison Type	Alphanumeric

3	The RLM is matched against the point value	
4	Pnt_1_Val	Yellow
5	The printer is moved to Region 2.	

Routing Control Object Configuration

Routing Control Object Configuration

Step 1 (page 330)	Enable the Tracker Configuration User Interface for RCO.
Step 2 (page 332)	Lock a folder or existing site.
Step 3 (page 332)	Configure a routing control site.

Step 4 (page 342)	Configure triggers.
Step 5 (page 343)	Configure decisions.
Step 6 (page 346)	Configure routing logic.
Step 7 (page 347)	Save RCO configuration data.
Step 8 (page 347)	Activate the Routing Control site.
Step 9 (page 348)	Set RCO alarming and logging in the TrackerCfg_UI.
Step 10 (page 349)	Use additional configuration functions and utilities.

Step 1. Enable the Tracker Configuration User Interface for RCO

Step 1. Enable the Tracker Configuration User Interface for RCO

! **Important:** You must have licensed and enabled the Tracker option to your project to enable access to the Tracker Configuration User Interface.

Step 1.1 (page 330)	Open the Tracker Configuration User Interface.
Step 1.2 (page 331)	View the Tracker Configuration User Interface: RCO section.
Step 1.3 (page 331)	View the Tracker Configuration Tree: RCO section.

Step 1.1. Open the Tracker Configuration User Interface

1. Select **Project>Tracker Configuration** in the Workbench left pane.
2. Select **Tracker Configuration** in the Workbench right pane.
3. Do one of the following.

A	Click Edit>Properties on the Workbench menu bar.	
B	Click the Properties button on the Workbench toolbar.	
C	In the Workbench left pane:	
	Either	Or
	Double-click Tracker Configuration .	a. Right-click Tracker Configuration . b. Select Properties on the Popup menu.
D	In the Workbench right pane:	
	Either	Or
	Double-click Tracker Configuration .	a. Right-click Tracker Configuration . b. Select Properties on the Popup menu.
E	Press Alt+Enter on the keyboard.	

4. Right-click **Tracker Configuration**.

5. Select Properties on the Popup menu.

6. Right-click **Tracker Configuration**.

7. Select Properties on the Popup menu.

Step 1.2. View the Tracker Configuration User Interface: RCO Section

The Tracker Configuration user Interface (TrackerCfg_UI) contains the tools required for configuring the automated routing control object functionality.

Step 1.3. View the Tracker Configuration Tree: RCO Section

The RCO directory is organized in a hierarchy so that information for a control site can be easily accessed, viewed, and modified.

Each level of data in the RCO hierarchy has a function as follows:

1	RCO Site	Contains all of the instructions for making production routing decisions in the RCO configuration.
2	Control sites folder	Contains all of the instructions for making production routing decisions for a group of control sites.
3	Control site	Contains all of the instructions for making production routing decisions for a control sites.
4	Trigger folder	Folder that holds all of the configured triggers for the control site.
5	Trigger	One or more points that will activate the control site when its defined conditions are met.
6	Decisions folder	Folder that holds all of the configured decisions for the control site.

7	Decision	Selected for configuration. Instructions for executing an output module, which in turn, sets a series of points. Actual production routing is performed at this juncture.
8	Decision	Not selected
9	Routing_Logic folder	Folder that holds all of the configured routing logic modules for the control site.
10	Routing logic modules	Modules that consist of function blocks to achieve a particular routing result in line with factory conditions.

Step 2. Lock a Folder or Existing Site

1. Select the site or folder to be locked.

Note: The first time you open the TrackerCfg_UI, the project the folder will be the main level folder.

2. Do one of the following.

Method 1

- Click File>Lock on the TrackerCfg_UI menu bar.

Method 2

- a. Right-click to display the Popup menu.
- b. Do one of the following:

For the top level folder

If you are the only user for the entire project and you selected the top-level folder

Select Lock All.

You can now work with the dialog boxes and fields in the entire project.

For any folder

Select Lock.

You can now work with the dialog boxes and fields for items in the selected folder's tree.

Step 3. Configure a Routing Control Site

Step 3. Configure a Routing Control Site

A site level in the Tracker Configuration User Interface consists of points and logic modules that are common for all items and events.

- Configuration hierarchy overview
- Control site configuration steps

Configuration hierarchy overview

! **Important:** Configuration of lower-level data items, for example, triggers or decisions, will override site level configuration.

1	Control site configuration is common for all items and events.
2	Configuration of lower-level data items overrides control site configuration.

Control site configuration steps

To configure a control site, perform the following tasks:

Step 3.1 (page 333)	Create a new site.
Step 3.2 (page 334)	Name and describe RCO.
Step 3.3 (page 335)	Configure the General Setup group.
Step 3.4 (page 336)	Configure the Output Modules group.
Step 3.5 (page 341)	Configure the Execution Sequence group.
Step 3.6 (page 342)	Configure the automatic triggers group.

Step 3.1. Create a New Site (RCO)

1. Lock the folder in which you are going to place the Routing Control Object.
2. (Optional) Create a new folder.
 - a. Do one of the following.
 - Click File>New on the TrackerCfg_UI menu bar.
 - Right-click the folder; select Routing Control Object from the Popup menu.

A new folder displays at the bottom of the selected folder's tree.

a. Name the folder.

3. Do one of the following to create a new Routing Control Object.

Method 1

Click File>New>Routing Control Object.

Method 2


a. Right-click the RCO folder.

b. Click New>Routing Control Object on the Popup menu.

A New_Site icon is created in the TrackerCfg_UI left pane.

The New_Site displays the following.

A	Three subfolders are automatically created with the new site.
	<ul style="list-style-type: none"> • Triggers, • Decisions and • Routing Logic.
B	Configuration continues in right pane.

 **Note:** Organize information in the directory by creating folders. From the File menu, select New>Folder and a folder icon will display in your directory. You can then drag and drop sites into and between folders.

Step 3.2. Name and Describe the Site

Configure the top group of the RCO dialog box.

The options are as follows.

rect 140, 84, 447, 114 ([page 334](#))

rect 139, 167, 446, 197 ([page 335](#))

rect 139, 140, 446, 170 ([page 335](#))

rect 140, 114, 447, 144 ([page 334](#))

	Option	Description
1	Name	Unique name for the RCO site. Important: The name can be a total of 16 characters or less.
2	Description	Brief description of the purpose of the RCO.

3	Resource ID	Click either:
		Browse for an existing resource in the project, or
		Designate a new resource.
4	Enable Unexpected point processing	Check to allow RCO to generate an alarm when a point change that is unexpected comes into a site.
		<p>Unexpected point processing applies to three RCO point types:</p> <ul style="list-style-type: none"> • Trigger, • Validation and • Verification. <p>The Enabled Unexpected point processing option:</p> <ul style="list-style-type: none"> • Allows the RCO system to disable the control site when a point change that is not expected occurs on one of these point types. • Displays an error in the RCO_UI, • Generates an alarm. <p>Note: Configure the alarm state using the Alarming/Logging dialog box for details.</p>

Step 3.3. Configure the General Setup Group

Use the following fields to configure the General Setup.


rect 123, 186, 402, 217 ([page 335](#))

rect 123, 215, 402, 246 ([page 335](#))

rect 124, 247, 403, 278 ([page 336](#))

rect 123, 277, 402, 308 ([page 336](#))

	Field	Description
1	RLM Point	Updated during RCO runtime with the name of the routing logic module (RLM) to be executed for the site.
		<p>Guidelines The RLM point is a:</p> <ul style="list-style-type: none"> • Virtual point. • String data type. <p>It is either updated at Runtime with the name of the routing logic module to be executed, or a user can set the point with the name of the RLM. Important: Make sure that Enable Point is checked on the General tab of the Point Properties dialog box.</p>
2	Status Point	Updated to indicate the status of the control site.
		Status point values are:
	0	Enabled control site.
	1	Disabled control site.
	2	Control site is in suspend mode.

		<p>Guidelines The Status point is a:</p> <ul style="list-style-type: none"> • Virtual point. • Integer data type. <p>It is either updated at Runtime with the value of the control site status, or a user can set the point to manage the status of the control site. Ensure that Enable Point is checked on the General tab of the Point Properties dialog box.</p>
3	Danger Point	Determine if any decisions without a 'local danger point' are to be rolled back.
		Boolean values are:
	0	No danger and decision is to be eliminated.
	1	In danger and decision is to be rolled back.
		<p>Guidelines The Danger point is a:</p> <ul style="list-style-type: none"> • Device or a virtual point. • Boolean data type. <p>It is updated at runtime with the danger status of the control site. If a control site has:</p> <ul style="list-style-type: none"> • Eliminated all decisions and • An Always Evaluate flag of Breakable with Danger Point High, <p>then the RCO system automatically checks the danger point. When the Danger point = 1, the RCO is in a danger state and the system automatically rolls the list back to the previous logic block. Ensure that Enable Point is checked on the General page of the Point Properties dialog box.</p>
4	Initialize Module	The initialize script allows a user to call many different function blocks.
		<p>This script will only run the first time the site is triggered. For example you can Initialize one of the following:</p> <ul style="list-style-type: none"> • Site attributes with any COM objects or values needed by the site during normal execution. • Points <p> Important: Do not select or make a decision in the logic module. Any decision changes will be ignored.</p>

Step 3.4. Configure the Output Modules Group

Step 3.4. Configure the Output Modules Group

- Output module field descriptions.
- Tools for filling in the Output Module fields.
- Output module Popup menu.

Output module field descriptions

Use the following four fields to specify Output Modules.

rect 101, 226, 388, 253 ([page 337](#))

rect 102, 311, 389, 336 ([page 337](#))

rect 103, 281, 390, 307 ([page 337](#))

rect 100, 256, 387, 275 ([page 337](#))

Field	Description
1	Single
	The Single output module that will be executed when a single (one) decision survives the routing logic and after the decision output logic is executed, i.e. control site works properly.
	Guidelines This module is most commonly used to run output logic that is common for all successfully executed decision.
2	Multiple
	The Multiple output module that will be executed when multiple (two or more) decisions survive the routing logic.
	Guidelines This module is used to fix the problem of having multiple decisions left and can be used to generate an alarm for user intervention. Note: The control site cycle ends after the Multiple Output module is run. (Use the RCO Execution Sequence as a guide.)
3	None
	The None output module that will be executed when there are no (zero) decisions that survive the routing logic.
	Guidelines This module can be used to generate an alarm for user intervention, or to set a done point that will in turn prepare the site to re-trigger the control site. Note: The control site cycle ends after the None Output module is run. (Use the RCO Execution Sequence as a guide.)
4	Runtime Error
	The Runtime Error output module that will be executed when routing logic fails during the decision-making cycle.
	Guidelines This module can be used to generate an alarm to initiate user intervention, or to set a done point that will in turn prepare the site to re-trigger the control site.

Tools for filling in the Output Module fields

The two options for filling in these fields are:

Option 3.4.1 (page 338)	Create a new output module.
Option 3.4.2 (page 340)	Use a predefined output module.

Output module Popup menu

Click next to output module fields to display the popup menu.

Option	Description
--------	-------------

New...	Open the Output Module Wizard to create a new output module.
Edit...	Edit the function block in a selected output module.
Browse...	Open the Module Browser to select a predefined output module.
Delete	Delete the selected item.
Scripts	Open the Program Editor.

Option 3.4.1. Create a New Output Module

Option 3.4.1. Create a New Output Module

1. Click adjacent to the field where you want to create a new output module.
2. Select **New** from the Popup menu.

The Output Logic Wizard opens.


3. Identify the block.

Fields are as follows.

Field	Description
Name	Unique identifier for the output module.
Description	Describes the purpose of the module.

4. Do either or both of the following to add function blocks to the module.

Option 3.4.1.1 (page 338)	Add New Function Blocks to a Logic Module
Option 3.4.1.2 (page 340)	Cut/Copy/Paste Existing Function Blocks to and from Logic Modules

 **Tip:** Use the toolbar in the Wizard to add or delete function blocks, and to change the position of the function block up or down in the list. Also, you can use the checkbox next to each function block to enable or disable it in the list. Be sure to compile when you have made changes to the function block list.

Option 3.4.1.1. Add New Function Blocks to a Logic Module

1. Click on an Output or Routing Logic Wizard toolbar.

The Select a Function Block dialog box opens.

2. Expand the Function Block Class folder to display the corresponding function blocks.
Descriptions are as follows:

Class	Description
Conditional	Evaluates If-Then-Else conditions.
Core	Generic, all-purpose blocks.
Diagnostics	Performs diagnostic functions.
Include	Enables the inclusions of basic scripts.
Output	Produces an output in the system, such as generating an alarm.
PRT	Performs PRT functions.
Routing	Performs routing functions.

3. Double-click the function block that you will add to the output module.

The dialog box for the function block opens.

4. Enter a brief **Description** for the module.
5. Set parameters by double clicking each of the Tags in the list box.

Note: The dialog box that opens is dependent upon the tag.

6. Set the value for the parameter using the provided input option, e.g. drop-down list or Popup menu.

Note: The input options in the Parameter dialog box vary by parameter. There can be drop-down lists, Popup menus, browse buttons or edit controls that can be used to set the value of the parameter. In the example above, there is a drop-down list.

7. Click **OK** to return to the function's list of parameters.
8. Continue to add additional required parameter values to the list.
9. Click **OK** to return to the Output Logic Wizard.
10. Continue to add function blocks by either [cutting/copying/pasting \(page 340\)](#) or adding new blocks.
11. Click **Compile** to compile the output or routing module and to display any syntactic errors that may exist.

12. Click **OK** to save the configuration of the output module and (for the output logic module) close the Wizard.

Option 3.4.1.2. Cut/Copy/Paste Existing Function Blocks to and from Logic Modules

1. Open the source logic module that has the function block you want to copy or cut.
2. Right-click the function block you want to cut or copy.
3. elect Cut or Copy on the Popup menu.
4. Close the source logic module.
5. Open the target logic module.
6. Make sure that at least one function block is listed.

Note: You can delete the function block after you paste or add another block.

7. Right-click a function block in the target module.
8. Select Paste on the Popup menu.

The pasted function block is inserted in the list in the module before the block used to open the Popup menu.

Note: You can also cut, copy and/or paste function blocks within a logic module.

9. Continue to add function blocks by either cutting/copying/pasting or [adding new \(page 338\)](#) blocks.
10. Click **Compile** to compile the output or routing module and to display any syntactic errors that may exist.
11. Click **OK** to save the configuration of the output module and (for the output logic module) close the Wizard.

Option 3.4.2. Use Predefined Output Module

1. Click adjacent to the field where you want to add a predefined output module.

The Module Browser window displays.

2. Do one of the following:
 - Double-click an output module in the list or


- Select an output module from the available list and click **OK**.


Step 3.5. Configure the Execution Sequence group

You can configure the Output Modules at the site level to execute a set of Function Blocks or to execute custom BCE scripts that you created.

- rect 83, 263, 379, 297 [\(page 341\)](#)
- rect 88, 443, 384, 472 [\(page 342\)](#)
- rect 86, 417, 382, 440 [\(page 342\)](#)
- rect 86, 385, 382, 419 [\(page 341\)](#)
- rect 85, 354, 381, 388 [\(page 341\)](#)
- rect 85, 324, 381, 358 [\(page 341\)](#)
- rect 83, 294, 379, 326 [\(page 341\)](#)

Use the following fields to configure the execution sequence.

Item	Field	Description
1	Ready Point	Name of an analog CIMPLICITY Point that will be updated to indicate that the automation equipment is ready for a decision to be executed. The RCO runtime system will wait until the Ready Point is updated to the value specified in the Ready Value field before output script for a selected decision is executed
2	Ready Value	Value written to Ready Point to indicate that the automation equipment is ready for a decision to be executed.
3	Done Point	Name of an analog CIMPLICITY Point to be updated when a decision has been executed. When the RCO runtime system has completed the execution of the output script associated with the selected decision, it updates the Done Point to the value specified in the Done Value field.
4	Done Value	Value written to the Done Point by the RCO when the decision output script has been executed. The value entered must be numeric.
		<p>To insure that RCO recognizes an updated Done Point:</p> <ol style="list-style-type: none"> 1. Configure a site with a Done point. When the Done value updates to a new value the site is done. 2. Change the point to a value other than the Done point value either through a script or manually. <p>Result: RCO notifies you the next time it sets the point to the Done value.</p> <p> Important: If an updated Done point value is the same as the previous value, RCO will not know that the value is updated. Therefore, a decision may time out waiting for a Done point even though the point is set to the correct value.</p>
5	Sync Val Point	Name of a point that will be updated to indicate that a decision has been successfully completed. An RCO waits for this point to be set to the value specified in the Verif. Value.

Item	Field	Description
6	Sync Val Value	Value to be written to Verification Point by the automation equipment or other process to indicate that the decision execution has been successfully completed. The site will stop all execution until the value indicates that the decision has been completed.  Note: If Sync Val Value is -1, then the validation will complete when the Sync Val Point is updated (irrespective of its value).
7	Sync Val Timeout	Value of a time out in seconds after which the: <ul style="list-style-type: none"> • Get Sync Val fails and the • Site is disabled. <p>A value of 0 makes the site wait 10 days for the verification point to get set to the verification value. After 10 days, it causes a synchronous validation failure.</p>

Step 3.6. Configure the Automatic Triggers Group


The Automatic Triggers group enables you to configure when your site will be triggered.

Configuration options are as follows.


rect 35, 405, 301, 432 [\(page 342\)](#)

rect 35, 430, 301, 457 [\(page 342\)](#)

	Option	Description
1	Trigger Interval	The number of seconds before automatically triggering the site in the field.
		If the site is not triggered within this time period, it will be triggered automatically.
2	Trigger On Start	(Checked) Trigger the site immediately after starting.

 **Note:** You may use both of the options in the Automatic Triggers group.

Step 4. Configure Triggers

 **Note:** Make sure the site is locked.

1. Right-click the RCO Triggers folder in the TrackerCfb_UI left pane.
2. Do one of the following.

Method 1

Select File>Routing Control Objects on the Tracker menu bar.

Method 2

- a. Right-click the Triggers icon.
- b. Select New Trigger Sequence from the Popup menu.

A new Trigger item displays in the left pane; the right pane displays the dialog for configuring a trigger.

3. Enter the following:

	Field	Description
A	Name	A unique name for the new trigger.
B	Description	(Optional) A brief description describing the purpose of the trigger.

4. Click to display the Trigger Details dialog box.

rect 21, 34, 292, 74 ([page 343](#))


rect 18, 81, 178, 201 ([page 343](#))

5. Configure the trigger details as follows.


Field	Description	
Trigger Point	Point value is used to determine the trigger sequence of events. The point type depends on what value is being compared.	
Trigger Type	The checked radio button specifies when the trigger will be activated, as follows.	
	On Update	Value of the Point must change to activate the trigger.
	Transition High	Value of the Point must start high to activate the trigger. The initial state of this trigger upon CIMPPLICITY startup is high.
	Transition Low	Value of the Point must change from a non-zero value to 0 to activate the trigger. The initial state of this trigger upon CIMPPLICITY start is low, so the point must change to a non-zero value, and then to a 0 value.
	Value	Point changes to the value you specify to activate the trigger. When radio button is clicked, the Value field will activate. Type the value in the field.

6. Click **OK**.

The new trigger is listed in the TrackerCfg_UI left pane; when selected, its details display in the right pane of the window.

 **Note:** More than one trigger can be added to the Routing Control Object. Repeat procedure to add more triggers.

Step 5. Configure Decisions

 **Note:** Make sure the site is [locked \(page 332\)](#) .

1. Select **Decisions** icon or folder in selected site tree.
2. Do one of the following:

Method 1

- a. Select File on the Tracker menu bar.
- b. Select Routing Control Objects.

Method 2


- a. Right-click the **Decisions** icon.
- b. Select New Decision from the Popup menu.

A new Decision item displays in the left pane; the right pane displays the dialog for configuring a decision.

3. Define the decision.

Fields are as follows.

rect 156, 67, 356, 89 [\(page 344\)](#)
 rect 156, 87, 356, 109 [\(page 345\)](#)
 rect 156, 107, 392, 129 [\(page 345\)](#)
 rect 156, 127, 396, 149 [\(page 345\)](#)
 rect 156, 145, 399, 167 [\(page 345\)](#)
 rect 155, 165, 355, 187 [\(page 345\)](#)
 rect 156, 87, 356, 109 [\(page 345\)](#)
 rect 156, 185, 384, 207 [\(page 345\)](#)
 rect 157, 205, 386, 227 [\(page 345\)](#)
 rect 158, 225, 388, 247 [\(page 345\)](#)
 rect 156, 246, 388, 268 [\(page 345\)](#)
 rect 157, 265, 385, 285 [\(page 345\)](#)
 rect 155, 283, 387, 305 [\(page 345\)](#)
 rect 155, 302, 385, 321 [\(page 345\)](#)
 rect 155, 318, 386, 340 [\(page 345\)](#)
 rect 156, 339, 356, 361 [\(page 345\)](#)
 rect 154, 361, 276, 383 [\(page 345\)](#)

 **Tip:** Use the Browser and Popup buttons to the right of fields to browse for/edit existing items or to create new items.


Field	Description
Decision ID	Numeric ID that must be unique.

Decision Name	Unique name.
Description	(Optional) Brief textual description.
Source	Region that the item will move from.
Destination	Region the item will move to.
Balance Weight	Numeric weight associated with the decision. At runtime, this weight will be used when evaluating / selecting an appropriate decision for execution.
Danger Point	The RLM will look for the site danger point when this field is blank. If both points are undefined, the decision will be eliminated. For decisions with an Always Evaluate flag of Breakable, this point determines if any decisions without local danger point is rolled back. A value of "1" means "In Danger" and the decision should be rolled back.
Ready Point	CIMPLICITY analog Point that will be updated indicating that the automation equipment is ready for a decision to be executed. The RCO runtime system will wait until the Ready Point is updated to the value specified in the Ready Value field before the output module for a selected decision is executed.
Ready Point Value	Value written to Ready Point indicating that the automation equipment is ready for a decision to be executed.
Output Logic Module	Specify the name of an output logic module to be executed when this decision is selected. Do either of the following. <ul style="list-style-type: none"> • Select an existing logic module from the Module Browser. • Create a new module be adding new (page 338) or pasting (page 340) existing function blocks.
Async Val. Point	Point that will be updated indicating that the decision has been completed.
Async Val. Point Value	Value to be written to the Validation point by automation equipment or other process to mark the decision status as either: <ul style="list-style-type: none"> • In process or • Completed. The site can continue execution when the value is In process.
Sync Val Point	Point that will be updated indicating when the decision has been completed.
Sync Val Point Value	Value to be written to the Sync Val point by automation equipment or another process to mark the already executed decision status as completed . The site will stop all execution until the value indicates that the decision has been completed.
Sync Val Timeout	Value of a time out in seconds after which the: A. <code>Get_Sync_Val</code> fails and the B. Site is disabled.
	A value of 0 makes the site wait indefinitely for the verification point to get set to the verification value.
Disabled Decision	Disables decision.

4. Click OK.

The new decision is listed in the left pane; when selected, its details display in the right pane of the window.

Step 6. Configure Routing Logic

 **Note:** Make sure the site is locked.

1. Select in the selected site tree.
2. Do one of the following.

Method 1

Select File>Routing Control Objects on the Tracker menu bar.

Method 2

- a. Right-click the **Routing Logic** icon.
- b. Select New Logic Module from the Popup menu.

A new routing logic item displays in the left pane; the right pane displays the dialog for configuring a logic item.

3. Use the fields and check box to define the routing logic.

Options are as follows.


rect 158, 60, 415, 90 ([page 346](#))
 rect 158, 91, 415, 121 ([page 346](#))
 rect 158, 117, 261, 147 ([page 346](#))
 rect 164, 147, 399, 211 ([page 346](#))
 rect 328, 317, 400, 347 ([page 346](#))

Option	Description
Name	Enter a unique name for this routing logic module.
Description	(Optional) Enter a unique description for this routing logic module.
Default Script	Each site can have one default script. The default script is run if there is no RLM point configured. The default script name is also written to the RLM point on startup if the point is blank.
Function blocks	Go to 5.
Compile	Compiles the default script or Routing Logic.

4. Add function blocks the same way that you do for output logic modules.

Option 3.4.1.1 (page 338)	Add New Function Blocks to a Logic Module
Option 3.4.1.2 (page 340)	Cut/Copy/Paste Existing Function Blocks to and from Logic Modules

The logic module is a configured part of the routing logic for the site.

 **Note:** More than one module can be added to the Routing Control Object. Repeat procedure to add more routing logic modules. Depending on the function block you selected, different types of dialog boxes display requiring unique configuration and parameter values. Similarly, the Parameter dialog boxes vary by parameter. There can be drop-down lists, popup menus, browse buttons or edit controls that can be used to set the value of the parameter.

Step 7. Save RCO Configuration Data

Once you have configured the routing control Site, you must save the data and activate each configured folder. Routing control sites within active folders are supported in the RCO runtime environment, while inactive folders will be ignored.

Method 1

1. Click File on the menu bar.
2. Select **Save**.

Method 2

Click .

Using either method, the directory structure and configuration information will be saved to the database.

Step 8. Activate Routing Control Sites

Once the data has been configured for the routing control sites, you must activate them.

Use either method to activate the site.

Method 1

1. Click a folder in the directory.
2. Click File on the menu bar.
3. Select **Activate**.

Method 2

4. Right-click the folder in the directory.
5. Select **Activate** from the popup menu.

The folder is activated and will be included in the runtime environment.

Step 9. Set RCO Alarming and Logging in the TrackerCfg_UI

Step 9. Set RCO Alarming and Data Logging in the TrackerCfg_UI

You can configure alarming and logging options from both the TrackerCfg_UI and the RCO_UI.

The difference is as follows.

From the:	Alarming and Logging changes are:
TrackerCfg_UI	Stored in the database, but are not effected dynamically.
RCO_UI	Effected dynamically, but are not stored in the database.

 **Important:** Make sure the RCO site is locked in order to enable the Alarming / Logging option on the popup menu.

1. Right-click a **Site** icon in the Tracker window left pane.
2. Select Alarming/Logging from the Popup menu.

The Alarming/Logging Options dialog box opens.

3. Check the options that should be enabled.


1	Enable Alarming enables alarms for selected conditions.	
2	Checked states have enabled alarms	
3	Checked selects the log destination as follows.	
	Disable	Disables logging
	Data Logger	Logs event to the database logger.
	Trace File	Logs detailed data to a trace file with a .log extension in the project directory.
4	Checked conditions cause data to be logged.	

4. Click **OK** to save alarming and logging options, or click **Cancel**.
5. Enable data logging from the data logging side.

- a. Configure the **datalog.idt** file for RCO logging.
- b. Configure the **data_field.idt** file for RCO logging.

RCO Alarms displayed in the Workbench

1. Open the CIMPPLICITY project that includes your RCO configuration.
2. Expand the Advanced folder in the Workbench left pane.
3. Select the **Alarms** icon to view configured alarms in the right pane of the Workbench.
4. Type RCO in the right pane or scroll to view the pre-configured RCO alarms.

 **Tip:** Modify alarm properties by double-clicking the alarm and making changes in its Properties dialog box.

Step 10. Use Additional Configuration Functions and Utilities

Step 10. Use Additional Configuration Functions and Utilities

There are additional functions that you can perform to customize the RCO configuration.

For example you can add custom scripts and subroutines to the global library for inclusion in RCO scripts, assign a custom name to predefined Production Tracking (PRT) attributes, or assign additional points to be used in the RCO runtime environment. Also provided are utilities that enable you to verify the RCO configuration data and view a list of errors.

You can take advantage of all of the following features:

Option 10.1 (page 350)	Customize RCO folder properties.
Option 10.2 (page 350)	Assign custom names to PRT attributes.
Option 10.3 (page 351)	Make further runtime region specifications.
Option 10.4 (page 352)	Check the RCO configuration data.
Option 10.5 (page 352)	Run the error list utility.

Option 10.1. Customize RCO Folder Properties

You can customize the properties of the folders in your directory. This includes the main folder that is generated by default in the directory, and subfolders that you create to organize your RCOs.

Two important properties can be set:

Thread Count	Specifies the maximum number of sites that can execute simultaneously. Valid range is 1 to 10.
Global Library	Compatible file that includes BCE scripts and/or subroutines that can be called at runtime from any RCO script. This file may not include a 'sub main'.

1. Select the folder in the directory.
2. Right-click the folder and select **Lock** from the popup menu.

A check mark next to Lock indicates the folder is locked.

3. Right-click the folder and select **Properties** from the popup menu.

The properties dialog box opens. Notice that the name of the folder appears in the title bar.

Configure the folder properties as follows.

Field	Enter
Thread Count	The maximum number of sites, included in this folder, to execute simultaneously. The valid range is 1 to 10 sites at a time.
Global Library	File containing the BCE scripts and/or subroutines that you want to add to the RCO Module. Important: Use .bcl scripts only; do not use .bclrt scripts. Note: Click next to the field to display the Open dialog box and find the files.

4. Click **OK** to activate the changes, or **Cancel** to cancel the operation.

The file name displays in the Global Library field and is now available to be called at runtime from any RCO script.

Option 10.2. Assign Custom Names to PRT Attributes

1. Click Tools on the RCO menu bar.
2. Select **Attribute Maintenance** to display the Attribute Maintenance dialog box.
3. Click to open a second Attribute Maintenance dialog box.
4. Define an attribute as follows:

Description	A custom name that identifies the attribute for users.
Attribute	Attribute associated with the descriptive name.

5. Click **OK** to return to the Attribute Maintenance dialog box.
6. Click **OK** to close the dialog box.

Option 10.3. Make further Runtime Region Specifications

1. Click Tools on the Tracker window menu bar.
2. Select **Region Maintenance** to open the Region Maintenance list box.
3. Do one of the following.
 - Edit an existing maintenance for a region

Double-click a region.

The Region Maintenance dialog box for the selected region opens for [editing \(page 351\)](#).

- Create new maintenance for a region

Click .

A blank Region Maintenance dialog box opens for [configuration \(page 351\)](#).

4. Fill in or edit the involved fields, as follows.

Field	Description
Region ID	ID of the region.
Ready Mask	Value is binary and with Region Status Point value to determine if the region is ready.
Region Script	Reserved for future expansion.
Global Var. Point	Reserved for future expansion.
Enabled Point	If 0, the region is disabled; otherwise, it is enabled.
Capability Point	
1 – 2 – 3	Specifies the capability of the region using attributes. These points contain a set of attribute values used to determine the capability of a region and used by the Check Process Capability function block. Type of point must be a Text point of type STRING. If the point is an array, each element of array will be scanned until a match is found for the attribute.

Quantity Point	Enter the total number of occupied locations in the region.
Quantity Value	Instead of configuring the quantity in a point, it may be configured as a single value. Routing control will first look for the point, if the point is undefined it will look for the quantity value.
Status Point	Enter the region status point from PRT. Point must be BOOL, UINT, UDINT, INT, DINT, or REAL.
Capacity Point	Enter the maximum locations a region can hold. This may be different from PRT. Point must be BOOL, UINT, UDINT, INT, DINT, or REAL.

5. Click OK to add the new configuration to the runtime environment.

Option 10.4. Check the RCO Configuration Data

1. Click Tools on the menu bar.
2. Select **Verify**.

Option 10.5. Run the Error List Utility

Click Tools>Error List on the menu bar.

The Error dialog opens

1	Errors in configuration.
2	When clicked, displays area on screen where a selected error occurred.

2. Click **Close** to exit the dialog box.

Routing Control Object User Interface (RCO_UI)

Routing Control Object User Interface (RCO_UI)

The RCO Runtime User Interface enables you to view your control sites during runtime. Control Sites are the critical points in your production process where control decisions are made by RCO. Aside from the comprehensive data you can view about each site in the window, there are a number of other functions you can perform and other methods for viewing additional information.

Using the RCO Runtime User Interface, you can:

Step 1 (page 353)	Connect to the RCO Runtime User Interface.
Step 2 (page 358)	Select the control site.

Step 3 (page 358)	Review the RCOUI features.
Step 4 (page 362)	Create or open profiles.
Step 5 (page 365)	Configure an RCO profile.
Step 6 (page 367)	Manage decisions.
Step 7 (page 369)	Manage triggers.
Step 8 (page 372)	Set alarming and logging options through the RCOUI.
Step 9 (page 372)	Search for control sites.
Step 10 (page 373)	Respond to RCO error conditions.

Step 1. Connect the RCO Runtime User Interface

Step 1. Connect the RCO Runtime User Interface

Before you start the RCO Runtime User Interface, there must be an RCO project running with active control sites. The interface is used specifically for viewing RCO runtime data and for performing certain control functions.

In order to access the Configuration dialog to create new profiles, the **/config** switch is needed in the Command line. The command looks like the following: **C:\rco_ui/config < profile file >**, where **C** is the drive where the RCO UI is stored.

 **CAUTION:** When upgrading your CIMPLICITY software, you must upgrade your RCO projects or you may not be able to access the RCO Runtime User Interface.

You can connect to the RCO_UI using the following options:

Option 1.1 (page 354)	Connect to local project.
Option 1.2 (page 355)	Connect to projects in an existing RCO profile.

Option 1.3 (page 356)	Connect to project from the command line.
Option 1.4 (page 357)	Connect to the configuration dialog.

Option 1.1. Connect to a Local Project

1. Make sure the Tracker project is running.
2. Use one of the following.
 - Workbench
 - Start menu

Workbench

- a. Select **Runtime>Production Tracking>Routing Control Objects UI** in the Workbench left pane.
- b. Select **Routing Control Objects UI** in the Workbench right pane.
- c. Do one of the following.

1	Click Edit>Properties on the Workbench menu bar.	
2	Click the Properties button on the Workbench toolbar.	
3	In the Workbench left pane:	
	Either	Or
	Double-click Routing Control Objects UI .	a. Right-click Routing Control Objects UI . b. Select Properties on the Popup menu.
4	In the Workbench right pane:	
	Either	Or
	Double-click Routing Control Objects UI .	a. Right-click Routing Control Objects UI . b. Select Properties on the Popup menu.
5	Press Alt+Enter on the keyboard.	

Start menu

- a. Click Start on the Windows task bar.
- b. Select (All) Programs>Proficy HMI SCADA - CIMPLICITY version>RCO Runtime User Interface.

Results

- If you have not logged into the CIMPLICITY project,


A CIMPLICITY Login dialog box opens.


Enter your **User ID** and **Password**.

- When you are logged in:

A Search dialog box displays showing a list of active control sites in your local project, when you use either method.

3. Right-click **Routing Control Objects UI**.
4. Select Properties on the Popup menu.
5. Right-click **Routing Control Objects UI**.
6. Select Properties on the Popup menu.
7. Select the site you want to view.
8. Double-click the site to open the RCO User Interface.

 **Note:** If there is only one active control site in your local project, the Search dialog box will be bypassed and the RCO User Interface will open directly to the active site.

 **Tip:** Place a shortcut of the RCO Runtime User Interface icon on your desktop for quick access.

Option 1.2. Connect to Projects in an Existing RCO Profile

1. Do one of the following:

Method 1


- a. Open Windows Explorer.
- b. Select an RCO profile file (.rui).
- c. Double-click the file.

Method 2

- a. Place a shortcut of the profile file (.rui) on your Windows desktop.
- b. Double-click the shortcut.

The Search dialog box displays showing the list of active control sites in the profile, when you use either method.

2. Select the site you want to view.
3. Click **OK** to open the RCO User Interface.

 **Note:** there is only one active control site in the profile, or a default site is configured, the Search dialog box will be bypassed and the RCO User Interface will open directly to the active site.

Option 1.3. Connect to Projects from the Command Line

1. Do one of the following:

Method 1

- a. Place a shortcut of the **RCO Runtime User Interface** (rco_ui.exe) file on your Windows desktop.
- a. Right-click the icon and select **Properties** from the popup menu to display the RCO Runtime UI Properties dialog box.
- b. Enter in the Target field the

Path to rco_UI.exe followed by the path to the profile file, for example

```
"C:\Program Files\Proficy\Proficy CIMPLICITY\exe\rco_ui.exe" E:\RCO
\Profile.rui
```

1	Path to rco_ui.exe.
2	Path to profile (.rui) file.

- a. Click **OK**.
- b. Double-click the icon.

Method 2

- a. Click **Start** on the Windows task bar.
- b. Select **Run** to display the Run dialog box.
- c. Enter **rco_ui** followed by the fully qualified path to the profile file,

Example

```
rco_ui d:\RCO\Profile1
```

- a. Click **OK**.

The Search dialog box displays showing the list of active control sites in the profile, when you use either method.

2. Select the site you want to view.
3. Click **OK** to open the RCO User Interface.

 **Note:** If there is only one active control site in the profile, or a default site is configured, the Search dialog box will be bypassed and the RCO User Interface will open directly to the active site.

Option 1.4. Connect to the Configuration Dialog

If you want to configure a new profile, or edit an existing profile, you must use the switch **/config** in the command in order to activate the Configure option in the RCO Runtime User Interface.

Do one of the following.

Method 1

1. Place a shortcut of the RCO Runtime User Interface file (rco_ui.exe) on your Windows desktop.
2. Right-click the icon and select Properties from the popup menu to display the RCO Runtime UI Properties dialog box.
3. In the Target field, after the executable, do one of the following.
 - Enter **/config** to connect to the local project.

e.g. " C:\Program Files\Proficy\Proficy CIMPLICITY\exe\rco_ui.exe" /config .

- Enter **/config** and the fully qualified path to a profile file (.rui) to connect to the specified profile.

e.g. " C:\Program Files\Proficy\Proficy CIMPLICITY\exe\rco_ui.exe" /config C:\RCO\Profile1.rui .

4. Click **OK**.
5. Double-click the RCO Runtime User Interface icon.

Method 2

6. Click **Start** on the Windows Task bar.
7. Select Run to open the Run dialog box.
8. Enter one of the following in the **Open** field.
 - **rco_ui /config** to connect to the local project.

e.g. **rco_ui /config** .

- **rco_ui /config** and the fully qualified path to the profile to connect to the specified profile.

e.g. `rco_ui /config C:\RCO\Profile1 .`


A	Connect to local project.
B	Connect to profile (.rui) file.

9. Click **OK**.

Using either method, the Configure option on the RCO UI is enabled. You can either create a new profile or edit an existing profile.

Step 2. Select the Control Site

Commonly when the RCO Runtime User Interface is invoked, the Search dialog box will open showing a list of all the sites configured in the profile or the local project. You can view runtime data for one control site at a time by selecting it from the list in the Search dialog box.

 **Note:** The Search dialog box will be bypassed when:

- A default site has been configured in a profile.
- Only one site has been configured in a profile.
- The local project only has one active site.


1. Open the Search window using one of the connection options.

The Search dialog box will display a list of all the configured sites in the profile or the local project.

Note: Columns can be sorted by the column title that is clicked.

2. Select the site from the list to be viewed in the RCO user interface.

3. Double-click the site or click **OK** to open the Routing Control Object User Interface (RCOUI) window and view runtime data about the site.

 **Note:** The column headings in the Search dialog box are active control buttons that, when clicked, can be used to rearrange the items in the list.

Step 3. Review the RCOUI Features

Step 3. Review the RCOUI Features

The RCOUI provides several features to help you monitor and complete RCO decisions.

1 (page 359)	Status field.
2 (page 359)	Status bar.
3 (page 360)	Control site data.
4 (page 360)	RCOUI toolbar.
5 (page 361)	RCOUI menu bar.
6 (page 362)	RCOUI Complete Decision/Manual Control popup menu.

1. Status Field

The Status field displays the qualified path of the selected control site and its current status. The status of the control site will determine which functions are available from the within the RCOUI window. Functions that are not available appear dimmed on menus and the toolbar.

Valid status for a control site can be:

Status	Description
Enabled	Locations automatically make and execute control decisions and manage production flow.
Disabled	Locations do not execute control decisions or manage production flow. Manual commands are required from the operator.
Suspend	Locations continue to execute current cycle, but will not re-trigger.
Manual Execution	Manual execution is activated.

2. Status Bar

The status bar, located at the bottom of the window, provides additional information about the selected control site. The most common messages include, but are not limited to the following:

- Process ID.
- Waiting on Verification Point.
- Verification Point timeout.
- Control Location currently not available.

- Ready.

3. Control Site Data

The control decisions for the selected control site are displayed in the window in column format. Each column represents a field of data for the line item decisions in the list.

The following information is available:


1	Seq	Number of the decision for the control site. RCO uses a consecutive numbering scheme.	
2	Start Time	Date and time the decision cycle started.	
3	Decision	Decision on which the control cycle is working.	
4	Dec. Attrib.	Value of the decision's attribute.	
5	Type	Source of the decision as follows:	
		Manual	Initiated by a user.
		Automatic	Initiated by RCO.
6	Status	Current status of the decision to include one of the following types:	
		Pending	Control site disabled; decision waiting to be released.
		Execute	Executing.
		Complete	Completed.
		Failed	Failed to execute.
		Unknown	Status of decision is not known.
		In process	Awaiting asynchronous validation.
		Canceled	Canceled.

4. RCOUI Toolbar

The buttons on the toolbar provide one-click access to the most commonly used functions. They are described as follows:

1	Open	Open window displays for opening a profile.
2	Configure	Opens the Configuration dialog box for editing a profile.
3	Enable/Disables Control Site	Toggles the status of the selected control site.
4	Manual Control	Enables manual control decisions. Dimmed when site is disabled.
5	Reset Trigger	Resets a control site trigger source.
6	Manual Trigger	Manually trips the selected trigger.


7	Alarming/Logging	Opens the Alarming/Logging Options dialog box.
8	Cancel Decision	Cancels selected decisions, when 'in process'.
9	Triggers	Opens the Control Site Triggers Display dialog box.
10	Decisions	Opens the Valid Control Site Decisions dialog box for the selected control site.
11	Search	Opens the Search dialog box.
12	Execute Current Decision	Executes the selected decision.

 **Note:** The **Configure** button is only active on the toolbar when the [/config \(page 357\)](#) switch is used on the Command line.

5. RCOUI Menu Bar

You can use the menu options to edit a profile, reset a trigger, perform a search in the RCO database, or display Help for using the RCO Runtime User Interface.

File Menu	Description
Open	Open window displays for opening a profile.
Configure...	Opens the Configuration dialog box for editing a profile.
Exit	Quits the application.
Options Menu	Description
Enable Ctrl Site	Toggles the status of the selected control site.
Suspend Site	Suspends the selected control site. Current cycle continues to execute, but will not re-trigger.
Manual Control	Enables manual control decisions when the control site is DISABLED.
Reset Trigger	Resets a control site trigger source.
Manual Trigger	Manually trips the selected trigger.
Alarming/Logging	Opens the Alarming/Logging Options dialog box.
Cancel Decision	Cancels selected decisions when 'in process'.
View Menu	Description
Toolbar	Check mark indicates toolbar is displayed in the RCO_UI.
Status Bar	Check mark indicates status bar is displayed in the RCO_UI.
Triggers	Opens the Control Site Triggers Display dialog box.
Decisions	Opens the Valid Control Site Decisions dialog box for the selected control site.
Search	Launches the Search dialog box.

 **Note:** The **Configure** option is only active on the menu bar when the [/config switch \(page 357\)](#) is used on the Command line.

6. RCOUI Complete Decision/Manual Control Popup Menu

In order to access the popup menu from the RCO Runtime UI, place the cursor in the Status box and right-click.

Complete Decision	Complete the pending decision for the control site.
Manual Control	Opens the Manual Control dialog box. The site must be disabled for this option to be active on the menu.

Step 4. Create or Open an RCO Profile

Step 4. Create or Open an RCO Profile

Option 4.1 (page 362)	Open a profile.
Option 4.2 (page 363)	Create a new RCO profile automatically
Option 4.3 (page 363)	Create or edit an RCO profile using a text editor

Option 4.1 Open a Profile in the RCOUI

- Do one of the following:
 - Select File>Open on the RCOUI window menu bar or
 - Click .

The Open dialog box opens.

- Select the profile (.rui) file you want to open.
- Click **Open**.

The Search dialog box opens listing the control sites configured for the profile.

- Select a control site from the list to view in the RCO Runtime User Interface.

 **Note:** If there is only one active control site in the profile, or a default site is configured, the Search dialog box will be bypassed and the RCO User Interface will open directly to the active site.

Option 4.2. Create a New RCO Profile Automatically

1. Click Start on the Windows task bar.
2. Select Run... on the Start menu.

The Run dialog box opens.

3. Enter `rco_ui /config` in the **Open** field.

Where

`rco_ui /config`, which adds extra security to limit access to creating additional profiles, is required to enable the Add Profile option in the RCOUI window.

4. Click OK.

If:	Then:
No project is running.	a Select CIMPLICITY Project dialog box opens for you to start a project.
More than one project is running locally.	a Select project to connect dialog box opens for you to select the project you want to work with first.
One project is running locally.	RCOUI and RCO Runtime UI windows open for that project.

Option 4.3. Create or Edit an RCO Profile Using a Text EditorOption 4.3. Create or Edit an RCO Profile Using a Text Editor

You can create an RCO Profile using the text editor of your choice, or in the Runtime User Interface. In your profile, you can choose to add sites from one project or from multiple projects.


In order to connect to multiple projects, they must be enabled for broadcast over the network.

1. Open a text editor, for example Notepad.
2. Enter lines for the SYSTEM section, as follows

SYSTEM Section	Enter
Line 1	[SYSTEM] Where SYSTEM is the heading enclosed in brackets. This section specifies the configuration of the profile.
Line 2 (Optional)	AUTOMATICREFRESH=Yes or No Where

Yes	Enables the RCO UI to automatically refresh the decision list. (Default)
-----	--

No	Disables the feature and force manual updates.
----	--


Line 3 (Optional)	<code>REFRESHRATE= n</code> Where n =Value in seconds to automatically refresh the decision list. Minimum value = 7 (Default)
Line 4	<code>NUMSITES=n</code> Where n =The number of sites you plan to add to this profile.  Tip: If you do not know the number of sites, leave the value blank for now and fill in later.
Line 5 (Optional)	<code>DEFAULTSITE==\\PROJECT1\SITE1=</code> Where <code>\\PROJECT1\SITE1=</code> is the fully qualified project name and site name of the control site that you want to automatically default to when you access the RCO UI using this profile. When this profile is accessed, the RCO UI will open directly to the control site designated in this field.

3. Press **Enter** twice to leave a blank line between the sections.

4. Enter lines for the SITES section as follows.

SITES Section	Enter
Line 1	<code>[SITE]</code> Where SITE is the heading enclosed in brackets. This section specifies the project or projects and their corresponding sites.
Line 2	<code>\\PROJECTNAME\SITENAME=\\PROJECT1\SITE1=</code> Where <code>\\PROJECT1</code> is the fully qualified project name SITE1 is the site name = (Optional) Enter a description after the = Note: The site names correspond to the names that display in the Tracker window left pane.
Lines 3 - n	Repeat entering all of the sites that should be included in the profile.

5. Update the NUMSITES field with the accurate site count.

 **CAUTION:** If the value of the **NUMSITES** field is less than the actual number of sites listed, the Search dialog box will only display the number of sites indicated in the field. When updating the profile using the text editor, be sure to update the **NUMSITES** field with the accurate site count.

6. Select File>Save to open the Save as dialog box.

7. Select the directory where the profile is to be stored.

8. In the **File Name** field, enter the name of the profile and **.rui** as the extension.

The file must be an **.rui** file.

9. Click **Save** to save your profile.

Example. Site Names in a Manually Created Profile

1. You can find the site names in the Tracker window for each project you are including, when you create a profile in Notepad.

Sites for a profile can be found in the Tracker window:	
A	Project name.
B	Site list and number of sites in the profile.

2. The sites appear in the profile's Search window.

Step 5. Configure an RCO Profile

Step 5. Configure an RCO Profile

Option 5.1 (page 365)	Add and delete sites in a profile.
Option 5.2 (page 366)	Set and delete default sites in a profile.
Option 5.3 (page 366)	Save a configured profile.
Option 5.4 (page 366)	Change the status of a control site.
Option 5.5 (page 367)	Save a configured profile.

Option 5.1. Add and Delete Sites in a Profile


1. Click **Add** to display the Add Sites to profile dialog box.
2. Select a **Project** from the drop-down list. Note that in order to access the project, it must be broadcast over the network, and you may be required to have a user name and password.
3. Select the site to be added to the profile. Use the **Ctrl** key to select multiple sites.
4. Click **Add Selected** to move the selected sites into the Profile file box.
5. Repeat steps 2-4 to add sites from other projects.
6. Click **Done** when you are through adding sites to the profile.

Delete Sites from a Profile

7. Select the site to be deleted from the list. Use the **Ctrl** key to select multiple sites.
8. Click **Delete** to display a message box.
9. Click **Yes** to delete the sites from the profile, or click **No** to cancel the operation.

Option 5.2. Set and Delete the Default Site in a Profile

1. Select the site to be used as the default site in the profile. Note that when this profile is accessed, the Search dialog box will be bypassed and the default site will display automatically.
2. Click **Set default site** and the qualified path to the site will display in the **Default site** field.

 **Note:** You can override the default site set in the profile by repeating the above procedure. It is not necessary to delete the default site first and then set a new one.


Delete a Default Site from a Profile

3. Place the cursor in the **Default site** field.
4. Click **Delete default site** and the site will be removed from the **Default site** field.

 **Note:** When the cursor is placed in the **Default site** field, the **Set default site** button toggles to **Delete default site** automatically.

Option 5.3. Configure Settings for a Profile

1. Select the Settings tab.
2. Check **Refresh decision list** to enable automatic updates of the information in the RCO UI for the configured control sites.
3. Type in or use the arrows to set the **Refresh rate** in seconds to indicate how often the RCO UI should automatically update information.

 **Note:** The buttons on the bottom of the window perform the same functions in the Sites tab.

Option 5.4. Change the Status of a Control Site

Option 5.4. Change the Status of a Control Site

1. Make sure the site is enabled.
2. Select Options>Suspend on the RCOUI menu bar.
3. Make sure the site is enabled.
4. Select Options>Enable Manual Execution on the RCOUI menu bar.

A user can complete a decision.

Status Error Conditions

If you cannot change the status of the control site, the RCO system will generate an error message.

If the following error message displays:	Then:
Selected Control Site is not currently accessible	The RCO process required to carry out the request is not currently online. This could indicate a problem with RCO.
Status Change or Cancel currently pending	The request to change the status at this location has already been submitted, or the current status change is pending completion of the control cycle in progress.

Option 5.5. Save a Configured Profile

1. Click **Save profile** to display the Save dialog box.
2. Type the name of the profile in the **File name** field.
3. Navigate to the folder in your directory to store the profile.
4. Click **Save** to save the profile file in the selected directory.

Step 6. Manage Decisions

Step 6. Manage Decisions

The RCOUI window provides the tools to view and manage decisions for a control site.

You can:

Option 6.1 (page 368)	View valid control site decisions.
Option 6.2 (page 368)	Perform manual decisions.
Option 6.3 (page 369)	Complete a decision manually.
Option 6.4 (page 369)	Cancel decisions.

Option 6.1. View Valid Control Site Decisions

1. Do one of the following:

Method 1. View menu

- a. Click View on the menu bar.
- b. Select **Decisions**.

Method 2. Quickest

Click the **Decisions** button on the toolbar.

The Valid Control Site Decisions dialog box opens when you use either method.

A	Checked enables/cleared disables decision.
B	Checked disables for only one cycle.

2. Use the function buttons and check boxes to control decisions as follows:

To...	Then...
Disable one decision,	Clear the checkbox next to the decision.
Disable decisions for one cycle	Check the Decision(s) disabled for only one cycle box.
Disable all decisions,	Click Disable All .
Enable one decision,	Check the box next to the decision.
Enable all decisions,	Click Enable All .

Option 6.2. Perform Manual DecisionsOption 6.2. Perform Manual Decisions

1. (If the site is enabled) click on the toolbar to disable the control site and enable manual control.

2. Do one of the following.

- Click on the toolbar to display the Manual Control dialog box.
- Select Options>Manual Control on the RCOUI window menu bar.
- Right-click the Status field area and select Manual Control from the popup menu.

The Manual Control dialog box displays when you use any method.

3. Select a **Decision to** execute from the drop-down list.

4. (Optional) Enter an attribute, in the **Attribute** field, to be examined by the function block.

The decision will display in the main window.

Decision Error Conditions

When performing a manual decision, certain conditions within the RCO system will generate an error message.

If the following error message displays	Then
Selected Control Site is not currently accessible	The RCO process required to carry out the request is not currently online. This could indicate a problem with RCO.

Option 6.3. Complete a Decision Manually

1. Enable manual execution.
2. Right-click the **Status** field area in the RCOUI window.
3. Select Complete Decision from the popup menu.

The Complete Decision message box opens.

4. Click OK to complete the decision.

Option 6.4. Cancel Decisions

1. Select the decision.
2. Do one of the following to cancel a decision in the RCOUI window.
 - Select Options>Cancel Decision on the menu bar.
 - Click the **Cancel Decision** button on the toolbar.

Using either of the above methods, the selected decision will stop executing at the control site.

Step 7. Manage Triggers

Step 7. Manage Triggers

Using the Control Site Trigger Display dialog, you can view and manage triggers for a control site.

Option 7.1 (page 370)	Display the control site trigger display dialog box.
Option 7.2 (page 370)	Review trigger details.
Option 7.3 (page 371)	Review trigger points details.
Option 7.4 (page 371)	Reset a trigger manually.
Option 7.5 (page 371)	Manually trip a trigger.


Option 7.1. Display the Control Site Trigger Display Dialog Box

Do one of the following to open the Control Site Trigger Display dialog box.

- Click the **Triggers** button on the RCOUI window toolbar or
- Select View>Triggers on the RCOUI window menu bar.

Result: The Control Site Trigger Display dialog box opens.

1	Triggers display
2	Ordered points that activate selected trigger.
3	Function buttons

 **Tip:** Click the **Update** button to refresh the status of the Triggers listed in the dialog box.

Option 7.2. Review Trigger Details

The Control Site Trigger Display dialog box provides you with information about the Triggers configured for a control site. The Triggers are listed in the top group of the dialog box in numeric sequence. When a Trigger is selected, its associated points are displayed in the Trigger Points group.

You can view the following details about Triggers in the top group of the dialog box:

Column	Description
Seq	Order of the Triggers in numeric sequence
Trigger Sequence	Trigger ID

Status	Current status of the trigger. Ratio determines the number of points triggered as compared to the number of points configured for the trigger. Example There are 3 configured points – 0/3. Two points have triggered 2/3. Ratio = two of three points have triggered with one point remaining.	
	Status Codes	
	Potential	Trigger values match the trigger sequence so far. Control site will potentially trigger. Additional data values are needed to determine if the site will trigger.
	Eliminated	Trigger values do not match the trigger sequence and therefore the site will not trigger.
	Triggered	Trigger values match the trigger sequence and the site is triggered.

Option 7.3. Review Trigger Points Details

Each Trigger has at least one associated point that must be activated in order to trigger the site. If there is more than one associated point, then all of the points in the sequence must be activated to trigger the source.

In order to view Trigger Points, select the Trigger in the top group of the dialog box. The associated points will display in the Trigger Points group.

Column	Description
Seq	Numeric order of the points as they must be activated to trigger the control site.
Point ID	ID for the associated point.

Option 7.4. Reset a Trigger Manually

1. In the Control Site Trigger Display dialog box, select the Trigger to be reset.
2. Click **Reset Trigger**.

A confirmation box displays.

3. Click **OK** in the confirmation box to reset the trigger sequence. Click **Cancel** to quit the operation.

Option 7.5. Manually Trip a Trigger

1. In the Control Site Trigger Display dialog box, select the Trigger to be manually tripped.
2. Click **Manual Trigger**.


A confirmation box displays.


3. Click **OK** in the confirmation box to set the trigger sequence. Click **Cancel** to quit the operation.

The Trigger will be tripped and the resulting status and point ratio will display.

Step 8. Set Alarming and Logging Options through the RCOUI

You can specify alarming and logging conditions using the Alarming / Logging Options dialog box. Check the options in the alarming group and logging group as needed to keep you apprised of the status of your RCO system during runtime.

 **Important:** Changes made to the Alarming / Logging Options from the RCOUI are not saved to the database. Changes are effected dynamically for this instance of RCOUI only.

 **Note:** Open the [TrackerCfg_UI \(page 348\)](#) to set alarming and logging options that will be saved to the RCO configuration.

Each log entry is limited to 120 characters. Characters that are more than 120 (121+) will be ignored.

Do one of the following:

- Select Options>Alarming/Logging on the RCOUI menu bar.
- Click the **Alarming/Logging** button on the toolbar.

The Alarming / Logging Options dialog box opens.

A	Enables alarming.
B	Selected states have enabled alarms .
C	Log destination.
D	Selected states are logged.

Step 9. Search for Control Sites

1. Do one of the following:
 - Select View>Search on the RCOUI menu bar.
 - Click the **Search** button on the toolbar.

The profile's Search window opens.


2. Select the control site from the list in the Search dialog box.
3. Click **OK** to view its runtime data in the RCOUI, or click **Cancel** to quit the operation.

Step 10. Respond to RCO Error Conditions

If the following error message displays,	Do the following:
General Error Conditions	
If an incorrect value is entered, the RCO system will generate an error message.	
Invalid Selection.	Make the correct selection to carry out the request.
Trigger Error Conditions	
If you cannot open the Control Site Trigger Display dialog box, the RCO system will generate an error message.	
Selected Control Site is not currently accessible	the RCO process required to carry out the request is not currently online. This could indicate a problem with RCO.
Reset Trigger Error Conditions	
If you cannot launch the Control Site Trigger Display dialog box, the RCO system will generate an error message.	
Selected Control Site is not currently accessible.	The RCO process required to carry out the request is not currently online. This could indicate a problem with RCO.
Source is already triggered.	The source is already in a triggered state and cannot be triggered again.
Manual Trigger Error Conditions	
If you cannot manually trip the trigger, the RCO system will generate an error message.	
Selected Control Site is not currently accessible.	The RCO process required to carry out the request is not currently online. This could indicate a problem with RCO.
Source is already triggered.	The source is already in a triggered state and cannot be triggered again.

Relocate RCO and TADB Databases

Relocate RCO and TADB Databases

 **Important:** Due to data incompatibility you cannot use a Microsoft Access database to back up Tracker databases. You must use a Microsoft SQL database. (This includes a CIMPLICITY SQL database.)

You can move or copy the Tracker RCO and TADB SQL databases to different servers and use them as back ups.

Step 1 (page 374)	Shrink the database that will be relocated.
Step 2 (page 374)	Check the database current location.


Step 3 (page 375)	Detach the database from its current location.
Step 4 (page 375)	Copy or move the database to a new location.
Step 5 (page 375)	Attach the database to its new location.

Step 1. Shrink the Database that will be Relocated.

1. Make sure your CIMPLICITY project is stopped.
2. Open SQL Server's Enterprise Manager.
3. Right-click the database you want to relocate.
4. Select All Tasks>Shrink Database on the Popup menus.
5. Enter 0 in the Set the **Maximum free space in files after shrinking** field.
6. Click OK.

A message displays telling you that the database has been shrunk successfully.

Step 2. Check the Database Current Location

 **Note:** This step confirms the current location of the database files that will be relocated.

1. Right-click the database you want to relocate.
2. Select Properties on the Popup menu.
The <Database Name> Properties dialog box opens.
3. Select the Data Files tab.
4. Note the location of your database file(s).
The files have an .mdf extension.
5. Click Cancel.

Step 3. Detach the Database from its Current Location

1. Right-click the database you want to relocate.
2. Select All Tasks>Detach Database on the Popup menus.

A Detach Database dialog box opens.

3. Click OK.

Result: A message displays telling you that the database has been detached successfully.

4. Click OK.

Step 4. Copy or Move the Database to a new Location.

1. Open Windows Explorer.
2. Open the folder that contains the .mdf and .ldf database files identified in the <Database name> Properties dialog box.
3. Copy or move the .mdf and .ldf database files to their new location.

The database is ready to be attached to its new location.

Step 5. Attach the Database to its new Location

1. Open the SQL Server Enterprise Manager in the new server.
2. Right mouse-click Databases.
3. Select All Tasks>Attach Database.

An Attach Database dialog box opens.

4. Click the Browser button to the right of the **MDF file of database to attach** field.
5. Select an .mdf file in the Browse for Existing File browser.
6. Click OK.

The database is attached in the new location.

RCO Function Blocks

RCO Function Blocks

Function blocks are the basic unit of execution in Routing Control Logic and output scripts.

A function block may be used in any type of script; however, some blocks have a specific purpose that limits their usefulness in certain types of modules. For example, a `Select Decision` function block in an output module would serve no purpose. The decision has already been selected by the time the output module is selected. A `Set Point` function block could be useful in either a logic or output module.

Routing Logic Modules (RLM) are used to select decisions. The list of available decisions is controlled by Routing Control Configuration. A logic module can only select or eliminate a decision in one specific site, it may not select or eliminate decisions in more than one site.

RCO supports the following types of function blocks:

BCO function blocks (available with Order Execution Mgt.).
Conditional function blocks.
Core function blocks.
Diagnostic function blocks.
Include file.
Output function blocks.
PRT function blocks.
Routing function blocks.
Range Source function blocks (available with Order Execution Mgt.).
RCO TADB function blocks (available with Order Execution Mgt.).

BCO Function Blocks

BCO Function Block List

(Available with Order Execution Mgt.)

BCO function blocks include:

Broadcast Form
Get ASCII Form
Get ASCII Form By Variable

Get WYSIWYG Form
Get WYSIWYG Form By Variable
Send ASCII Form
Send ASCII Form BY Variable
Send Form To Device
Send WYSIWYG Form
Send WYSIWYG Form By Variable
Set Point with ASCII Data
Set Point with ASCII Data Based on Region

Broadcast Form

1. Item Id Broadcast
2. Success/Failure Of Broadcast.

Get ASCII Form

(Available with Order Execution Mgt.)

Description

Get ASCII Form:

1. Fetches the data based on the item ID(s) belonging to a specified Region, Region Location, Item Type and Item Class.
2. Merges the data with the form.

Note: If there is more than one item that meets the specified criteria broadcast will occur for all the item IDs sequentially.

The basic sequence of actions is as follows:

3. Based on the information available in the input form, Tracker queries the Database at runtime (e.g. Query engine, Tracker Attributes, Extended Tracker Attributes).
4. When Tracker successfully gets the data, `Get ASCII Form` creates a file by merging this data with the form.
5. The file created by `Get ASCII Form` is:
 - a. Queued for use through the Windows Print Queue Manager.
 - b. Saved in a configured directory so it can be used for reprint/resend purposes.

Parameters

This function block has the following parameters:

Parameter	Description	
ASCII Form Name	The name of the ASCII form that is merged with the data that has been fetched.	
Region ID	ID of the region where the item is located.	
Region Location	Numeric values only Location of the item in the selected region. Based on the location entered in this field, the following items in the configured region will be broadcast using the supplied form	
	Location	Broadcast Items in the Configured Region
	-3	All the items in all the locations.
	-2	All the items in the first available location.
	-1	All the items in the last available location in the configured Region will be broadcast , using the supplied form.
	0	Invalid.
	Note: For any invalid region location, including 0, or by default, the System will broadcast the first available location in the configured Region using the supplied form.	
Item Class	Class of the item. Note: If Item Class is blank all items in the selected region, region location and of the selected item type will be broadcast.	
Item Type	Type of the item. Note: If Item Type is blank all items in the selected region, region location and of the selected item class will be broadcast.	

Get ASCII Form By Variable

(Available with Order Execution Mgt.)

Description

Get ASCII Form by Variable is similar to the Get ASCII Form function block.

The only difference between the two is that Get ASCII Form by Variable:

1. Fetches the data based on the item ID given in the **Value** (Variable) field or by the Item ID stored in the RCO Variable.
2. Merges the data with the form.

Parameters

This function block has the following parameters:

Parameter	Description
ASCII Form Name	The name of the ASCII form that is merged with the data that has been fetched.
Value	Either of the following: <ul style="list-style-type: none"> • An RCO Variable where the item id is stored. • The item ID.

Get WYSIWYG Form

(Available with Order Execution Mgt.)

Description

Get WYSIWYG Form:

1. Fetches the data based on the item ID(s) belonging to a specified Region, Region Location, Item Type and Item Class.
2. Merges the data with the form.

Note: If there is more than one item that meets the specified criteria broadcast will occur for all the item IDs sequentially.

The basic sequence of actions is as follows:

3. Based on the information available in the input form, Tracker queries the Database at runtime (e.g. Query engine, Tracker Attributes, Extended Tracker Attributes).
4. When Tracker successfully gets the data, Get WYSIWYG Form creates a file by merging this data with the form.
5. The file created by Get WYSIWYG Form is:
 - a. Queued for use through the Windows Print Queue Manager.
 - b. Saved in a configured directory so it can be used for reprint/resend purposes.

Note: Get WYSIWYG Form is similar to the **Get ASCII Form** Broadcast function block.

The only difference between the two is Get WYSIWYG Form is used with WYSIWYG forms.

Parameters

This function block has the following parameters:

Parameter	Description
WYSIWYG Form Name	The name of the WYSIWYG form that is merged with the data that has been fetched.

Region ID	ID of the region where the item is located.	
Region Location	Numeric values only Location of the item in the selected region. Based on the location entered in this field, the following items in the configured region will be broadcast using the supplied form	
	Location	Broadcast Items in the Configured Region
	-3	All the items in all the locations.
	-2	All the items in the first available location.
	-1	All the items in the last available location in the configured Region will be broadcast , using the supplied form
	0	Invalid.
	Note: For any invalid region location, including 0, or by default, the System will broadcast the first available location in the configured Region using the supplied form.	
Item Class	Class of the item. Note: If Item Class is blank all items in the selected region, region location and of the selected item type will be broadcast.	
Item Type	Type of the item. Note: If Item Type is blank all items in the selected region, region location and of the selected item class will be broadcast.	

Get WYSIWYG Form By Variable

(Available with Order Execution Mgt.)

Description

`Get WYSIWYG Form by Variable` is similar to the `Get WYSIWYG Form` function block.

The only difference between the two is that `Get WYSIWYG Form by Variable`:

1. Fetches the data based on the item ID given in the **Value** (Variable) field or by the Item ID stored in the RCO Variable.
2. Merges the data with the form.

Parameters

This function block has the following parameters:

Parameter	Description
WYSIWYG Form Name	The name of the ASCII form that is merged with the data that has been fetched.
Value	Either of the following: <ul style="list-style-type: none"> • An RCO variable where the item id is stored • The item ID.

Send ASCII Form

1. Item Id Broadcast
2. Success/Failure Of Broadcast.

Send ASCII Form By Variable

1. Item Id Broadcast
2. Success/Failure Of Broadcast.

Send Form To Device

1. Item Id Broadcast
2. Success/Failure Of Broadcast.

Send WYSIWYG Form

1. Item Id Broadcast
2. Success/Failure Of Broadcast.

Send WYSIWYG Form By Variable

1. Item Id Broadcast
2. Success/Failure Of Broadcast.

Set Point with ASCII Data

(Available with Order Execution Mgt.)

Description

Set Point with ASCII Data writes a ASCII form data obtained for a particular item to a CIMPLICITY point.

Parameters

This function block has the following parameters:

Parameter	Description
ASCII Form Name	The name of the ASCII form that is merged with the data that has been fetched for a specific Item ID.
Item ID	Select an RCO variable that stores an Item ID as its value, or enter an Item ID. Data for this item is merged with the form.

Point ID	ID of point to be updated. This point must be configured as a string of sufficient length to store the merge result.
----------	--

Set Point with ASCII Data Based on Region

(Available with Order Execution Mgt.)

Description

Set Point with ASCII Data Based on Region writes a ASCII form data obtained for a particular item at a particular region location to a CIMPLICITY point.

Parameters

This function block has the following parameters:

Parameter	Description
ASCII Form Name	The name of the ASCII form that is merged with the data that has been fetched for a specific Item ID.
Region ID	ID of the region where the item is located.
Region Location	Location of the item in the selected region.
Item Class	Class of the item.
Item Type	Type of the item.
Point Id	ID of point to be updated. This point must be an array point that is configured as a string of sufficient length to store the merge result.

Conditional Function Block List

Conditional Function Block List

RCO conditional function blocks include:

Case Block
Case Else Block
Check all region's (related to Decisions) Status (Extended)
Check all Region's (related to Decisions) Status
Check if any Source Region has Decision
Compare Attribute with Attribute
Compare Attribute with Point
Compare Attribute with Value

Compare Point with Point
Compare Point with Value
Compare RCO Variables
Else Block
Else If (Expression)
End If
End Select Block
End While
If (Expression)
If (Solve expression)
Select Block
While (Expression)

Case Block

Description

A BCE **Case** clause of a **Select Case** statement. The function blocks following this block, and preceding the next **Case** , **Case Else** or **End Select** block, are executed if the expression in the block evaluates to true.

Parameters

This function block has the following parameter:

Parameter	Description
Expression	A comma separated list of expressions to be compared against test expression, from the Select case clause, using any of the following syntaxes: expression [,expression] expression to expression is rational_operator expression The resultant type of the expression in expression list must be the same as that of the test expression.

Case Else Block

Description

A BCE **Case Else** clause of a **Select Case** statement. The function blocks following this block and preceding the **End Select** block are executed if no other expression in the block are evaluated to true.

Parameters

This function block has no parameters.

Check all Region's (related to Decisions) Status

Description

Expression block used in the Conditional Blocks. This block will perform a comparison of region status bits. The user can specify whether to use the decision source or destination flag.

Parameters

This function block has the following parameters:

Parameter	Description
Combine with	Used when combining multiple Expression blocks. Defines the Boolean operation to be performed with each Expression Block.
Status To Check	Region status to be checked.
Condition	All Regions Set, Some Regions Set, or No Regions Set.
Region Type	Check each Decision Source or Destination region.

Check All Region's (related to Decisions) Status Pseudo Code

```

If the condition is No regions are set
  Set return value to false
Else
  Set return value to true
End if
RCOGetDecisionList
For each decision
  If using source region
    Set region id to current decision source region
  Else
    Set Region id to current decision destination region
  End if
  Select Case the Condition
    Case "ALL Regions Set"
      If the specified status bit is not set
        Set return value to False
      Exit For
    End If
  Case Some Regions set
    If the specified status bit is Set
      Set return value to True
    Exit For
  
```

```

End If
Case "No regions Set"
  If specified status bit is Set
    Set return value to False
  Exit For
End If
End Select
Next Decision

```

Check all Region's (related to Decisions) Status (Extended)

Description

Expression block used in the Conditional Blocks. This block will perform a comparison of region status bits. The user can specify whether to use the decision source or destination flag.

Parameters

This function block has the following parameters:

Parameter	Description
Combine with	Used when combining multiple Expression blocks. Defines the Boolean operation to be performed with each Expression Block.
Status To Check	Region status to be checked
Condition	All Regions Set, Some Regions Set, or No Regions Set.
Region Type	Check each Decision based on 'SOURCE', 'DEST' or 'SPECIFY' region.
Region ID	Region ID if 'SPECIFY' given in Region Type.

Check all Region's (related to Decisions) Status Pseudo Code

```

If the condition is Some regions are set
  Set return value to false
Else
  Set return value to true
End if
RCOGetDecisionList
For each decision
  If using source region
    Set region id to current decision source region
  ElseIf using specify region
    Set region id to region specified in 'Region ID'
  ElseIf using destination region
    Set Region id to current decision destination region
  End if
  Select Case the Condition
    Case "ALL Regions Set"

```

```

    If the specified status bit is not set
    Set return value to False
    Exit For
  End If
  Case Some Regions set
    If the specified status bit is Set
    Set return value to True
    Exit For
    End If
  Case "No regions Set"
    If specified status bit is Set
    Set return value to False
    Exit For
    End If
  End Select
Next Decision

```

Check if any Source Region has Decision

Description

Returns true if any of the decision's source regions have their 'Waiting For Decision' flag cleared, while the 'Head Item Valid' flag is set.

Parameters

This function block has the following parameter:


Parameter	Description
Combine with	Used when combining multiple expression blocks. Defines the Boolean operation to be performed with each expression block.

Check if any Source Region has Decision Pseudo Code

```

  For All Source Regions of Possible Decisions
  If 'Head Item valid' Flag is set And 'Waiting For Decision' Flag is not
  set
  Return TRUE
  End If
  Next
  Return FALSE

```

 **Note:** This function block should be used before 'Eliminate Decisions Based on Region Status' function block to operate properly.

Compare Attribute

Compare Attribute is obsolete. Use the current function blocks:

- Compare Attribute with Attribute
- Compare Attribute with Point
- Compare Attribute with Value

Compare Attribute with Attribute

Compare Attribute with Attribute

Description

Expression block which compares two attributes. If the attributes meet the comparison criteria, then the condition evaluates to true.

Parameters

This function block has the following parameters:

Parameter	Description
Combine with	Used when combining multiple Expression blocks. Defines the Boolean operation to be performed with each Expression Block.
Region ID	ID of Region where item is located.
Location	Location of the item in the selected region.
Item Class	Class of the item.
Attribute ID	ID of Attribute whose value is to be compared.
Start Character	Starting character within the attribute value.
Attribute Length	Number of characters of attribute to compare. Specify 0 for the entire attribute.
Comparison Operator	Type of Comparison - Greater Than, Less Than, Equals.
Region ID	ID of Region where item is located.
Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute ID	ID of Attribute whose value is to be compared.
Start Character	Starting character within the attribute value.
Attribute Length	Number of characters of attribute to compare Specify 0 for the entire attribute.
Comparison Type	Numeric or alphanumeric comparison. If numeric comparison is selected, values are converted to integers before comparison.

Compare Attribute with Attribute Parameters Example

Parameters

This function block has the following parameters:

Parameter	Value
Combine with	
Region ID	SRC1
Location	1
Item Class	VEHICLE
Attribute ID	COLOR
Start Character	0
Attribute Length	0
Comparison Operator	Equals
Region ID	DST1
Location	PRT_LAST
Item Class	VEHICLE
Attribute ID	COLOR
Start Character	0
Attribute Length	0
Comparison Type	Alphanumeric

The above function block uses Compare Attribute to Attribute, with the specified parameters.

Since the head location of region SRC1 is **YEL** and the tail location of DST1 is **YEL** and the configured comparison operator is Equals, the if statement will evaluate to true and decision SRC1_to_DST1 is selected.

Compare Attribute with Point

Compare Attribute with Point

Description

Expression Block which compares an attribute value to a point value. If the two values meet the comparison criteria, then the condition is considered true.

Parameters

This function block has the following parameters:

Parameter	Description
Combine with	Used when combining multiple expression blocks. Defines the Boolean operation to be performed with each Expression Block.
Region ID	ID of region where item is located.
Location	Location of the item in the selected region.
Item Class	Class of the item.
Attribute ID	ID of attribute whose value is to be compared.
Start Character	Starting character within the attribute value.
Attribute Length	Number of characters of attribute to compare. Specify 0 for the entire attribute.
Comparison Operator	Type of Comparison - Greater Than, Less Than, Equals.
Point ID	ID of CIMPLCITY point whose value is compared to the attribute.
Comparison Type	Numeric or alphanumeric comparison. If numeric comparison is selected, values are converted to integers before comparison.

Compare Attribute with Point Example

Parameters

This function block has the following parameters:

Parameter	Value
Combine with	
Region ID	SRC1
Location	1
Item Class	VEHICLE
Attribute ID	COLOR
Start Character	0
Attribute Length	0
Comparison Operator	Equals
Point ID	COMPARE_POINT
Comparison Type	Alphanumeric

The above function block uses Compare Attribute to Point, with the specified parameters.

Since the head location of region SRC1 is YEL and the point value is **GRN** and the configured comparison operator is Equals, the if statement will evaluate to False and decision SRC1_to_DST1 is not selected.

Compare Attribute with Value

Compare Attribute with Value

Description

Expression Block which compares an attribute value to a value. If the two values meet the comparison criteria, then the condition is considered true.

Parameters

This function block has the following parameters:

Parameter	Description
Combine with	Used when combining multiple expression blocks. Defines the Boolean operation to be performed with each expression block.
Region ID	ID of Region where item is located.
Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute ID	ID of Attribute whose value is to be compared.
Start Character	Starting character within the attribute value.
Attribute Length	Number of characters in attribute to compare. Specify 0 for the entire attribute.
Comparison Operator	Type of Comparison - Greater Than, Less Than, Equals.
Value	User specified comparison value.
Comparison Type	Numeric or alphanumeric comparison. If numeric comparison is selected, values are converted to integers before comparison.

Compare Attribute with Value Example

Parameters

This function block has the following parameters:

Parameter	Value
Combine with	

Region ID	SRC1
Location	1
Item Class	VEHICLE
Attribute ID	COLOR
Start Character	0
Attribute Length	0
Comparison Operator	Equals
Value	YEL
Comparison Type	Alphanumeric

1. The above function block uses Compare Attribute with Value, with the specified parameters.
2. Since the head location of region SRC1 is **YEL** and the comparison value is **YEL** and the configured comparison operator is Equals, the if statement will evaluate to True and decision SRC1_to_DST1 is selected.

Compare Point with Point

Compare Point with Point

Description

Expression Block which compares two points. If the two values meet the comparison criteria, then the condition is considered true.

Parameters

This function block has the following parameters:

Parameter	Description
Combine with	Used when combining multiple Expression blocks. Defines the Boolean operation to be performed with each Expression Block.
Point ID	ID of CIMPLCITY point whose value is to be used in the comparison.
Comparison Operator	Type of Comparison - Greater Than, Less Than, Equals.
Point ID	ID of CIMPLCITY point whose value is to be used in the comparison.
Comparison Type	Numeric or alphanumeric comparison. If numeric comparison is selected, values are converted to integers before comparison.

Compare Point with Point Example

Parameters

This function block has the following parameters:

Parameter	Value
Combine with	
Point ID	COMPARE_1
Comparison Operator	Equals
Point ID	COMPARE_2
Comparison Type	Numeric

1. The above function block uses Compare Point with Point, with the specified parameters.
2. Point **COMPARE_1** is an integer point and has a value of 10.
3. Point **COMPARE_2** is an integer point and has a value of 10.
4. Since both points are numeric, a numeric type comparison is performed between the two values. Since the specified operation is Equals and the points match, the if statement will evaluate to true and decision SRC1_to_DST1 is selected.

Compare Point with Value

Compare Point with Value

Description

Description Expression Block that compares a point value to a user value. If the two values meet the comparison criteria, then the condition is considered true.

Parameters

This function block has the following parameters:

Parameter	Description
Combine with	Used when combining multiple expression blocks. Defines the Boolean operation to be performed with each expression block.
Point ID	ID of CIMPLCITY point whose value is to be used in the comparison.
Comparison Operator	Type of Comparison - Greater Than, Less Than, Equals.
Value	User specified value.
Comparison Type	Numeric or alphanumeric comparison. If numeric comparison is selected, values are converted to integers before comparison.

Compare Point with Value Example

Parameters

This function block has the following parameters:

Parameter	Value
Combine with	
Point ID	COMPARE_1
Comparison Operator	Equals
Value	11
Comparison Type	Numeric

1. The above function block uses Compare Point with Value, with the specified parameters.
2. Point COMPARE_1 is an integer point and has a value of 10.
3. Since the point is numeric, a numeric comparison is performed between the user value of 11 and the point value of 10. Since they don't match, the if expression evaluates to false and decision "SRC1_to_DST1" is not selected.

Compare RCO Variables

Compare RCO Variables

Description

Expression Block which compares two RCO Variables. If the two values meet the comparison criteria, then the condition is considered true.

Parameters

This function block has the following parameters:

Parameter	Description
Combine with	Used when combining multiple Expression blocks. Defines the Boolean operation to be performed with each Expression Block.
RCO Variable 1	First RCO Variable.
Comparison Operator	Type of Comparison - Greater Than, Less Than, Equals.
RCO Variable 2	Second RCO Variable.

Compare RCO Variables Example

Parameters

This function block has the following parameters:

Parameter	Value
Combine with	
RCO Variable 1	XXX
Comparison Operator	Less Than
RCO Variable 2	YYY

1. The above function block uses Compare RCO Variables, with the specified parameters.
2. Since XXX is less than YYY, the condition evaluates to True and Decision SRC1_to_DST1 is selected.

Else Block

Description

Used in conjunction with the If (Expression) Block to start a group of function blocks to be executed when the If condition evaluates to False.

Parameters

This function block has no parameters.

Else If (Expression)

Description


Conditional function block used in conjunction with If (Expression) block. If the first If (Expression) evaluated to false, this block then executes and allows a second condition to be evaluated. The user must provide the conditional expression used in this block. All function blocks between the If (Expression) and the Else or End if Blocks are executed if the expression evaluates to True.

Routing Control has special types of function blocks called "Expression Blocks". Expression Blocks do not show up in Routing Configuration's function block list, but are available to any Conditional Function Block.

Parameters

This function block has the following parameter:

Parameter	Description
Expression	One or more user specified Conditional Function Blocks.

 **Important:** We do not recommend adding more than 3 Conditional Expressions to an If Block.

End If

Description

Used to terminate an If (Expression) block. Each and every If (Expression) block must have a matching End if Block.

Parameters

This function block has no parameters.

End Select Block

Description

Used to Terminate a Select Block. Each Select Block should have a matching End Select Block.

Parameters

This function block has no parameters.

End While

Description

Used to Terminate a **While Block** . Each **While Block** should have a matching **End While** .

Parameters

This function block has no parameters.

If (Expression)

Description

Conditional function block. This block provides the same functionality as the BCE If...Then...Else statement. The user must provide the conditional expression used in this block. All function blocks between the If (Expression) and the Else or End if Blocks are executed if the expression evaluates to True.

Routing Control has special types of function blocks called "Expression Blocks". Expression Blocks do not show up in Routing Configuration's function block list, but are available to any Conditional Function Block.

Parameters


This function block has the following parameter:

Parameter	Description
Expression	One or more user specified Conditional Function Blocks.

Example

The user may select as many Expression Blocks as necessary. Each block is combined with an And / Or.

The first Conditional Expression Block selected has a field for "Combine With". This is not used on the first block in a condition, and should be left blank.

 **Note:** We do not recommend adding more than 3 Conditional Expressions to an If Block.

If (Solve Expression)

(Available with Order Execution Mgt.)

Description

Conditional function block with a Query expression

This block provides the same functionality as the BCE If...Then...Else statement and the Conditional If (Expression) function block. The user must provide query conditional expression to be used in this block. All function blocks between the If (Solve Expression) and the Else or End if Blocks are executed if the query expression evaluates to True.

Parameters

This function block has the following parameter:

Parameter	Description	
Solve Expression	One or more user specified query conditional expressions. Do one of the following. <ul style="list-style-type: none"> Type an existing (valid) query expression. Click the Query Browse button to open the Query Expression Browser to browser for or create a new expression. 	
Region ID	ID of the Region that contains the first item to run the query against.	
Region Location	Specifies the region location.	
	Use	For the
	PRT_FIRST	First item in the region.
	PRT_LAST	Last item in the region

Select Block

Description

A BCE **Select** clause of a **Select Case** statement. This block starts the statement. It must be followed by at least one **Case Block** and an **End Select Block** .

Parameters

This function block has the following parameter:

Parameter	Description
Expression	An RCO Variable or any String or numeric expression. Strings must be in closed in double quotes.

Example

The above function block uses a Select Block and all related blocks.

1. First, load tribute into an RCO Variable. For this example, the Attribute "COLOR" is loaded into RCO Variable 1.
2. The next **Select Block** will perform a comparison on the contents of the RCO Variable. This example uses an RCO Variable, but any string or numeric expression will also work.
3. The next block is a **Case Block** , and it is doing a comparison on "RED" and "GRN". If the RCO Variable contains either of these two values, then the function blocks immediately following the **Case Block** , up to the next **Case Block** , **Case Else** or **End Select** , will be executed. After the statements are executed, execution will jump to the block immediately following the **End Select Block** .
4. The same operation will be performed on the next **Case Block** .
5. The next block is the **Case Else Block** . The function blocks immediately following the **Case Else Block** are executed if none of the other **Case Blocks** in this statement evaluated to True.

6. Finally, the **End Select Block** terminates the **Select Statement** .

While (Expression)

Description

Similar to an If (Expression), except the function blocks between the While (Expression) and the End While block are executed until the condition evaluates to False.


Routing Control has special types of function blocks call "Expression Blocks". Expression Blocks do not show up in Routing Configuration's function block list, but are available to any Conditional Function Block.

Parameters

This function block has the following parameter:

Parameter	Description
Expression	One or more user specified Conditional Function Blocks.

Example

 **Important:** We do not recommend adding more than 3 Conditional Expressions to a **While** Block.

Core Function Blocks

Core Function Blocks List

Core function blocks are:

Append a Decision in ExtDecBuffer	Set Attribute to Text and Integer
Clear the Cached Items	Set Attribute with Attribute
Consume a Trigger	Set Attribute with Attribute (Ext. Attr.)
Create Item Extended Attribute	Set Attribute with Point
Custom Block	Set Attribute with RCO Variable
Delete at a Decision in ExtDecBuffer	Set Attribute with Value
Delete Item Extended Attribute	Set Auto-trigger Interval
Delete Site Attribute	Set Decision Attribute
Exit Sub	Set Decision Status
Fill Array Point	Set Extended Attribute by Item

Fill Array Point (Ext. Attr.)	Set Extended Attribute with Point
Fill RCO Array Point	Set Extended Attribute with RCO Variable
Flush the ExtDecBuffer	Set Extended Attribute with Value
Get Decision Attribute	Set Item Extended Attribute
Get Decision Status	Set Message Point From RCO Variable
Get Ext Decision List	Set Point with Attribute
Get Item Extended Attribute	Set Point with Extended Attribute
Get Production Data	Set Point with Point
Get Site Attribute	Set Point with Timestamp
Increment Attribute	Set Point with Value
Increment Extended Attribute	Set RCO Variable with Attribute
Increment Point	Set RCO Variable with Expression
Insert a Decision in ExtDecBuffer	Set RCO Variable with Point
Logstatus Generator	Set RCO Variable with a Special Value
Parse Message Point To RCO Variables	Set RCO Variable with Timestamp
Region Full Check	Set RCO Variable with Translate Value
Request External Decisions	Set RCO Variable with Value
Reset a Trigger	Set RCO Variable with Value (Extended)
Select next External Decision	Set Site Attribute
Set Array Point	Set Wait Encountered Flag

Append a Decision in ExtDecBuffer

Description

`Append a Decision in ExtDecBuffer` allows a new decision to be appended in decision buffer in RCO.

Parameters

This function block has the following parameters:

Parameter	Description
INPUT	
Controller	Control location ID of the controller that provided the external decisions. This is used to identify decision buffer in which decisions will be appended.

Decision Name	Name of the decision to be appended.
Decision Attribute	Attribute of decision to be appended.
Req Flag	Request indicator of the decision (0 or 1).
OUTPUT	
Status	Status of the call, could be RCO_SUCCESS, RCO_WARNING or RCO_ERROR. If status is not RCO_SUCCESS, error information is returned in Error member of RcoDecisionBuffer object.

Clear the Cached Items

Description

Clear the Cached Items clears the RCO cache.

Item data for a region location gets cached during the trigger sequence of an RCO site whenever any function requests the item data for that location. If the location data is sought again, it is retrieved from the cache. For each decision, the last item of destination region and the first item of source region are cached by default when the site is triggered.

Parameters

This function block has no parameters:

Consume a Trigger

Description

Consume a Trigger changes the trigger sequence so that it is again waiting for the first point in the trigger sequence from an undefined state.

Under normal operating procedures, once a site begins execution by a trigger, the trigger sequence is in an undefined state. It is effectively waiting for the next point directly after the last point in the trigger sequence.

Only one trigger sequence per site may be buffered.

Triggers are only buffered if the site has unexpected point processing enabled.

Parameters

This function block has the following parameter.

Parameter	Description

Trigger Name	Name of the trigger sequence to be consumed. If the parameter is blank, all triggers are consumed.
--------------	--

Create Item Extended Attribute

Description

Adds a CIMPLICITY extended attribute to a PRT item.

Parameters

This function block has the following parameters:

Parameter	Description
Item ID	Specify an Item ID. Select an RCO variable that is set to the ID of the item you want, or enter an Item ID.
Attribute Id	Specify extended attribute to add.
Attribute Value	Specify extended attribute value.

Custom Block

Description

Custom Block, which is an advanced function block, inserts custom BCE code into a Logic or Output module.

Parameters

This function block has the following parameter:

Parameter	Description
Code	Enter BCE code that is to be inserted directly into the module.

Delete a Decision in ExtDecBuffer

Description

Delete a Decision in ExtDecBuffer deletes a decision from the decision buffer. If two or more decisions are in buffer with same name, the first decision with that name is deleted. If no decision are found with the name an error is generated.

The difference between **Delete at a Decision in ExtDecBuffer** and **Delete a Decision in ExtDecBuffer** is that the former takes the offset; the latter takes the name.

Example

The following are the decisions.

Offset	Name
1	Decision Left
2	Decision Right
3	Decision Up
4	Decision Down

Delete a Decision in ExtDecBuffer would take Decision Right as a parameter.

Delete at a Decision in ExtDecBuffer would take the offset 2 as the parameter.

Parameters

This function block has the following parameters:

Parameter	Description
INPUT	
Controller	Control location ID of the controller that provided the external decisions. This is used to identify decision buffer from which decision will be deleted.
Decision Name	Name of the decision to be deleted.
OUTPUT	
Status	Status of the call, could be RCO_SUCCESS, RCO_WARNING or RCO_ERROR, If status is not RCO_SUCCESS then error information is returned in Error member of RcoDecisionBuffer object.

Delete at a Decision in ExtDecBuffer

Description

Delete at a Decision in ExtDecBuffer deletes a decision from the decision buffer. If two or more decisions are in buffer with same offset of the decision, the first decision with that offset is deleted. If no decisions are found with the offset, an error is generated.

The difference between **Delete at a Decision in ExtDecBuffer** and **Delete a Decision in ExtDecBuffer** is that the former takes the offset; the latter takes the name.

Example

The following are the decisions.

Offset	Name
1	Decision Left
2	Decision Right
3	Decision Up
4	Decision Down

Delete a Decision in ExtDecBuffer would take Decision Right as a parameter.

Delete at a Decision in ExtDecBuffer would take the offset 2 as the parameter.

The two functions delete the same decision.

Parameters

This function block has the following parameters:

Parameter	Description
INPUT	
Controller	Control location ID of the controller that provided the external decisions. This is used to identify decision buffer from which decision will be deleted.
Decision Offset	Offset of the decision to be deleted.
OUTPUT	
Status	Status of the call, could be RCO_SUCCESS, RCO_WARNING or RCO_ERROR, If status is not RCO_SUCCESS then error information is returned in Error member of RcoDecisionBuffer object.

Delete Item Extended Attribute

Description

Remove an extended attribute from a PRT item.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	Specify the Region, Leave blank to use selected decision's Source for the region.
Item ID	Specify Item ID. Select an RCO variable that is set to the ID of the item you want to delete, or enter an Item ID.

Attribute ID	Specify the extended attribute to remove.
--------------	---

Delete Site Attribute

Description

Deletes an attribute globally available to the entire RCO site.

Parameters

This function block has the following parameters:

Parameter	Description
Name	Attribute name associated with the current RCO site.

Exit Sub

Description

Exit Sub , an advanced function block, causes a Logic or Output module to exit immediately.

Parameters

This function block has no parameters.

Fill Array Point

Description

Fill Array Point sets an array point offset with a value or an attribute value.

Parameters

This function block has the following parameters:

Parameter	Description
Point ID	ID of array point to be updated.
Index	Offset of array point to be updated.
From	Source of the update value. Either Attribute or Value.
Value	If From parameter is value, then this is the value to be written to the element.
Region ID	ID of Region where item is located.

Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute ID	ID of Attribute whose value is to be set into the point.
Start Character	Starting character within the attribute value.
Length	Number of characters from the attribute value to use. Enter 0 for all.

Fill Array Point (Ext. Attr.)


Description

Fill Array Point sets an array point offset with a value of a CIMPLICITY standard or extended attribute value.

Parameters

This function block has the following parameters:

Parameter	Description
Point Name	ID of array point to be updated.
Index	Offset of array point to be updated.
From	Source of the update value. Either Attribute or Value.
Set Value	If From parameter is value, then this is the value to be written to the element.
Region ID	ID of Region where item is located.
Region Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute ID	ID of Attribute whose value is to be set into the point.
Attribute Start Byte	Starting byte within the attribute value.
Attribute Length	Number of characters from the attribute value to use. Enter 0 for all.

 **Note:** All the function blocks that deal with extended attributes can be provided `INTERNALHOLD` text to read the value of an internal hold.

Values are:

0	Internal Hold is not set.
1	Internal Hold is set.

However, you cannot use any of the extended attribute function blocks to modify system attributes.

Fill RCO Array Point

Description

Fill RCO Array Point sets a single element of an array point. The array point is not updated until the Set Array Point function block is executed.

Parameters

This function block has the following parameters:

Parameter	Description
Point ID	ID of the CIMPLICITY array point, to be updated.
Offset	Offset of the array point to be updated.
Value	Value to be written to the array point. Select an RCO Variable or enter a value.

Flush the ExtDecBuffer

Description

Flush the ExtDecBuffer deletes all the decisions from decision buffer in RCO.

Parameters

This function block has the following parameters:

Parameter	Description
INPUT	
Controller	Control location ID of the controller that provided the external decisions. This is used to identify decision buffer from which decisions will be deleted.
OUTPUT	
Status	Status of the call, could be RCO_SUCCESS, RCO_WARNING or RCO_ERROR. If status is not RCO_SUCCESS, error information is returned in Error member of RcoDecisionBuffer object.

Get Decision Attribute

Description

Get Decision Attribute retrieves the name of the user-defined decision attribute.

Parameters

This function block has no parameters.

Pseudo Code

Not available for this release.

Return Value

The value of the user-defined decision attribute. The decision attribute can be set from either an RCO_UI manual trigger, or from the **SetDecisionAttribute** function block.

Get Decision Status

Description

Get Decision Status gets the current decision status, which will be a string up to 20 characters long.

This is the decision that is currently being selected by the RCO. You cannot change the decision status until a decision has been selected. If no decision has been selected yet (Many decisions still possible) or all decisions have been eliminated (None still possible) then nothing happens when the block is executed.

Parameters

This function block has no parameters.

Pseudo Code

Not available for this release.

Return Value

None

Get Ext Decision List

Description

Get Ext Decision List populates the RcoDecisionBuffer basic object with decision data stored in Decision buffer in RCO. The decisions can be accessed using DecArray(Index) function of RcoDecisionBuffer.

Parameters

This function block has the following parameters:

Parameter	Description
INPUT	
Controller	Control location ID of the controller that provided the external decisions. This is used to identify decision buffer from which decisions will be retrieved.
OUTPUT	
Status	Status of the call, could be RCO_SUCCESS, RCO_WARNING or RCO_ERROR. If status is not RCO_SUCCESS, error information is returned in Error member of RcoDecisionBuffer object.

Get Item Extended Attribute

Description

Get Item Extended Attribute retrieves the value of an extended attribute belonging to a specific item.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	Specify the Region, Leave blank to use the selected decision's Source as the region.
Item ID	Specify the ID of the item that has an extended attribute. Select an RCO variable that is set to the ID of the item you want, or enter an Item ID.
Attribute ID	Specify the name of the extended attribute you want to get.
Item Expression	(Optional) A simple expression; i.e. one or more sequential characters that appear in the Item ID. Note: This parameter is not used if an Item ID is specified.
Name Expression	(Optional) A simple expression; i.e. one or more sequential characters that appear in the Attribute ID. Note: This parameter is not used if an Attribute ID is specified.
Value Expression	(Optional) A simple expression; i.e. one or more sequential characters that appear in the attribute value. Note: This parameter is not used if an Attribute Value is specified.
Attribute Value	(Optional) Specify attribute value. Select a RCO variable set to value you want an attribute to match, or enter a value.

Get Production Data

Get Production Data is a reserved function. Consult the factory for use.

Get Site Attribute

Description

Stores the value of a globally available variable to the entire RCO site.

Parameters

This function block has the following parameters:

Parameter	Description
Name	Specify the name of the site attribute.
Value	Select an RCO variable where you want to store the value obtained from the site variable.

Increment Attribute

Description

Increment Attribute increments an attribute by a value.

Parameters

This function block has the following parameters:


Parameter	Description
Region ID	ID of Region where item is located.
Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute ID	ID of Attribute whose value is to be set into the RCO Variable.
Quantity	Amount to increment the attribute.

Pseudo Code

```

Get item
If item found
  Get attribute value
  If attribute is found
    Convert the attribute value to numeric
    Add quantity to numeric attribute value
    Write attribute value back to item
  End if
End if

```

 **Note:** The attribute value is converted to numeric using the rules of the Basic "Val" function.

The value of the attribute is overwritten with the new value. If the attribute couldn't be converted to a numeric value, then the attribute will be updated with the Quantity value.

Increment Extended Attribute

Description

Increment Attribute increments an extended attribute by a value.

Parameters

This function block has the following parameters:


Parameter	Description
Region ID	ID of Region where item is located.
Region Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute ID	ID of Attribute whose value is to be set into the RCO Variable.
Increment by Value	Amount to increment the attribute.

Pseudo Code

```

Get item
If item found
  Get extended attribute value
  If attribute is found
    Convert the attribute value to numeric
    Add quantity to numeric attribute value
    Write extended attribute value back to item
  End if
End if

```

 **Note:** The extended attribute value is converted to numeric using the rules of the Basic "Val" function.

The value of the attribute is overwritten with the new value. If the attribute couldn't be converted to a numeric value, then the attribute will be updated with the Quantity value.

Increment Point

Description

Increment Point increments a point by a value.

Parameters

This function block has the following parameters:

Parameter	Description
Point ID	ID of point to be updated.
Quantity	Amount by which to increment the point. Enter a negative number to decrement the point value.

Insert a Decision in ExtDecBuffer

Description

Insert a Decision in ExtDecBuffer allows a new decision to be inserted in decision buffer in RCO. If index at which decision is required to be inserted is more than size of buffer, it is appended at the end. If index is less than one, it is inserted as first decision.

Parameters

This function block has the following parameters:

Parameter	Description
INPUT	
Controller	Control location ID of the controller that provided the external decisions. This is used to identify decision buffer in which decisions will be inserted.
Decision Name	Name of the decision to be inserted.
Decision Attribute	Attribute of decision to be inserted.
Req Flag	Request indicator of the decision (0 or 1).
Index	Index at which the decision will be inserted.
OUTPUT	
Status	Status of the call, could be RCO_SUCCESS, RCO_WARNING or RCO_ERROR, If status is not RCO_SUCCESS, error information is returned in Error member of RcoDecisionBuffer object.

Logstatus Generator

Description

LogStatus Generator writes a message to the CIMPLICITY status log.

Parameters

This function block has the following parameters:

Parameter	Description
Level	Error level: Success, Warning or Failure.
Procedure Name	Specifies the procedure/module.
Status Message	Text of message to be written to log.
Error Code	Optional error code.
Reference code	Optional Reference Code.
User Value 1	Substituted for the first occurrence of %s in Status Message.
User Value 2	Substituted for the second occurrence of %s in Status Message.
User Value 3	Substituted for the third occurrence of %s in Status Message.
User Value 4	Substituted for the fourth occurrence of %s in Status Message.

Parse Message Point To RCO Variables

Description

Parse Message Point To RCO Variables:

- Has a token list format, which is comma delimited and colon divided.

Example

```
"<name>: <value> , <name>: <value> "
```

- Parses the tokenized contents of a message point.
- Saves the specified tokens to RCO Variables.

Example

1. You have a message point containing 3 tokens:

```
"ID:4001,DESC:Body Carrier,COLOR:Blue"
```

2. Parse Message Point To RCO Variables lets you specify 3 RCO variables:

```
RCO Variable 1, 2, and 3.
```

3. The RCO variable values can be set to the three token values:

"4001", "Body Carrier", and "Blue".

Parameters

This function block has the following parameters:

Parameter	Description	
Message Source Type	Type of point name that will be used. Options are:	
	RCO Id + Suffix	Message point will consist of an RCO site with a specified suffix. A Message Point Suffix must be specified if this option is chosen. Example Two sites have the same suffix; RCO1_Message1 RCO2_Message1 This block will work in both sites with exactly the same configuration.
	Select Point	Hard codes a single point as the message source. A Message Point Id must be specified if this option is chosen.
Message Point Suffix	Message point ID suffix. Must be specified if RCO Id + Suffix is selected as the Message Source Type. Example For a point RCO1_Message1 The suffix is Message1.	
Message Point Id	Point ID to use as message source. Must be specified if Select Point is selected as the Message Source Type.	
Token 1 Id	ID of the first token to be written to the RCO variable.	
Token 1 Variable	Variable to set with the value of the first token.	
Token 2 Id	ID of the second token to be written to the RCO variable.	
Token 2 Variable	Variable to set with the value of the second token.	
Token 3 Id	ID of the third token to be written to the RCO variable.	
Token 3 Variable	Variable to set with the value of the third token.	
Token 4 Id	ID of the fourth token to be written to the RCO variable.	
Token 4 Variable	Variable to set with the value of the fourth token.	
Token 5 Id	ID of the fifth token to be written to the RCO variable.	
Token 5 Variable	Variable to set with the value of the fifth token.	
Token 6 Id	ID of the sixth token to be written to the RCO variable	

Token 6 Variable	Variable to set with the value of the sixth token.
------------------	--

Region Full Check

Description

Region Full Check gets the total number of items in a region and in process to that region.

Parameters

This function block has the following parameters:

Parameter	Description
Region Id	ID of the region to check.
Total Items	Variable to store Total items in the region.
In process Items	Variable to store in process items in the region.

Request External Decisions

Description

Request External Decisions requests the number of decisions from an external process which can provide decisions, the process can be a local process or remote process. The received decisions are stored in a buffer that subsequently provides these decisions to RCO. Moreover, RCO can also append, insert, delete and flush these decisions. This function block uses RcoDecisionBuffer.

RequestDecisions call to request decisions. If the call to this function block returns success, then the basic object RcoDecisionBuffer also get refreshed to hold new decisions

Parameters

This function block has the following parameters:

Parameter	Description
INPUT	
Controller	Control location ID of the controller that provides the external decisions. This parameter is used to assign Id to RcoDecisionBuffer object whose RequestDecisions function is called by this function block. For later operations on this buffer, this ID is used to identify the buffer.
Project id/ Process_id	Projectname and Process id combined together using a delimiter '/'. Project id: Name of the project in which External process is running and from which decisions are required. Process _id: Id of External process from which decisions are required. When double clicked on this parameter, a dialog is displayed to select project and external processes.
Decisions attribute	Attribute of requested decisions [OPTIONAL]

Decisions Required	Number of decisions required from external process.
OUTPUT	
Status	Status of the call, could be RCO_SUCCESS, RCO_WARNING, RCO_ERROR, If status is not RCO_SUCCESS, error information is returned in Error member of RcoDecisionBuffer object.

Reset a Trigger

Description

Reset a Trigger resets a trigger so it is waiting for the first point in the trigger sequence.

Parameters

This function block has the following parameters:

Parameter	Description
INPUT	
Trigger Name	Name of the trigger or blank for all triggers.
OUTPUT	
Trigger Sequence(s)	Is reset so it is waiting for the first point again.

Set Array Point

Description

Set Array Point writes the stored array point elements to a CIMPLICITY point.

Parameters

This function block has the following parameters:

Parameter	Description
Point ID	ID of the array point.
No Set Elements	Specifies what to do with elements that weren't updated. Clear : Non set elements are set to space or zero. Existing : Non set elements are unchanged.

Select next External Decision

Description

Select next External Decision retrieves the first decision from the decision buffer and executes it if it is also configured in RCOsite executing this function block. If the decision is not configured in RcoSite, it is not executed and an error is generated . The decision also gets removed from buffer after execution of this function block.

If no decisions are left in decision buffer then error is returned indicating no decisions left in buffer. At this point Request External Decisions function block is required to be executed if more decisions are sought from buffer. However, user can also append/insert decisions using Append and Insert function blocks.

Parameters

This function block has the following parameters:

Parameter	Description
INPUT	
Controller	Control location ID of the controller that provided the external decisions. This is used to identify decision buffer from which decisions will be selected.
OUTPUT	
Status	Status of the call, could be RCO_SUCCESS, RCO_WARNING or RCO_ERROR, If status is not RCO_SUCCESS, error information is returned in Error member of RcoDecisionBuffer object.

Set Attribute

Set Attribute is obsolete. Use the current function blocks:

Set Attribute with Attribute
Set Attribute with Point
Set Attribute with Value

Set Attribute to Text and Integer

Set Attribute to Text and Integer is obsolete. Use **Set Point with Attribute** instead.

Set Attribute with Attribute

Description

Set Attribute with Attribute updates an attribute value with another attribute value.

Parameters

This function block has the following parameters:


Parameter	Description
Destination Region ID	ID of Region where item is located.
Destination Location	Location of the item in the Region.
Destination Item Class	Class of the item.
Destination Attribute ID	ID of Attribute whose value is to be set.
Source Region ID	ID of Region where item is located.
Source Location	Location of the item in the Region.
Source Item Class	Class of the item.
Source Attribute ID	ID of Attribute whose value is to be used to set the Destination Attribute ID.
Source Attribute Start Char	Specify the start character if setting a substring of the source Attribute.
Source Attribute Length	Specify the number of characters to set starting from Source Attribute Start Char.

Pseudo Code

```

Read source attribute value
If attribute is found
  Read destination attribute
  If attribute is found
    Update destination attribute value with the source attribute value.
  End if
End if


```

 **Note:** System attributes cannot be modified.

Set Attribute with Attribute (Ext. Attr.)

Description

Set Attribute with Attribute (Ext. Attr.) updates a standard or extended attribute value of one item with the value of a standard or extended attribute belonging to another item.

 **Note:** To set an extended attribute value to the value of a standard attribute, or vice-versa, use an intermediate RCO variable.

Parameters

This function block has the following parameters:

Parameter	Description
Destination Region ID	ID of Region where item is located.
Destination Location	Location of the item in the Region.
Destination Item Class	Class of the item.
Destination Attribute ID	Name of the standard or extended attribute whose value is to be set.
Source Region ID	ID of Region where item is located.
Region Location	Location of the item in the Region.
Source Item Class	Class of the item.
Source Attribute ID	Name of the standard or extended attribute whose value is to be used to set the Destination Attribute ID.
Attribute Start Byte	Specify the start character if setting a sub-string of the source Attribute.
Attribute Length	Specify the number of characters to set starting from Source Attribute Start Byte.

Pseudo Code

```

Read source extended attribute value
If extended attribute is found
  Read destination extended attribute
  If extended attribute is found
    Update destination extended attribute value with the source extended
    attribute value.
  End if
End if

```

Note:

- All the function blocks that deal with extended attributes can be provided `INTERNALHOLD` text to read the value of an internal hold.

Values are:

0	Internal Hold is not set.
1	Internal Hold is set.

However, you cannot use any of the extended attribute function blocks to modify system attributes.

- System attributes cannot be modified.

Set Attribute with Point


Description

Set Attribute with Point writes a point value to an attribute value.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	ID of Region where item is located.
Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute ID	ID of Attribute whose value is to be set.
Point ID	Point ID whose value is written to the specified attribute.

 **Note:** System attributes cannot be modified.

Set Extended Attribute with RCO Variable

Set Attribute with RCO Variable is obsolete. Use **Set Attribute with Value** .

Set Attribute with Value


Description

Set Attribute with Value updates an attribute with a value.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	ID of Region where item is located.
Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute ID	ID of Attribute whose value is to be set.
Value	Value or RCO Variable which is written to the specified attribute.

 **Note:** System attributes cannot be modified.

Set Auto-trigger Interval

Description

Set Auto-Trigger Interval changes the timed trigger interval. This controls the elapse time in seconds, since the last trigger, before a site will automatically trigger.

Parameters

This function block has the following parameters:

Parameter	Description
New Interval	Select Disable or enter a new interval.

Set Decision Attribute

Description

Set Decision Attribute sets the default decision attribute, which can hold a 32-character string defined by the user.

Parameters

This function block has the following parameters:

Parameter	Description
Value	User define 32-character string. Value can be typed in or can be set from an RCO Variable.

Pseudo Code

Not available for this release.

Return Value


None.

Set Decision Status

Description

Set Decision Status sets the current decision status, which can hold a 16-character string defined by the user.

This is the decision that is currently being selected by the RCO. You cannot change the decision status until a decision has been selected. If no decision has been selected (Many decisions still possible) or all decisions have been eliminated (None still possible) then nothing happens when the block is executed.

 **Note:** The decision status is displayed in the RCO_UI decision history list.

Parameters

This function block has the following parameters:

Parameter	Description
Value	User defined 20-character string. Value can be typed in or can be set from an RCO Variable.

Pseudo Code

Not available for this release.


Return Value

None.

Set Extended Attribute by Item

Description

Set Extended Attribute by Item updates an extended attribute value of a PRT item with the value or RCO variable.

 **Note:** You can search for the item region across multiple projects. Enter a comma-delimited list of project names where the item may be found into the **Project(s)** field and recompile.

Parameters

This function block has the following parameters:

Parameter	Description
Item Id Variable	Item Type. Do one of the following. <ul style="list-style-type: none"> • Enter an Item Type. • Select an RCO variable that was previously set to an Item Type name.

Attribute Id Variable	Name of the extended attribute whose value is to be set. Do one of the following. <ul style="list-style-type: none"> • Enter the name of an extended attribute. • Select an RCO variable that was previously set to an attribute name.
Value Variable	The value to set extended attribute to. Do one of the following. <ul style="list-style-type: none"> • Enter a value. • Select an RCO variable that was previously set to an appropriate value.
Project(s)	(Optional) Names of one or more projects in which the selected region may be found. Note: Leave this field blank to limit the search for regions to the local project only.

Set Extended Attribute with Point

Description

Set Attribute with Point writes a point value to an extended attribute value.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	ID of Region where item is located.
Region Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute ID	ID of Attribute whose value is to be set.
Point Name	Point ID whose value is written to the specified attribute.

Set Extended Attribute with RCO Value

Description

Set Extended Attribute with RCO Value updates an extended attribute with the value of an RCO variable.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	ID of Region where item is located.

Region Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute ID	ID of Attribute whose value is to be set.
Variable	RCO Variable which is written to the specified extended attribute.

Set Extended Attribute with Value

Description

Set Extended Attribute with Value updates an extended attribute with a value.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	ID of Region where item is located.
Region Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute ID	Name of the extended attribute whose value is to be set.
Attribute Value	Value or RCO Variable which is written to the specified extended attribute.

Set Item Extended Site Attribute

Description

Update a PRT item with user specified information including extended attributes.

Parameters

This function block has the following parameters:

Parameter	Description
Region	Specify the Region, Leave Blank to use Selected Decision's Source.
Item ID	Specify new Item ID, or select an RCO variable set to this value. Note that, for serialized items, Item ID and Reference ID cannot be changed at the same time.
Attribute Id	Specify extended attribute to add.
Attribute Value	Specify extended attribute value, or select an RCO variable set to this value.

Set Message Point From RCO Variable

Description

Set Message Point From RCO Variables:

- Has a token list format, which is comma delimited and colon divided.

Example

```
"<name>:<value>,<name>:<value>"
```

- Sets a message point with the tokenized list of variable values.

Example

1. You have a 3 variables:

```
RCO Variable 1 (ID), 2 (DESC), and 3 (COLOR).
```

2. The RCO variable values are:

```
"4001", "Body Carrier", and "Blue".
```

3. Set Message Point From RCO Variables sets the message point with a tokenized list of values from the RCO variables.

```
"ID:4001,DESC:Body Carrier,COLOR:Blue"
```

Parameters

This function block has the following parameters:

Parameters

This function block has the following parameters:

Parameter	Description	
Message Dest Type	Type of point name that will be used. Options are:	
RCO Id + Suffix	RCO	Message point ID will consist of an RCO site with a specified suffix. A Message Point Suffix must be specified if this option is chosen. Example Two sites have the same suffix; RCO1_Message1 RCO2_Message1 This block will work in both sites with exactly the same configuration.

	Select Point	Hard codes a single point as the message source. A Message Point Id must be specified if this option is chosen.
Message Point Suffix		Message point ID suffix. Must be specified if RCO Id + Suffix is selected as the Message Source Type. Example For a site RCO1_Message1 The suffix is Message1.
Message Point Id		Point ID to use as message source. Must be specified if Select Point is selected as the Message Source Type.
Token 1 Id		ID of the first token to be written to the tokenized list.
Token 1 Variable		Variable to set to the value of the first token.
Token 2 Id		ID of the second token to be written to the tokenized list.
Token 2 Variable		Variable to set to the value of the second token.
Token 3 Id		ID of the third token to be written to the tokenized list.
Token 3 Variable		Variable to set to the value of the third token.
Token 4 Id		ID of the fourth token to be written to the tokenized list.
Token 4 Variable		Variable to set to the value of the fourth token.
Token 5 Id		ID of the fifth token to be written to the tokenized list.
Token 5 Variable		Variable to set to the value of the fifth token.
Token 6 Id		ID of the sixth token to be written to the tokenized list.
Token 6 Variable		Variable to set to the value of the sixth token.

Set Point with Attribute

Description

Set Point with Attribute writes attribute value to CIMPLICITY point.

Parameters

This function block has the following parameters:

Parameter	Description
-----------	-------------

Point ID	ID of point to be updated with attribute value.
Array Offset	Array offset of point, select "No Array" if point is not an array.
Region ID	ID of Region where item is located.
Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute ID	ID of Attribute whose value is to be set into the point
Start Character	Starting character within the attribute value.
Attribute Length	Ending character within the attribute value. Specify 0 for the entire attribute.

Pseudo Code

```
Read Attribute Value
Write attribute value to point
```

Set Point with Extended Attribute

Description

Set Point with Extended Attribute writes a CIMPLICITY extended attribute value to a CIMPLICITY point.

 **Note:** Configure a text point named TRACKER_PROJECTS in the current project if you use this Set Point with Extended Attribute function block.

Parameters

This function block has the following parameters:

Parameter	Description
Point Name	ID of point to be updated with attribute value.
Array Element of the Point	Array offset of point, select "No Array" if point is not an array.
Region ID	ID of Region where item is located.
Region Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute	ID of Attribute whose value is to be set into the point.
Attribute Start Byte	Starting byte within the attribute value.
Attribute Length	Ending character within the attribute value. Specify 0 for the entire attribute.

Pseudo Code

```
Read Extended Attribute Value
Write extended attribute value to point
```

Set Point with Point

Description

Set Point with Point sets a point with the value of another point.

Parameters

This function block has the following parameters:

Parameter	Description
Point ID	Destination point ID.
Array Offset	Array offset of destination point to be set.
Point ID	Source point ID.
Array Offset	Array offset of source point to be read.

Pseudo Code

```
Read point value
Write point value to destination point
```

Set Point with Timestamp

Description

Set Point with Timestamp sets a point with the current time.

Parameters

This function block has the following parameters:

Parameter	Description
Parameter	Description
Point ID	Point to be updated.
Format	Format of the time stamp to be written to the point.

Pseudo Code

```
Get current time
Write current time to specified point
```

Set Point with Value

Description

Set Point with Value writes a value to a CIMPLICITY point.

Parameters

This function block has the following parameters:

Parameter	Description
Point ID	ID of point to be updated.
Value	Value or RCO Variable to be written to the point.

Set RCO Variable with a Special Value

Set RCO Variable with a Special Value is called in a script to set a state or error flag that can be accessed in another script.

INPUT

Use one of the following names of the RCO variable.

Name	Description
Decision Count	Count of decisions in decision list.
Error Flag	VBA Error Flag
Error Class	VBA Error Class
Error Code	VBA Error Code
Error Message	VBA Error Message
Error Source Type	VBA Error Source Type
Error Source Name	VBA Error Source Name
Script Type Logic	Constant specifying a script type of logical module.
Script Type Output	Constant specifying a script type of output module.
Script Type None	Constant specifying a script type of No decisions remaining.
Script Type Many	Constant specifying a script type of many decision remaining.

Script Type Single	Constant specifying a script type of 1 decision remaining.
Point Error: General	General point error.
PRT Error: General	General Tracking Error
PRT Error: No Attrib	Tracking error attribute not found.
PRT Error: No Item	Tracking error Item not found.

Set RCO Variable with Attribute

Set RCO Variable with Attribute

Description

Set RCO Variable with Attribute sets an RCO variable with attribute value.

Parameters

This function block has the following parameters:

Parameter	Description
RCO Variable	RCO Variable to be updated.
Region ID	ID of Region where item is located.
Location	Location of the item in the Region.
Item Class	Class of the item.
Attribute ID	ID of Attribute whose value is to be set into the RCO Variable.
Start Character	Starting character within the attribute value.
Attribute Length	Ending character within the attribute value. Specify 0 for the entire attribute.
Conversion	Specifies the type of conversion to perform on the attribute. RCO Variables are Variant variables. They can be any type, depending on the value assigned. Conversion Type String means the value is stored in the RCO Variable as a string. The other conversion types store the value in the RCO Variable as an Integer.

Set RCO Variable with Attribute Example


Parameters

This function block has the following parameters:

Parameter	Value

RCO Variable	RCO Variable 1
Region ID	SRC1
Location	1
Item Class	Thing
Attribute ID	BLOCK
Start Character	0
Attribute Length	0
Conversion	HEX

1. Value of attribute "BLOCK" is a string "1BDF" which is the hexadecimal representation of 7135
2. The value stored in RCO Variable 1, after the conversion is, 7135 (base 10).

 **Note:** This function uses the Basic PRT Extensions, not the internal RCO Extensions. The PRT information is not retrieved from the Cache, but is gotten directly from PRT.

Set RCO Variable with Expression

Description

Set RCO Variable with Expression sets the specified RCO variable to the basic expression given in the value parameter.


Parameters

This function block has the following parameters:

Parameter	Description
Variable	Specifies what variable to set.
Value	The Basic Expression to be assigned to the specified RCO Variable.

Pseudo Code

```
Set RCOVar# to Value
```

 **Note:** The value must be a valid Basic Expression.

Set RCO Variable with Point

Description

Set RCO Variable with Point sets an RCO Variable with a point value.

Parameters

This function block has the following parameters:

Parameter	Description
Point ID	ID of Point to be used to set RCO variable.
Array Element or Variable	If point is an array, specify array offset or a RCO Variable containing the array element.
RCO Variable	The RCO Variable to be updated.

Set RCO Variable with Timestamp

Description

Set RCO Variable with Timestamp sets an RCO variable with the current time.

Parameters

This function block has the following parameters:

Parameter	Description
Parameter	Description
RCO Variable	RCO Variable to be updated.
Format	Format of the time stamp to be written to the RCO Variable.

Set RCO Variable with Translate Value

Set RCO Variable with Translate Value

Description

Set RCO Variable with Translate Value scans an array for a value and assign the value's offset to a RCO Variable Parameters.

Parameters

This function block has the following parameters:

Parameter	Description
-----------	-------------

Point ID	ID of array point to be scanned.
Search Text	Specifies the text to search for in the array point.
RCO Variable	Specifies what variable to set with the offset of the search text in the array point.

Set RCO Variable with Translate Value Example

Parameters

This function block has the following parameters:

Point Offset	Value
0	YELLOW
1	BLUE
2	GREEN
Parameter	Value
Point ID	ARRAY_POINT_ID
Search Text	GREEN
RCO Variable	RCO Variable 1 Value of RCO Variable 1 after function block is executed, 2.

Set RCO Variable with Translate Value Pseudo Code

```

Read point
For each element in the array point
  Get String Representation
  If current array offset = search text
    Store offset
    Break out of loop
  End if
Next Element
Return offset

```

 **Note:** Comparison is case insensitive.

Set RCO Variable with Value

Description

Set RCO Variable with Value sets an RCO variable with a user specified value.

Parameters

This function block has the following parameters:

Parameter	Description
RCO Variable	The RCO Variable to be updated.
Value	New value to be stored in the RCO Variable.

Set RCO Variable with Value (Extended)

Description

Set RCO Variable with Value (Extended) sets an RCO variable with a user-specified value or another RCO Variable. The block allows substrings to be copied.)

Parameters

This function block has the following parameters:

Parameter	Description
Destination Variable	The RCO Variable to be updated.
Offset	Specifies the offset of the substring in the destination.
Length	Specifies the length of the substring of the destination.
Value	This value or RCO Variable is assigned to the destination RCO Variable. Important: The value needs to be enclosed by quotes if it is not an RCO variable name.
Length	Specifies the length of the substring of the source.

Set Site Attribute

Description

Adds an attribute globally available to the entire RCO site.

Parameters

This function block has the following parameters:

Parameter	Description
Name	Specify the name of the site attribute you want to set.
Value	Select an RCO variable that contains the value to set, or enter a value.

Set Wait Encountered Flag**Description**

Set Wait Encountered Flag sets a flag to cause routing control to run the wait decision.

Parameters

This function block has no parameters.

Diagnostic Function BlocksDiagnostic Blocks List

Following is a list of Diagnostic blocks.

	Dump Available Decisions
	Site Log Generator

Dump Available Decisions**Description**

Dump Available Decisions writes a list of available still possible decisions to the status log.

Parameters

This function block has the following parameters:

INPUT

Tag	String user defined value to be written with decision name.
-----	---

OUTPUT

Still possible decisions are written to the status log.

If no decisions are possible, then No Decisions Left are written.

Site Log Generator**Description**

Site Log Generator generates a line in the CIMPLICITY status log.

Parameters

This function block has the following parameters:

Parameter	Description	
Status Level	Success, warning or failure status message of the entry.	
Procedure Name	Specifies the procedure/module name to label the 40-character logged message with:	
	Message	Specifies the message to be logged. (40-character string).
	Error Code	Specifies an integer to identify a particular message type.
	Reference Code	Specifies an integer to identify a particular message type.
	User Value1	A user defined value to be inserted into the Alarm Message.
	User Value2	A user defined value to be inserted into the Alarm Message.
	User Value3	A user defined value to be inserted into the Alarm Message.
	User Value4	A user defined value to be inserted into the Alarm Message.

OUTPUT

Entry in the CIMPLICITY status log.

Include File

Description

Includes the contents of the specified file into the generated script.


Parameters

This function block has the following parameters:

Parameter	Description
Include File Name	The path of the file to be included.

Include File Pseudo Code

```
#include"c:\proficy\includes\FuncsLib.txt"
```

 **Note:** This function block works in the same way the #include"<file>" workin C/C++. This function block accepts a files that has only simple VB statements, and not the functions / sub-routines declaration. Since the entire contents of the file would be included with the Main function of the generated script.

Output Function Blocks

Output Blocks List

Following is a list of Output blocks:

Attribute Counts
Conditional Alarm with Attribute
Conditional Alarm with Point
Extended Attribute Counts
Generate Alarm
Process Bits & Words
Set Asynch Validation Value
Update Counts
Update Rule Counts (Ext. Attr.)
Update Spacing, Attribute and Total Released Counts
Update Spacing, Attribute and Total Released Counts (Ext. Attr.)

Attribute Counts

Attribute Counts

Description

Attribute Counts provide attribute counts by region. Given an array of Regions and Attributes, count the number of items in the region and update a point value with the count. This can be run from any output module or logic module.

Parameters

This function block has the following parameter:

Parameter	Description
Region ID	Region ID of item.
Location	Location in region to use. If this value is All , then all items in the region will be dumped and copied.
Attribute Name	Name of the array point that lists the attribute that will be compared (Compare value(s) of the corresponding element in the Attribute Value array Point to the attribute value of the head item for each of the source regions).
Attribute Value	Attribute value(s). Each element in the array can contain one or multiple values with a comma separator. Do not include quotes. Example element 0 = 1 ton,3/4 ton element 1 = K

Attribute Counts	The number of jobs released since a job with the corresponding attribute value was released.
------------------	--

Attribute Counts Example

Following are the Region configurations.

Region: ELPO_1

Location	Item Class	Engine	Diesel
1	Vehicle	L05	1 Ton
2	Vehicle	L05	1 Ton
3	Vehicle	L05	2 Ton
4	Vehicle	L09	2 Ton
5	Vehicle	L07	½ Ton
6	Vehicle	L08	¾ Ton

Region: ELPO_2

Location	Item Class	Diesel	Color
1	Vehicle	1 Ton	RED
2	Vehicle	1 Ton	BLUE
3	Vehicle	2 Ton	GREEN
4	Vehicle	2 Ton	WHITE
5	Vehicle	½ Ton	WHITE
6	Vehicle	½ Ton	BLUE

Run the new function block with the following Array Point Values.

Region ID Array	Item Class Array	Attribute Name Array	Attribute Value Array	Count Point ID Array
ELPO_1	Vehicle	ENGINE	L09,L05	CNT_ENGINE
ELPO_1	Vehicle	DIESEL	1 Ton,1/2 Ton	CNT_SIZE
ELPO_2	Vehicle	COLOR	RED	CNT_RED
ELPO_2	Vehicle	COLOR	WHITE	CNT_WHITE

After the function block completes the point value will be as follows:

Point ID	Value
CNT_ENGINE	4

CNT_SIZE	4
CNT_RED	1
CNT_WHITE	2

Attribute Counts Pseudo Code

```

For each element in the Region Array point
  Dump the Region listed in the element in Region ID Array
  Set counter to 0
  For each item in Region that matches the current element in Item Class
  Array
    If (the element has an Attribute matching the current
    element in the Attribute Name Array) than
      If (the attribute value matches the current element in the
      Attribute Value Array) than
        Increment Counter
      End if
    End if
  Next item
  Get the point id of the current element in Count Point ID Array Point
  Set the point value to the counter
Next Region

```

Conditional Alarm with Attribute

Description

Conditional Alarm with Attribute is a group function block to generate an alarm based on a comparison of an attribute value to a RCO Variable.

Conditional Alarm with Point

Description

Conditional Alarm with Point is a group function block to generate an alarm based on a comparison of a point value to an RCO Variable.

Parameters

This function block has no parameters.

Extended Attribute Counts

Description

Extended Attribute Counts provide extended attribute counts by region. Given arrays of extended attribute names and extended attribute values, count the number of items in the region that have a matching extended attribute name and value. The count for each name/value match is stored in the corresponding element of the Count array. This function block can be run from any output module or routing logic module.

Parameters

This function block has the following parameter:

Parameter	Description
Region ID	Region ID of item.
Region Location	Location in region to use. If this value is All , then all items in the region will be dumped and copied.
Attribute Name Array	Name of the array point that stores the names of the extended attributes that will be compared. Note: The values of corresponding elements in the Attribute Value array are compared with the values for that attribute (if found) of items in the specified location(s) within the source region.
Attribute Value Array	Name of the array point that stores the compare value(s). Each element in the array can contain one or multiple values with a comma separator. Do not include quotes. Example element 0 = 1 ton,3/4 ton element 1 = K
Attribute Count Array	Array point containing the actual attribute counts.

Generate Alarm

Description

Generate Alarm generates a CIMPLICITY Alarm

Parameters

This function block has the following parameter:

Parameter	Description
Alarm Name	The id of the Alarm. Must be a valid alarm of type \$CIMBASIC.
Project ID	The project to generate the alarm on. An empty string "" indicates the current project.
Resource ID	Resource ID to generate the alarm against. Used to control routing of the alarm.
Userld	String (optional). The User ID which generated the alarm.
Reference ID	String (optional). A Reference ID used to distinguish two identical alarms.

Alarm Message	Message for the alarm use "%s" for the parameters.
User Value	A user defined value to be inserted into the alarm message.
User Value	A user defined value to be inserted into the alarm message.
User Value	A user defined value to be inserted into the alarm message.
User Value	A user defined value to be inserted into the alarm message.

Process Bits & Words

Process Bits & Words

Description

Process Bits \$ Words populates an RCO Variable with an Attribute value, convert the attribute as requested, and write the converted value to a point.

This function block is a group function block. It consists of two existing function blocks, Set RCO Variable with Attribute Value and Set RCO Array Point. These two blocks are treated like they are separate blocks, except there is a special data entry screen used to collect the data and treat the blocks as a single entity.

Parameters

This function block has no parameters.

Process Bits & Words Example

Following is a Data Entry Screen example.

Set Asynch Validation Value

Description

Set Asynch Validation Value sets the value to be checked of Validation point for asynchronous validation of a decision. This value holds good for the decision in which it is specified and for that particular cycle. This function block will be used in output module of decision for which the value is to be set.

Parameters

This function block has the following parameters:

Parameter	Description
-----------	-------------

Async Validation Value	Value to be checked. This can be specified using a RCO variable or the value directly.
------------------------	--

Update Counts

Update Counts

Description

Update Counts , for the attributes of the vehicle associated with the source region for the executed decision, increments the Group Count Array Point and the total release count.

If the routed Item's attribute value matches the parsed value (from the Attribute Value array point) corresponding to the attribute name (from the Attribute Name array point), the Spacing Count and the Attribute Released Count elements will be incremented by 1.

 **Note:** This block can be used in a merge RCO to update counts by examining the item that has been moved to the destination region.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	Destination region ID for item.
Location	Location in region to use. If this value is All , then all items in the region will be dumped and copied.
Item Class	Class of the item that contains the attribute. Blank will specify all items at the region.
Rule Type	Type of elimination to be performed.
	G Grouping Eliminate if the count exceeds or equals the Rule.
	S Spacing Eliminate if the count is less then the rule.
	P Percentage
Attribute Name	Name of the array point that lists the attribute names to be compared. Compare value(s) of the corresponding element in the attribute value array point to the attribute value of the head item for each of the source regions.
Attribute Value	Name of array points that lists the Attribute value(s). Each element can contain one or multiple values with a comma separator. Do not include quotes. Example Element 0 = 1 ton,3/4 ton Element 1 = K
Rule	(Grouping and Spacing Rules) The number of jobs.
Count	(Grouping and Spacing Rules) The number of jobs released.
Attribute Released Count	The actual number of items released with the corresponding attribute value. (Percentage Rule Only)

Total Released Count	(Percentage Rule) Integer (analog) point. Counter point stores and continues to increment the total number of items released.
----------------------	---

Update Counts Pseudo Code

```

For each array element
  { \\ start for each element in array element
  Get attribute name, Get attribute value, Get Item Class
If attribute name and value are not blank
  { \\ start if attribute name and value are not blank.
  If( Location == All)
  {
  for each locations\\ ( nLocCnt = 1 to MaxLocation)
  {
  Call function UpdateRuleCounters( attrib name, attrib value,
Count (group/spacing) array point[P6], item class, rule count array
point[P7], region id, nLocCnt, attb name array point, attb val array
pt, Rule type array point[P8],Attrib count array point[P9],Total release
count[P10],
array index)
  }\\endifor all locations
  }\\endif all location
  else \\ for head location
  {
  Call function UpdateRuleCounters( attrib name, attrib value,
Count (group/spacing) array point[P6], item class, rule count array
point[P7], region id, Location, attb name array point, attb val array
pt, Rule type array point[P8],Attrib count array point[P9],Total release
count[P10],
array index)
  }
  }//endiif attribute value and name are blank
  }//end for - next array element
  Update array Points Counts
This is a function iterative called within Update Rule Counts function
block.
Function - UpdateRuleCounters - Pseudo Code (this function is called in
Update Rule Counts function block)
Get the items at the decisions location (value passed in)
  If (number of items at head > 0)
  { \\ start if items > 0
  For each item in head location
  { \\ start for each item in location
  Get the attribute value
  if(rule type is group)
  {
  If (the attribute was found and the attribute values match)
  {
  Increment group count[P6].
  }
  }
  else

```

```

        {
            group count[P6] = 0;
        }
    }\\endif rule type group
    else if(rule type is Spacing)
    {
        If (the attribute was found and the attribute values match)
        {
            Spacing count[P6] = 0;
        }
        else
        {
            Increment Spacing count[P6].
        }
    }\\endif rule type spacing

    if (Attribute count point and total release count are configured)
    {
        If (the attribute was found and the attribute values match)
        {
            Increment Attribute count[P9].
        }
    }

    Increment
total release count[P10].
    }
        }\\end for each item
in head location
    }\\endif number of items at head >
0

```

Update Rule Counts (Ext. Attr.)

Description

Update Rule Counts increments the Group Count Array Point and the total release count for the standard and/or extended attributes of the item associated with the source region for the executed decision.


If the value of the routed Item's extended attribute matches a value (from the Attribute Value array point) corresponding to the attribute name (from the Attribute Name array point), the Spacing Count and the Attribute Released Count elements will be incremented by 1.

Parameters

This function block has the following parameters:

Parameter	Description
-----------	-------------

Region ID	Destination region ID for item.		
Region Location	Location in region to use. If this value is All , then all items in the region will be dumped and copied.		
Item Class Array	Name of the array point that lists the classes of the items that have the corresponding attributes. Leave blank to specify all items of any class present in the region.		
Attribute Name Array	Name of the array point that stores the names of the standard and/or extended attributes to be compared. Note: The values of corresponding elements in the Attribute Value array are compared with the values for that attribute (if found) of the item(s) in the source region. Example If the specified Region Location is First, then this function block compares the value of the corresponding element in the Attribute Value Array point to the attribute value of the head item for each source region.		
Attribute Value Array	Name of array points that stores the Attribute value(s). Each element can contain one or multiple values with a comma separator. Do not include quotes. Example Element 0 = 1 ton,3/4 ton Element 1 = K		
Count	(Grouping and Spacing Rules) The number of jobs released.		
Rule	(Grouping and Spacing Rules) The number of jobs.		
Rule Type Array	Type of comparison to be performed.		
	G	Grouping	Eliminate if the count exceeds or equals the Rule.
	S	Spacing	Eliminate if the count is less then the rule.
	P	Percentage	
Attribute Released Count Array	(Percentage Rule Only) The actual number of items released with the corresponding attribute value.		
Total Released Count	(Percentage Rule) Integer (analog) point. Counter point stores and continues to increment the total number of items released.		

 **Note:** All the function blocks that deal with extended attributes can be provided `INTERNALHOLD` text to read the value of an internal hold.

Values are:

0	Internal Hold is not set.
1	Internal Hold is set.

However, you cannot use any of the extended attribute function blocks to modify system attributes.

 **Note:** This block can be used in a merge RCO to update counts by examining the item that has been moved to the destination region.

Update Spacing, Attribute and Total Released Counts

- Description
- Parameters

Description

Updates spacing and attribute count array points and Release count point each time an item is released.

- If the routed Item has an attribute whose:
 - Name is in ATTRIBUTE_NAME Array Point's current element, and
 - Value matches the ATTRIBUTE_VALUE Array Point's current element.

Then

The following happens to the corresponding element of each of the following arrays:

Array Point	Action	Reason for the action
Spacing Count	Initialized to 0.	Number of items released since last matching item was released
Attribute Release Count	Incremented by 1.	Total number of items with the attribute matching released so far.
Total Release	Incremented by 1.	Total number of items released since the start, regardless of a match or not.

Example

An item has:

- An attribute named: COLOR.
- A Spacing rule for the value: GREEN.

Each time a GREEN item is released the:

- **Spacing rule** is set to 0.

0 items of any other color have been released since the GREEN item was released.

- **Attribute Release Count** is incremented by 1.

1 more GREEN item was released.

- **Total Release Count** is incremented by 1.

1 more **item** was released.

Item ID	Attr. Name	Attr. Value	Spacing	Attr. Release Count	Release Count
ITEM1	COLOR	GREEN	0	+1	+1

- If the routed Item does not have an attribute whose:
 - Name is in ATTRIBUTE_NAME Array Point's current element nor a
 - Value that matches the ATTRIBUTE_VALUE Array Point's current element

Then

The following happens to the corresponding element of each of the following arrays,

Array Point	Action	Reason for the action
Spacing Count	Incremented by 1.	Number of items released since last matching item was released
Attribute Release Count	Remains unchanged.	Total number of items with the attribute matching released so far
Total Release	Incremented by 1.	Total number of items released since start, regardless of match or not.

Example

An item has:

- An attribute named: COLOR.
- Spacing rule for the value: GREEN.

Each time a **NON-GREEN item** is released the:

- **Spacing rule** is incremented by 1.

+1 item of any other color has been released since the last GREEN item was released.

- **Attribute Release Count** has No Change..

No additional GREEN item was released.

- **Total Release Count** is incremented by 1.

1 more **item** was released.

Item ID	Attr. Name	Attr. Value	Spacing	Attr. Release Count	Release Count
ITEM1	COLOR	RED	+1	NC	+1

 **Note:** This block can be used in a merge RCO to update counts by examining the item that has been moved to the destination region.

Parameters

This function block has the following parameter:

Parameter	Description
Region ID	Destination region ID for item.
Location	Location in region to use. If this value is All , then all items in the region will be dumped and copied.
Attribute Name Array	Name of the array point that lists the attribute that will be compared (Compare value(s) of the corresponding element in the Attribute Value array Point to the attribute value of the head item for each of the source regions).
Attribute Value Array	Attribute value(s). Each element in the array can contain one or multiple values with a comma separator. Do not include quotes. Example Element 0 = 1 ton,3/4 ton Element 1 = K
Spacing Count Array	The number of items released since an item with the matching attribute value was released. This is initialized to zero every time an item is released with the attribute name and value combination matching the Attribute name and value combination in the Attribute Name and Attribute Value array points
Attribute Release Count Array	The total number of items released, since the start that had a matching attribute name and value combination in the Attribute Name and Attribute Value array points.
Total Released Count	Integer (analog) point. Counter point stores and continues to increment the total number of items released.

Update Spacing, Attribute and Total Released Counts (Ext. Attr.)

- Description
- Parameters

Description

Updates Spacing and Attribute count array points and Release count point each time an item is released.

- If the routed Item has an attribute whose:
 - Name is in ATTRIBUTE_NAME Array Point's current element, and
 - Value matches the ATTRIBUTE_VALUE Array Point's current element.

Then

The following happens to the corresponding element of each of the following arrays:

Array Point	Action	Reason for the action
Spacing Count	Initialized to 0.	Number of items released since last matching item was released
Attribute Release Count	Incremented by 1.	Total number of items with the attribute matching released so far.
Total Release	Incremented by 1.	Total number of items released since the start, regardless of a match or not.

Example

An item has:

- An attribute named: COLOR.
- A Spacing rule for the value: GREEN.

Each time a GREEN item is released the:

- **Spacing rule** is set to 0.

0 items of any other color have been released since the GREEN item was released.

- **Attribute Release Count** is incremented by 1.

1 more GREEN item was released.

- **Total Release Count** is incremented by 1.

1 more **item** was released.

Item ID	Attr. Name	Attr. Value	Spacing	Attr. Release Count	Release Count
ITEM1	COLOR	GREEN	0	+1	+1

- If the routed Item does not have an attribute whose:
 - Name is in ATTRIBUTE_NAME Array Point's current element nor a
 - Value that matches the ATTRIBUTE_VALUE Array Point's current element

Then

The following happens to the corresponding element of each of the following arrays,

Array Point	Action	Reason for the action
Spacing Count	Incremented by 1.	Number of items released since last matching item was released
Attribute Release Count	Remains unchanged.	Total number of items with the attribute matching released so far

Total Release	Incremented by 1.	Total number of items released since start, regardless of match or not.
---------------	-------------------	---

Example

An item has:

- An attribute named: COLOR.
- Spacing rule for the value: GREEN.

Each time a **NON-GREEN item** is released the:

- **Spacing rule** is incremented by 1.

+1 item of any other color has been released since the last GREEN item was released.

- **Attribute Release Count** has No Change..

No additional GREEN item was released.

- **Total Release Count** is incremented by 1.

1 more **item** was released.

Item ID	Attr. Name	Attr. Value	Spacing	Attr. Release Count	Release Count
ITEM1	COLOR	RED	+1	NC	+1

 **Note:** This block can be used in a merge RCO to update counts by examining the item that has been moved to the destination region.

Parameters

This function block has the following parameter:

Parameter	Description
Region ID	Destination region ID for item.
Location	Location in region to use. If this value is All , then all items in the region will be dumped and copied.
Attribute Name Array	Name of the array point that lists the attribute that will be compared (Compare value(s) of the corresponding element in the Attribute Value array Point to the attribute value of the head item for each of the source regions).
Attribute Value Array	Attribute value(s). Each element in the array can contain one or multiple values with a comma separator. Do not include quotes. Example Element 0 = 1 ton,3/4 ton Element 1 = K

Spacing Count Array	The number of items released since an item with the matching attribute value was released. This is initialized to zero every time an item is released with the attribute name and value combination matching the Attribute name and value combination in the Attribute Name and Attribute Value array points
Attribute Release Count Array	The total number of items released, since the start that had a matching attribute name and value combination in the Attribute Name and Attribute Value array points.
Total Released Count	Integer (analog) point. Counter point stores and continues to increment the total number of items released.

PRT Function Blocks

PRT Function Blocks List

Following is a list of PRT function blocks.

Add PRT Item
Associate Item
Delete Item
Disassociate Item
Load PRT Item into Item Variable
Load PRT Region Data into Region Variable
Move Item
Set PRT Item Data
Set PRT Item Data (Ext. Attr.)
Set PRT Item with another PRT Item
Set PRT Item with another PRT Item (Ext. Attr.)
Set Source Region as Decision Made
TADB function blocks related to PRT functions are as follows.
Get TADB Item Attribute
Modify TADB Item Attribute
TADB Commit
TADB Initialize
TADB Set Attribute

Add PRT Item

Add PRT Item

Description

Adds a new Item to PRT.

Parameters

This function block has the following parameters:

Parameter	Description
Item ID	Specify the Item ID for the new item.
Ref Id	Specify the Reference ID for the item.
Item Type	Specify the Item Type or pick from a RCO Variable.
Region ID	Specify the Region in which to create the item.
Region Location	Specify the location in the region where item has to be created.
Comment	Specify the description for the item to be created.

Add PRT Item Pseudo Code:

```

Create a new PrtItem
Assign ItemId
Assign RefId
Assign ItemTypeId
Assign RegionId
Assign RegionLoc
Assign Comment
Assign UserId to SiteName.ScriptName
Assign Status
Add the item

```

Associate Item

Associate Item

Description

Associate two PRT Items. Will move a PRT item into a location already occupied by a second PRT Item. This function block is used in place of a PRT Combine Region.

Parameters

This function block has the following parameters:

Parameter	Description
Source Region ID	Specifies the source region.
Source Region Location	Specifies source region location.
Destination Region ID	Specifies the destination region.
Destination Region Location	Specifies destination region location.
Item Type	Specifies item type to move. [optional].
Item Class	Specifies item class to move. [optional].
Insert	Specifies how the items should be added to the destination region location. Click either Add or Insert .
Auto Associate	Specifies if item(s) should be associated to the first serialized item in the destination region location.

Associate Item Example

Parameters

This function block has the following parameters:

First Blocks Parameters

Parameter	Description
Source Region ID	SRC1
Source Region Location	1
Destination Region ID	DST1
Destination Region Location	PRT_LAST
Item Type	
Item Class	VEHICLE
Insert	Insert
Auto Associate	False

Second Blocks Parameters

Parameter	Description
Source Region ID	SRC2

Source Region Location	1
Destination Region ID	DST1
Destination Region Location	PRT_LAST
Item Type	
Item Class	VEHICLE
Insert	Add
Auto Associate	False

1. An Output Module with two Associate Item Blocks is executed.
2. The first block takes the item from the head location of Region SRC1 and inserts it into the tail location of region DST1.
3. The second block takes the item from the head location of region SRC2 and adds it to the tail location of region DST1.

Associate Item Pseudo Code

```

Locate the PRT Item
If Insert specified
  Insert the item into the specified location of Destination region
Else
  Add the item to the specified location
End if

```

Delete Item

Description

Delete a PRT Item from the tracking model with a supplying disposition and optional comment.

Parameters

This function block has the following parameters:

Parameter	Description	
Region ID	Specifies the region from which the item is to be deleted.	
Region Location	Specifies the region location.	
Item Type	Specifies the item type to be deleted [optional].	
Item Class	Specifies the item class to be deleted [optional].	
Disposition	How the item is to be deleted.	
	Scrap	Scrap the item

	Delete	Delete the item
	No Check Scrap	Scrap the item-overriding region locking status.
	No Check Delete	Delete the item-overriding region locking status.
Comment	Comment to send to log file. [optional].	

Disassociate Item

Description

Move a single item out of a region. This block can be used in place of a PRT disperse region. If the item doesn't exist, raise an exception.

Parameters

This function block has the following parameters:

Parameter	Description
Source Region ID	Specifies the source region.
Source Region Location	Specifies the source region location.
Destination Region ID	Specifies the Destination Region.
Destination Region Location	Specifies Destination Region Location.
Item Type	Specifies item type to move.
Item Class	Specifies item class to move.
Insert	Specifies how the items should be added to the destination region location.

Load PRT Item into Item Variable


Description

Load a BCE PrtItem object with specific item information.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	Specify the Region ID.
Item Class	Specify the item class.
Source Region Location	Select the Region Location or enter a number greater than 1.

 **Note:** The variable name is **item** .

Load PRT Region Data into Region Variable


Description

Load a BCE PrtRegion object with a specific item information. If the region doesn't exist, raise an exception.

Parameters

This function block has the following parameter:

Parameter	Description
Region ID	Specify the Region ID to load.

 **Note:** PrtRegion object name is Region.

Move Item

Move Item

Description

Move a PRT Item. If no item class or item type is specified, then all items at the source region are moved.

Parameters

This function block has the following parameters:

Parameter	Description
Source Region ID	Region from which the item will be moved.
Source Region Location	Location in region from which the item will be moved.
Destination Region ID	Region to which the item will be moved.
Destination Region Location	Location in region to which the item will be moved.
Item Type	(Optional) Item type to move.
Item Class	(Optional) Item class to move.

Insert	How the item should be added to the destination region location. Options are: <ul style="list-style-type: none"> • ADD • INSERT
--------	---

Move Item Pseudo Code

```

If Item type and Item Class were not specified
  For each serialized item at region and location
    IF the item's reference id and item id are not blank
      Move the serialized item to the destination
      Moving the serialized item will also move the non-serialized items
    End if
  Next item
  If no serialized items exist at the location
    For each non serialized item
      Move to the destination location
    Next non serialized item
Else
  For each item of the specified item type or class type
    Move to new location
  Next
End if

```

Set PRT Item Data

Description

Update a PRT item with user specified information.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	Specify the Region, Leave Blank to use Selected Decision's Source.
Region Location	Specify the region location.
Item Class	Specify the item class.
Item ID	Specify new Item ID. Note that, for serialized items, Item ID and Reference ID cannot be changed at the same time.
Reference ID	Specify new Reference ID. Note that, for serialized items, Item ID and Reference ID cannot be changed at the same time.
Item Type	Specify item type.
Status Mask	Specify the mask for the status bits that will be effected by the Status parameter.

Status	Specify the item status.
Set All Attributes To Value1	Sets All Attributes of the item to the Value1 parameter (which defaults to empty string).
Attribute1	Specify attribute to add.
Value1	Specify attribute value.
Attribute2	Specify attribute to add.
Value2	Specify attribute value.
Attribute3	Specify attribute to add.
Value4	Specify attribute value.
Attribute5	Specify attribute to add.
Value5	Specify attribute value.
Attribute6	Specify attribute to add.
Value6	Specify attribute value.
Attribute7	Specify attribute to add.
Value7	Specify attribute value.
Attribute8	Specify attribute to add.
Value8	Specify attribute value.
Attribute9	Specify attribute to add.
Value9	Specify attribute value.
Attribute10	Specify attribute to add.
Value10	Specify attribute value.

Set PRT Item Data (Ext. Attr.)

Description


Update a PRT item with user specified information that includes standard and/or extended attributes.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	Specify the Region, Leave Blank to use Selected Decision's Source.
Region Location	Specify the region location.
Item Class	Specify the item class.

Item ID	Specify new Item ID. Note that, for serialized items, Item ID and Reference ID cannot be changed at the same time.
Reference ID	Specify new Reference ID. Note that, for serialized items, Item ID and Reference ID cannot be changed at the same time.
Item Type	Specify item type.
Status Mask	Specify the mask for the status bits that will be effected by the Status parameter.
Status	Specify the item status.
Set All Attributes To Value1	Sets All Attributes of the item to the Value1 parameter (which defaults to empty string).
Attribute1	Attribute to add.
Value1	Attribute value.
Attribute2	Attribute to add.
Value2	Attribute value.
Attribute3	Attribute to add.
Value4	Attribute value.
Attribute5	Attribute to add.
Value5	Attribute value.
Attribute6	Attribute to add.
Value6	Attribute value.
Attribute7	Attribute to add.
Value7	Attribute value.
Attribute8	Attribute to add.
Value8	Attribute value.
Attribute9	Attribute to add.
Value9	Attribute value.
Attribute10	Attribute to add.
Value10	Attribute value.

 **Note:** All the function blocks that deal with extended attributes can be provided `INTERNALHOLD` text to read the value of an internal hold.

Values are:

0	Internal Hold is not set.
1	Internal Hold is set.

However, you cannot use any of the extended attribute function blocks to modify system attributes.

Set PRT Item with another PRT Item

Set PRT Item with another PRT Item

Description

Copy one PRT Item to another.

Parameters

This function block has the following parameters:

Parameter	Description
Source Region ID	Specify the region. Leave blank to use selected decision's source region.
Source Region Location	Specify the region location.
Source Item Class	Specify the item class.
Destination Region ID	Specify the region. Leave blank to use selected decision's source region.
Destination Region Location	Specify the region location.
Destination Item Class	Specify the item class.
Status Transfer Mask	Specify what item status bits to copy.
Transfer Item Type ID	Specified whether or not to copy the item type ID.
Transfer All Attributes	Specify whether to copy all attributes or just some.
Attribute1	Specify the attribute to copy.
Attribute2	Specify the attribute to copy.
Attribute3	Specify the attribute to copy.
Attribute4	Specify the attribute to copy.
Attribute5	Specify the attribute to copy.
Attribute6	Specify the attribute to copy.
Attribute7	Specify the attribute to copy.
Attribute8	Specify the attribute to copy.
Attribute9	Specify the attribute to copy.
Attribute10	Specify the attribute to copy.

Set PRT Item with another PRT Item Pseudo Code

```

If the specified source region is blank
  Get the region id from the selected decision's source region
Else
  Use the specified source region
End if
If no items found in region at specified location
  Raise exception
End if
If the specified destination region is blank
  Get the destination region id from the selected decisions source region
Else
  Use the specified source region
End if
If the status bit mask is not zero
  'Transfer Status Bits
  Mask% = Not Status Mask
  ' = clear transferred Bits
  Item2Status = Item2Status And Mask%
  ' Copy transferred bits
  Mask% = Item1Status And Status Mask
  ' Transfer the bits
  Item2Status = Item2Status Or Mask%
End if
If transfer all attributes
  For each attribute in source item
    Read attribute
    Write attribute to destination items
  Next
Else
  For each of the 10 possible attributes
    If current attribute id is not blank
      Read source attribute
      Write value to destination attribute
    End if
  End if
End if

```

Set PRT Item with another PRT Item (Ext. Attr.)

Description


Copy one PRT Item with the standard and/or extended attributes of that item.

Parameters

This function block has the following parameters:

Parameter	Description
Source Region ID	Specify the region. Leave blank to use selected decision's source region.
Source Region Location	Specify the region location.

Source Item Class	Specify the item class.
Destination Region ID	Specify the region. Leave blank to use selected decision's source region.
Destination Region Location	Specify the region location.
Destination Item Class	Specify the item class.
Transfer Item Type ID	Specified whether or not to copy the item type ID.
Status Transfer Mask	Specify what item status bits to copy.
Transfer All Attributes	Specify whether to copy all attributes or just some.
Attribute1	Specify the attribute to copy.
Attribute2	Specify the attribute to copy.
Attribute3	Specify the attribute to copy.
Attribute4	Specify the attribute to copy.
Attribute5	Specify the attribute to copy.
Attribute6	Specify the attribute to copy.
Attribute7	Specify the attribute to copy.
Attribute8	Specify the attribute to copy.
Attribute9	Specify the attribute to copy.
Attribute10	Specify the attribute to copy.

 **Note:** All the function blocks that deal with extended attributes can be provided `INTERNALHOLD` text to read the value of an internal hold.

Values are:

0	Internal Hold is not set.
1	Internal Hold is set.

However, you cannot use any of the extended attribute function blocks to modify system attributes.

Set Source Region as Decision Made

Description

This is an output function block. Clears the 'Waiting For Decision' Region flag of the source region of the selected decision.

Parameters

This function block has no parameters.

Set Source Region as Decision Made Pseudo Code:

```
Clear Status Bit PRT_REGION_WAITING_FOR_DEC for Source Region of the
selected decision
```

Routing Function Blocks

Routing Function Blocks List

Routing function blocks include:

Balance Load
Block by Attribute
Block by Extended Attribute
Check Item Hold Status
Check Process Capability
Check Process Capability (Ext. Attr.)
Check Region Full
Check Region Ready
Consume a Trigger
Eliminate by Weight
Eliminate Decision
Eliminate Decision Based on Attribute Spacing
Eliminate Decision Based on Ext. Attribute Spacing
Eliminate Decision Based on Rule
Eliminate Decision Based on Rule (Ext. Attr.)
Eliminate Decision by Attribute
Eliminate Decision by Extended Attribute
Eliminate Decision by Attribute Attribute
Eliminate Decision by Attribute Point
Eliminate Decision by Attribute Value
Eliminate Decision by Ext. Attribute Attribute
Eliminate Decision by Ext. Attribute Point
Eliminate Decision by Ext. Attribute Value
Eliminate Decision by Increasing Weight Percentages

Eliminate Decision by Increasing Weight Percentages (Ext. Attr.)
Eliminate Decision by Point
Eliminate Decision by Point (Ext. Attr.)
Eliminate Decision by Point Point
Eliminate Decision by Point Value
Eliminate Decision by Solve
Eliminate Decision if not only Decision
Eliminate Decisions Based on Region Status
Eliminate Decisions with no Items in Source
Get Trigger Name
Item Type Translation
Look Back Blocking
Look Back Blocking (Ext. Attr.)
Select Decision
Select Decision By Attribute
Select Decision by Attribute Point
Select Decision by Attribute Value
Select Decision By Extended Attribute
Select Decision By Ext. Attribute Point
Select Decision By Ext. Attribute Value
Select Decision by Point Value
Select Decision by Solve
Select Decision Extended
Select Decision with Highest Order & Weight
Select Highest Order Decision
Select Lowest Order Decision
Select Most Empty Region
Select Most Full Region
Select Next Highest Order Decision
Select Oldest Item
Select Oldest Item (Ext. Attr.)

Balance LoadBalance Load**Description**

Balance Load allows a cost to be assigned to a decision. The cost takes into consideration the number of items in the source and destination regions, as well as the region with the most number of items. A threshold is then specified that the cost cannot exceed.

Parameters

This function block has the following parameters:

Parameter	Description	
Always Evaluate	Hard, Soft or Breakable. This parameter controls what happens when the extension eliminates all available decisions.	
	Hard	All decisions will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, None , is executed to handle this condition.
	Soft	Roll back any decisions eliminated by this extension.
	Breakable	Roll back any decisions eliminated by this extension that have a danger point value of TRUE (non zero). The routing modules will first use the danger defined for each individual decision; and, if no danger point has been configured for the decision, then the site-wide danger point will be used. If no danger point can be found for a decision, then the decision remains eliminated.
Threshold Value	Specifies the threshold value to be used in balancing the load.	

Balance Load Example

Decision	Weight	Source Qty	Destination Qty	Formula	Load
SRC1 to DST1	15	20	4	$(15 - 20) + 15 + 4$	14
SRC1 to DST2	15	20	9	$(15 - 20) + 15 + 9$	19
SRC2 to DST1	5	9	4	$(15 - 9) + 5 + 4$	15
SRC2 to DST2	10	9	9	$(15 - 9) + 10 + 9$	25

Max Source region cost of all available source regions is 15.

User Specified Threshold is 16.

1. Decision SRC1 to DST1, Load is 14, Decision is not eliminated because it does not exceed Threshold.

2. Decision SRC1 to DST2, Load is 19, Decision is eliminated because it does exceed Threshold.
3. Decision SRC2 to DST1, Load is 15, Decision is not eliminated because it does not exceed Threshold.
4. Decision SRC1 to DST1, Load is 14, Decision is eliminated because it does exceed Threshold

Balance Load Pseudo Code

```

For each decision
  If decision is not eliminated
    Get value of source region's quantity point from region maintenance
    table
    If (number of items in source region is greater than 0)
      Get the value of destination region quantity point from region
      maintenance
      If the number of items in destination region is greater than 0
        Calculate maximum source cost as follows
        Set nMaxSrcCost to the number of items in source region of
        decision #0
        For each decision from 1 to last
          If (Number of items in current region > nMaxSrcCost)
            Set nMaxSrcCost to number of items in region
          End If
        Next decision
        Set nWeight to current decision weight
        Calculate the load as follows:
        Load = (nMaxSrcCost - # of items in source region)
              + Current Decision weight
              + number of items in destination region
        End if // number of items in destination region > 0
        End if // number of items in source region > 0
        If (Load is >= specified Threshold Value)
          Eliminate the decision
        End if
      End if
    End if
  Next For

```

Block by Attribute

Block by Attribute

Description

Block by Attribute eliminates each decision where the item at the specified source region location does not match the item at the destination region tail location.

Parameters

This function block has the following parameters:

Parameter	Description	
Always Evaluate	Hard, Soft or Breakable. This parameter controls what happens when the extension eliminates all available decisions.	
	Hard	All decisions will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.
	Soft	Roll back any decisions eliminated by this extension.
	Breakable	Roll back any decisions eliminated by this extension that have a danger point value of True (non zero). The routing modules will first use the danger defined for each individual decision; and, if no danger point has been configured for the decision, then the site-wide danger point will be used. If no danger point can be found for a decision, then the decision remains eliminated.
Item Class	All items with matching class will be used.	
Location	Number of locations in source regions to scan.	
Attribute ID	Name of Attribute to compare.	
Start Character	Starting character of the attribute to be used.	
Attribute Length	Specifies the number of characters in the attribute to compare.	
End Character	Ending character of the attribute to be used. Enter zero in both the Start Character and Attribute Length fields to use all characters.	

Block by Attribute Example

Parameters

This function block has the following parameters:


Parameter Name	Parameter Value
Always Evaluate	Hard
Item Class	VEHICLE
Location	1
Attribute ID	COLOR
Start Character	0
Attribute Length	0

1. For decision SRC1 to DST1, the source region attribute value is YEL and the destination attribute value is YEL. The attributes match so the decision is not eliminated.
2. For Decision SRC2 to DST1, the source region attribute value is GRN and the destination attribute value is YEL. The attributes do not match so the decision is eliminated.

Block by Attribute Pseudo Code

```

For each decision
  Get Item at specified source region location
  If there are items in the source region
    Get item at tail destination region location
    If there are items in the destination region
      If the attribute values do not match
        Eliminate the decision
      End if
    Else
      Eliminate the decision because there are no
      Items in the destination region
    End if
  Else
    Eliminate the decision because there are no items
    In the source location
  End if
Next
    
```

 **Note:** If no items of the specified class are contained in the source region or in the destination region, then the decision is eliminated.

Block by Extended Attribute

Description

Block by Extended Attribute eliminates each decision where the item at the specified source region location does not have an extended attribute that matches the item at the destination region tail location.

Parameters

This function block has the following parameters:

Parameter		Description
Always Evaluate		Hard, Soft or Breakable. This parameter controls what happens when the extension eliminates all available decisions.
	Hard	All decisions will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.
	Soft	Roll back any decisions eliminated by this extension.

	Breakable	Roll back any decisions eliminated by this extension that have a danger point value of True (non zero). The routing modules will first use the danger defined for each individual decision; and, if no danger point has been configured for the decision, then the site-wide danger point will be used. If no danger point can be found for a decision, then the decision remains eliminated.
Item Class	All items with matching class will be used.	
Location	Number of locations in source regions to scan.	
Attribute ID	Name of extended attribute to compare.	
Start Character	Starting character of the extended attribute value to be used.	
Attribute Length	Specifies the number of characters in the extended attribute value to compare.	
End Character	Ending character of the extended attribute value to be used. Enter zero in both the Start Character and Attribute Length fields to use all characters.	

Check Item Hold Status

Check Item Hold Status

Description

Check Item Hold Status operates differently depending on the number of destination regions.

If there is only one destination region, **Check Item Hold Status** eliminates any decisions where the items at the source region specified locations are on Internal or External Hold.

If there are 2+ destination regions, **Check Item Hold Status** eliminates any decision where the source item is on hold and the destination region is on hold or where the source item is on hold and the destination region is not on hold.

Parameters

This function block has the following parameters:

Parameter	Description	
Always Evaluate	Hard, Soft or Breakable. This parameter controls what happens when the extension eliminates all available decisions.	
	Hard	All decisions will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, None , is executed to handle this condition.
	Soft	Roll back any decisions eliminated by this extension.

	Breakable	Roll back any decisions eliminated by this extension that have a danger point value of TRUE (non zero). The routing modules will first use the danger defined for each individual decision; and, if no danger point has been configured for the decision, then the site-wide danger point will be used. If no danger point can be found for a decision, then the decision remains eliminated.
Item Class	All items with matching class will be used.	

Check Item Hold Status Example

Merge Example

Parameters

This function block has the following parameters:

Parameter Name	Parameter Value
Always Evaluate	HARD
Item Class	VEHICLE

1. For decision SRC1 to DST1, the item is on External hold, so the decision is eliminated.
2. For decision SRC2 to DST2, the item is not on External or Internal Hold, so the decision is not eliminated.

Split Example

3. For decision SRC1 to DST1, the item is not on External or Internal Hold but the destination region is on hold, so the decision is eliminated.
4. For decision SRC1 to DST2, the item is not on External or Internal hold and the destination region is also not on hold, so the decision is **not** eliminated.

Blank Example

Since all source regions have no items matching the specified item class. All decisions are eliminated.

Check Item Hold Status Pseudo Code


```

If single destination region
  For each decision
    Get items at specified location/item class from decision source region
    If there are items at the region and location
      For each item at location
        If current item is on internal or external hold
  
```

```

        Eliminate the decision
      Next decision
    End if
  Next for
End if
Next decision
Else
  // multiple destination regions
  For each decision
    Get items from specified location/item class from decision source
    region
    If there are items at the region and location
      For each item at location
        If current item is on internal or external hold
          Set source hold flag
        End if
      Next
      Get the region status point from the region's "Region Maintenance
      record"
      If the point value indicates the region is on hold
        Set the region hold flag
      End if
      If (the source hold flag is NOT set and the region hold flag is set) or
        (the source hold flag is set and region hold flag is not set)
        Eliminate the decision
      End If
    Next decision
  If there were no items found in any of the source regions
    Eliminate all decisions
  End if

```

 **Note:** In single or multiple destination mode, if no items matching the specified item class were found in the specified location of any decision source regions, then all decisions are eliminated.

Check Process Capability

Check Process Capability

Description

Check Process Capability eliminates decisions based on the comparison of the destination region's attributes and the region's Capability Point Value.

Parameters

This function block has the following parameters:

Parameter	Description
Always Evaluate	Hard, Soft or Breakable. This parameter controls what happens when the extension eliminates all available decisions.

	Hard	All decisions will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.
	Soft	Roll back any decisions eliminated by this extension.
	Breakable	Roll back any decisions eliminated by this extension that have a danger point value of TRUE (non zero). The routing modules will first use the danger defined for each individual decision; and, if no danger point has been configured for the decision, then the site-wide danger point will be used. If no danger point can be found for a decision, then the decision remains eliminated.
Item Class	All items with matching class will be used.	
Attribute ID	Name of Attribute to compare.	
Start Character	Starting character of the attribute to be used.	
Attribute Length	Specifies the number of characters in the attribute to compare.	
End Character	Ending character of the attribute to be used. Enter zero in both the Start Character and Attribute Length fields to use all characters.	
Capability Point Number	The capability point to be used. There are three potential capability points, only one may be specified.	
Check	Select either the decision's destination or source region.	

Check Process Capability Example

Parameters

This function block has the following parameters:

Parameter Name	Parameter Value
Always Evaluate Flag	HARD
Item Class	VEHICLE
Location	1
Attribute ID	COLOR
Start Character	0
Attribute Length	0
Capability Point Number	2
Decision Region	

Capability Point 2 Values

YEL

RED

BLUE

1. For decision SRC1 to DST1, the color attribute has a value of "YEL," and "YEL" is found in the first element of the capability point, so the decision is not eliminated.
2. For decision SRC2 to DST1, the color attribute has a value of "GRN," but "GRN" is not found in the capability point, so the decision is eliminated.

Check Process Capability Pseudo Code

```

For each decision
  Get items of specified class at specified location
  If items were found
    Get capability point for destination region
    If point is an array
      For each element of the array
        If the attribute value and the current array element value match
          Next decision
        End if
      Next element
    If no matches found between point values and attribute value
      Eliminate the decision
    End if
  Else
    If the point value and the attribute do NOT match
      Eliminate the decision
    End if
  End if
Else
  \\ no items found at specified location
  Eliminate the decision
End if
Next decision

```

guide: Guidelines

- Comparison is case insensitive.
- The Capability point must be of type string.
- If the Capability point is an array of strings, then each element will be scanned for the attribute.
- If no items of the specified class are found at the source region, all decisions are eliminated.

Check Process Capability (Ext. Attr.)


Description

Check Process Capability (Ext. Attr.) eliminates decisions based on the comparison of the source item's standard or extended attribute and the region's Capability Point Value.

Parameters

This function block has the following parameters:

Parameter	Description																
Always Evaluate	Hard, Soft or Breakable. Controls what happens when the extension eliminates all items in the range source.																
	<table border="1"> <tr> <td>Hard</td> <td>All items will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, None, is executed to handle this condition.</td> </tr> <tr> <td>Soft</td> <td>Roll back any eliminations made by this extension.</td> </tr> <tr> <td>Breakable</td> <td>Roll back any eliminations made by this extension that have a danger point value of TRUE (non zero). Routing modules will first use the danger defined for each individual decision. However, one of the following may occur instead.</td> </tr> <tr> <td></td> <td> <table border="1"> <tr> <td>If:</td> <td>Then:</td> </tr> <tr> <td> <ul style="list-style-type: none"> No danger point has been configured for the decision. </td> <td>The site-wide danger point will be used.</td> </tr> <tr> <td> <ul style="list-style-type: none"> This function block is used in the decision output logic. </td> <td>The site-wide danger point will be used.</td> </tr> <tr> <td> <ul style="list-style-type: none"> No danger point can be found for a decision. </td> <td>The decision remains eliminated.</td> </tr> </table> </td> </tr> </table>	Hard	All items will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, None , is executed to handle this condition.	Soft	Roll back any eliminations made by this extension.	Breakable	Roll back any eliminations made by this extension that have a danger point value of TRUE (non zero). Routing modules will first use the danger defined for each individual decision. However, one of the following may occur instead.		<table border="1"> <tr> <td>If:</td> <td>Then:</td> </tr> <tr> <td> <ul style="list-style-type: none"> No danger point has been configured for the decision. </td> <td>The site-wide danger point will be used.</td> </tr> <tr> <td> <ul style="list-style-type: none"> This function block is used in the decision output logic. </td> <td>The site-wide danger point will be used.</td> </tr> <tr> <td> <ul style="list-style-type: none"> No danger point can be found for a decision. </td> <td>The decision remains eliminated.</td> </tr> </table>	If:	Then:	<ul style="list-style-type: none"> No danger point has been configured for the decision. 	The site-wide danger point will be used.	<ul style="list-style-type: none"> This function block is used in the decision output logic. 	The site-wide danger point will be used.	<ul style="list-style-type: none"> No danger point can be found for a decision. 	The decision remains eliminated.
Hard	All items will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, None , is executed to handle this condition.																
Soft	Roll back any eliminations made by this extension.																
Breakable	Roll back any eliminations made by this extension that have a danger point value of TRUE (non zero). Routing modules will first use the danger defined for each individual decision. However, one of the following may occur instead.																
	<table border="1"> <tr> <td>If:</td> <td>Then:</td> </tr> <tr> <td> <ul style="list-style-type: none"> No danger point has been configured for the decision. </td> <td>The site-wide danger point will be used.</td> </tr> <tr> <td> <ul style="list-style-type: none"> This function block is used in the decision output logic. </td> <td>The site-wide danger point will be used.</td> </tr> <tr> <td> <ul style="list-style-type: none"> No danger point can be found for a decision. </td> <td>The decision remains eliminated.</td> </tr> </table>	If:	Then:	<ul style="list-style-type: none"> No danger point has been configured for the decision. 	The site-wide danger point will be used.	<ul style="list-style-type: none"> This function block is used in the decision output logic. 	The site-wide danger point will be used.	<ul style="list-style-type: none"> No danger point can be found for a decision. 	The decision remains eliminated.								
If:	Then:																
<ul style="list-style-type: none"> No danger point has been configured for the decision. 	The site-wide danger point will be used.																
<ul style="list-style-type: none"> This function block is used in the decision output logic. 	The site-wide danger point will be used.																
<ul style="list-style-type: none"> No danger point can be found for a decision. 	The decision remains eliminated.																
Item Class	Items of this class will be compared.																
Attribute Name	Name of standard or extended attribute to compare.																
Attribute Start Byte	Starting byte of the attribute to be used in the comparison.																
Attribute Length	Number of characters in the attribute to compare.																
Capability Point Number	The capability point to be used. There are three potential capability points, only one may be specified.																
Check	Select either the decision's destination or source region.																

 **Note:** All the function blocks that deal with extended attributes can be provided `INTERNALHOLD` text to read the value of an internal hold.

Values are:

0	Internal Hold is not set.
1	Internal Hold is set.

However, you cannot use any of the extended attribute function blocks to modify system attributes.

Check Region Full

Check Region Full

Description

Check Region Full eliminates each decision where routing an item to the destination **region** would cause the region to exceed the maximum number of items configured in the Region Maintenance Table.

Parameters

This function block has the following parameter:

Parameter	Description	
Always Evaluate	Hard, Soft or Breakable. This parameter controls what happens when the extension eliminates all available decisions.	
	Hard	All decisions will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.
	Soft	Roll back any decisions eliminated by this extension.
	Breakable	Roll back any decisions eliminated by this extension that have a danger point value of TRUE (non zero). The routing modules will first use the danger defined for each individual decision; and, if no danger point has been configured for the decision, then the site-wide danger point will be used. If no danger point can be found for a decision, then the decision remains eliminated.

Check Region Full Example

1. Decision SRC1 to DST2, destination region has a quantity of 2. Routing an item to this region would not exceed the maximum. This decision is not eliminated.
2. Decision SRC1 to DST2, the destination region has a quantity of 3. Routing to this location would cause the item count to exceed the maximum count configured in Region Maintenance. This decision is eliminated.

Check Region Full Pseudo Code

```

For each decisions
  If the decision is still possible
    If the decision source region has a Quantity point in Region
      Maintenance
        Get the value of the quantity point
        Get the region capacity
        If the current region capacity + 1 > the configured capacity
          Eliminate the decision
        End if
      End if
    End if
  End if
Next decision

```

Check Region Ready

Check Region Ready

Description

Check Region Ready eliminates a decision based on the value of the region status point or the region enable point for both the source and destination region. If the binary 'and' of the value of the region status point and the region Ready Mask for either source or destination regions is 0, then the decision is eliminated. If the value of the source or destination region's enable point is zero, then the decision is eliminated. If any of the points are configured, but are unavailable, the decision is eliminated.

Parameters

This function block has the following parameter:

Parameter	Description	
Always Evaluate	Hard, Soft or Breakable. This parameter controls what happens when the extension eliminates all available decisions.	
	Hard	All decisions will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.
	Soft	Roll back any decisions eliminated by this extension.
	Breakable	Roll back any decisions eliminated by this extension that have a danger point value of TRUE (non zero). The routing modules will first use the danger defined for each individual decision; and, if no danger point has been configured for the decision, then the site-wide danger point will be used. If no danger point can be found for a decision, then the decision remains eliminated.

Check Region Ready Example

Region Maintenance Table

Region ID	Region Ready Mask	Region Status Point	Enable Point
SRC1	2	SRC1_STATUS	SRC1_ENABLE
SRC2	2	SRC2_STATUS	SRC2_ENABLE
DST1	2	DST1_STATUS	DST1_ENABLE

Values of various points.

Point ID	Value
SRC1_STATUS	2
SRC2_STATUS	2
DST1_STATUS	2
SRC1_ENABLE	0
SRC2_ENABLE	1
DST1_ENABLE	1

For decision SRC1 to DST1:

1. Value of Ready Mask for the source region is 2.
2. Value of region status point for the source region is 2.
3. Binary and of Ready Mask and region status point is 1. The source region is "Ready," so the decision is not eliminated.
4. Value of Ready Mask for the destination region is 2.
5. Value of region ready point for the destination region is 2.
6. Binary and of Ready Mask and status point is 1. The destination region is "Ready," so the decision is not eliminated.
7. The of the source region enable point is 0. The source region is disabled, eliminate the decision.

For decision SRC2 to DST1:

8. Value of Ready Mask for the source region is 2.
9. Value of region ready point for the source region is 2.

10. Binary and of Ready Mask and status point is 1. The source region is "Ready," so the decision is not eliminated.
11. Value of Ready Mask for the destination region is 2.
12. Value of region status point for the destination region is 2.
13. Binary and of Ready Mask and region status point is 1. The destination region is "Ready," so the decision is not eliminated.
14. The of the source region enable point is 1. The source region is enabled, so the decision is not eliminated.
15. The of the destination region enable point is 1. The destination region is enabled, so the decision is not eliminated.

Check Region Ready Pseudo Code

```


For each decision that is not yet eliminated
  Get region Ready Mask for decision destination region
  Get status point for decision destination region
  Set value_to_check to value of regions status point
  If region status point is unavailable than
    Eliminate the decision
  End if
  If binary 'and' of value_to_check and region Ready Mask is 0
    Eliminate the decision
  End if
  Get decision destination region "enable point," from region maintenance
  table
  Get value of destination region "enable point"
  If (enable point is unavailable)
    Eliminate the decision
  End if
If value of the enable point is 0
  Eliminate the decision
End if
Get decision source region's, Ready Mask from region maintenance table
Get decision source region's, status point from region maintenance table
Set value_to_check to value of source region, region status point
If point is unavailable
  Eliminate the decision
End if
If binary 'and' of value_to_check and region Ready Mask is 0
  Eliminate the decision
End if
Get decision source region, region enable point from region maintenance
table
Get value of source region, region enable point
If point is unavailable

```

```

Eliminate the decision
End if
If value of region enable point is 0
Eliminate the decision
End if

```

 **Note:** All points from the region maintenance table are added to the Routing Control Object's point cache at startup.

Consume a Trigger

Description

Unexpected Point Processing must be enabled for **Consume a Trigger** to work. The next trigger received for the site will be passed through unexpected point process and will be buffered. If a 2nd + trigger is received, then that trigger is unexpected and the site is disabled.

After the site completes the current cycle the buffer trigger will be processed.

Parameters

This function block has the following parameter:

Parameter	Description
Trigger Name	Name of trigger to consume or blank to consume next trigger.

Example

Not available for this release.

Pseudo Code

Not available for this release.

Return Value

None.

Eliminate by Weight

Eliminate by Weight

Description

Eliminate by Weight eliminates all decisions whose current weight is less than the specified value. This Eliminate by Weight function block, listed under Routing is different from the RSA [Eliminate by Weight \(page 478\)](#) function block listed under Range Source Architecture.

Parameters

This function block has the following parameter:

Parameter	Description
Threshold	All decisions whose weight is less than this value are eliminated.

Eliminate by Weight Example

Parameters

This function block has the following parameters:

Parameter Name	Parameter Value
Threshold	15

1. For decision SRC1 to DST1, the current weight is 10, the threshold is 15, so the decision is eliminated.
2. For decision SRC2 to DST2, the current weight is 5, the threshold is 15, so the decision is eliminated.

Eliminate by Weight Pseudo Code

```

For each decision
  If the decision weight is less than the specified threshold
    Eliminate the decision
  End If
Next decision

```

Eliminate Decision


Description

Eliminate Decision eliminates the specified decision.

Parameters

This function block has the following parameter:

Parameter	Description
Decision	Decision name to be eliminated. Select All to eliminate all decisions.

 **Note:** Comparison is case sensitive.

Eliminate Decision Based on Attribute Spacing

Eliminate Decision Based on Attribute Spacing

Description

Eliminate Decision Based on Attribute Spacing eliminates a decision using spacing by attribute counts.

Modifications

This function block has an additional parameter, an array point ID that lists the item class for each attribute. Only items that belong to the specified item class will be considered.

Once the count exceeds the rule, the count is reset.

Parameters

This function block has the following parameters:

Parameter	Description
Item Class	Name of array point that lists item class of the item that contains the attribute. Specify the class of the items to be considered or leave blank for all items.
Attribute Name	Name of the array point that lists the attribute names to be compared. Compare value(s) of the corresponding element in the attribute value array point to the attribute value of the head item for each of the source regions.
Attribute Value	Name of the array point that contains the Attribute value(s). Each element can contain one or multiple values with a comma separator. Do not include quotes. For example: element 0 = 1 ton,3/4 ton element 1 = K
Spacing Rule	The number of jobs that must be spaced between jobs with the corresponding attribute value(s).
Spacing Count	The number of jobs released since a job with the corresponding attribute value was released.
Spacing Breakability	If breakable (element = 1) and all sources are eliminated, roll back the decisions eliminated during the comparison by one set of related elements.

*Eliminate Decision Based on Attribute Spacing Example***Initial Point Values**

Offset	Item Class	Attribute Name	Attribute Value	Spacing Rule	Spacing Count	Spacing Breakability
0	VEHICLE	COLOR	YEL,GRN	2	2	0
1	VEHICLE	SIZE	S	3	0	1
2	VEHICLE	COLOR	BLACK	3	0	0

Site is triggered and logic module with a single function block, Eliminate Decision Based On Attribute Spacing, is executed.

Array Offset 0**SRC1.**

Head item with class of Vehicle, has a COLOR attribute with value of YEL. The possible attribute values are YEL or GRN. There is a match, so the spacing count is compared to the spacing rule. Since the spacing count is not less than spacing rule, the decision is not eliminated.

SRC2.

Head item with class of Vehicle, has a COLOR attribute with value of GRN. The possible attribute values are YEL or GRN. There is a match, so the spacing count is compared to the spacing rule. Since the spacing count is not less than the spacing rule, the decision is not eliminated.

There are still decisions possible so the breakability flag is not checked.

Array Offset 1**SRC1.**

Head item with class of Vehicle, has a SIZE attribute with value of S. The possible attribute value is S. There is a match, so the spacing count is compared to the spacing rule. Since the spacing count is less than the spacing rule, the decision is eliminated.

SRC2.

Head item with class of Vehicle, has a SIZE attribute with value of S. The possible attribute value is S. There is a match, so the spacing count is compared to the spacing rule. Since the spacing count is less than the spacing rule, the decision is eliminated.

There are no decisions possible, so the breakability flag is checked. The flag is set to 1.

Array Offset 2

SRC1.

Head item with class of Vehicle, has a COLOR attribute with value of YEL. The possible attribute value is BLACK. There is no match, so no further processing takes place.

SRC2.

Head item with class of Vehicle, has a COLOR attribute with value of GRN. The possible attribute value is BLACK. There is no match, so no further processing takes place.

Eliminate Decision Based on Attribute Spacing Pseudo Code

```

For each array element
  Get attribute name
  Get attribute value
  Get breakable flag
  If attribute name and value are not blank
    For each still possible decision
      Get the items at the decisions head location
      If (number of items at head > 0)
        For each item in head location
          Get the attribute value
          If the attribute was found
            If the attribute values match
              If the spacing count is greater than spacing rule
                Eliminate the decision
                If the breakable flag is not set
                  If (all decisions have been eliminated)
                    Alarm
                  End if
                End if
              End if
            End if
          End if
        Next item at head location
      Next valid decision
      If all decisions have been eliminated and breakable flag is set
        Rollback any decisions eliminated by this array offset
      End if
    Next element
  
```

*Eliminate Decision Based on Ext. Attribute Spacing***Description**

Eliminate Decision Based on Ext. Attribute Spacing eliminates a decision using spacing by attribute counts for extended attributes.

This function block has an Item Class Array point ID parameter that lists the item class for each attribute. Only items that belong to the specified item class will be considered for the corresponding rule.

Once the count exceeds the rule, the count is reset.

Parameters

This function block has the following parameters:

Parameter	Description
Item Class Array	Name of array point that lists item class of the item that contains the attribute. Specify the class of the items to be considered or leave blank for all items.
Attribute Name Array	Name of the array point that stores the names of the extended attributes to be compared. Note: The values of corresponding element in the Attribute Value array is compared with the value for that attribute (if found) of the specified item, i.e. the value of the corresponding element in the Attribute Value Array point is compared to the extended attribute value of the head item of the source region(s).
Attribute Value Array	Name of the array point that contains the Attribute value(s). Each element can contain one or multiple values with a comma separator. Do not include quotes. For example: element 0 = 1 ton,3/4 ton element 1 = K
Spacing Rule Array	Name of the array point that stores the number of jobs with different extended attribute values that must be processed between two jobs having the corresponding attribute value(s).
Spacing Count Array	The number of jobs released since a job with the corresponding extended attribute value was released.
Breakable Array	Name of the array point that stores a Breakable Flag element for each rule. If breakable (element = 1) and all sources are eliminated, roll back the decisions eliminated during the comparison by one set of related elements.

Eliminate Decision Based on Rule

Eliminate Decision Based on Rule

1. The function block attempts to match the respective Attribute Value element with a value for that attribute from the current item.
2. For a matching Attribute Value, the function block compares
 - the respective Count and Rule elements, or
 - the respective release limits and release counts
 according to the respective Rule Type element (**G**, **S**, or **P**).
3. If selecting the current item (incrementing the Count) would violate the rule, the comparison fails for that attribute and the item is either weighted or eliminated. The failure is saved as history data.

If selecting the current item would not violate this rule, the function block attempts to match the next attribute name/value match.**Notes**

- An elimination may be rolled back depending on the respective danger flag setting in the Breakable array.

Items remain in the range source unless they are eliminated.

- To compare items that satisfy a named solve (query) or query expression, enter the reserved name **SOLVE** for this element in the Attribute Name array. (This is instead of the name of an attribute). You can then enter a named query or query expression as the respective element in the Attribute Value array.

4. The number of red cars selected has not yet reached the Upper Release % Limit defined for Color/Red.
5. Selecting the red two-door would cause the percentage of four door cars selected to fall below its defined Lower Release % Limit
6. The red, two-door car is eliminated instead of being selected.

Eliminate Decision Based on Rule Example

Initial Point Values

Offset	Rule Type	Item Class	Attribute Name	Attribute Value	Rule	Count	Breakability
0	G	VEHICLE	COLOR	YEL,GRN	2	2	0
1	S	VEHICLE	SIZE	S	3	0	1
2	P	VEHICLE	COLOR	GRN	3	0	0

- The site is triggered
- A Logic module with a single function block, Eliminate Decision Based On Rules is executed.

Array Loc1

The rule defined in location one is a grouping rule.

SRC1.

Head item with class of Vehicle, has a COLOR attribute with value of YEL. The possible attribute values are YEL or GRN. There is a match, so the Count is compared to the Rule. Since the Count is greater then or equal to the Rule, the decision is eliminated.

SRC2.

Head item with class of Vehicle, has a COLOR attribute with value of GRN. The possible attribute values are YEL or GRN. There is a match, so the Count is compared to the Rule. Since the Count is greater then or equal to the Rule, the decision is eliminated.

Array Offset 1

The Rule defined in offset one is a Spacing rule.

SRC1.

Head item with class of Vehicle, has a SIZE attribute with value of S. The possible attribute value is S. There is a match, so the Count is compared to the Rule. Since the Count is less than the Rule, the decision is eliminated.

SRC2.

Head item with class of Vehicle, has a SIZE attribute with value of S. The possible attribute value is 'S'. There is a match, so the count is compared to the rule. Since the count is less than the rule, the decision is eliminated.

There are no decisions possible, so the breakability flag is checked. The flag is set to 1, we will roll back any decisions eliminated in this pass.

Array Offset 2

Initial Point Values

Offset	Rule Type	Item Class	Attribute Name	Attribute Value	Upper Rel %	Lower Rel %	Attrib Count	Total Count	Breakability
0	G	VEHICLE	COLOR	YEL,GRN	0	0	0	0	0
1	S	VEHICLE	SIZE	S	0	0	0	0	1
2	P	VEHICLE	COLOR	GRN	75	50	6	10	0

The Rule defined in offset one is a Percentage rule.

SRC1.

Head item with class of Vehicle, has a COLOR attribute with value of YEL. The possible attribute value is GRN. We don't have a match, so no further processing takes place.

SRC2.

Head item with class of Vehicle, has a COLOR attribute with value of GRN. The possible attribute value is GRN. The total of items released with this attribute is 6, with a total release count of 10, so 60% of the items have had a color attribute of GRN. Since 60% is less than 75 and greater than 50, the decision is not eliminated.

- The result is a **Wait** decision (no decision passes all filters) since at least the Percentage rule is not being processed correctly.

Eliminate Decision Based on Rule Pseudo Code

```

    For each array element
    { \\ start for each element in array element
    Get attribute name, Get attribute value, Get breakable flag , Get Item
    Class
    If attribute name and value are not blank
    { \\ start if attribute name and value are not blank.
    For each still possible decision
    { \\ start for each decision whose eliminate flag is not set
    Get the items at the decisions head location
    If (number of items at head > 0)
    { \\ start if items > 0
    For each item in head location
    { \\ start for each item in location
    Get the attribute value
    If (the attribute was found and the attribute values
    match)
    { \\ start if attribute found and value matched)
    Set eliminate flag to false
    if Rule type[P3] == Group
    if Group count[P7] is >= rule [P6]
    set eliminate flag to true
    end if
    Else if Rule type[P3] == Percentage
    Calculate the percentage released =
    Attribute Count[P10] / Total count [P11]* 100
    If (Percent Release is >= Upper Rel %[P8] or
    Percent Release is <= Lower Rel[P9] %)
    Set the eliminate flag.
    End if
    Else \\ if rule type is spacing
    If (spacing count[P7] < rule [P6])
    Set eliminate flag to true
    End if
    }\\endif attribute found and match
    }\\end for next item at location
    }\\end if item > 0
    }\\end for each possible decision
    if(breakability flag [P12]is set)
    { \\ start if breakability flag set
    if (all decisions are eliminated)
    {
    Roll back decisions eliminated by this pass of array element (rule
    type)
    ( i.e. do not eliminate decisions whose eliminate flag is set on
    this pass)
    }
    else \\ if all decisions are not eliminated and breakability flag is set
    {
    Eliminate decisions whose eliminate flag is set in this pass
    }
    }

```



```

}\\ end if breakability flag set
else \\ if breakability flag is not set for this pass
{
    Eliminate decisions whose eliminate flag is set in this pass
}
} //endiif attribute value and name are blank
} //end for - next array element

```

Eliminate Decision Based on Rule (Ext. Attr.)

Description

Eliminate Decision Based on Rule (Ext. Attr.) eliminates decisions based on standard or extended attribute Grouping and Spacing Rules. This block can be configured to eliminate by either type of rule and treats both rules with the same priority.


- Rules are as follows.
- Spacing rule uses the same logic as the function block, Eliminate Decision Based On Ext. Attribute Spacing. The spacing rule breaks up items with matching extended attributes. For example, you do not want two items with a color attribute of BLUE to be routed one after another. The Spacing rule would eliminate the decision that would send the second Blue item.
- Grouping rule breaks up blocks of items with a specified extended attribute. For example you do not want to send the color combination "Blue,Red,Blue," more than twice. The Grouping rule would tend to eliminate decisions that would result in this block.
- Percentage rule, eliminates a decision if the percentage of extended attributes released with a particular name/value combination falls outside of a specified range.
- The Output block, [Update Counts \(page 441\)](#), is required for recording the results of the decision. The count arrays will never be updated without the Update Counts block.

Parameters

This function block has the following parameters:

Parameter	Description	
Always Evaluate	Hard, Soft or Breakable. Controls what happens when the extension eliminates all items in the range source.	
	Hard	All items will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.
	Soft	Roll back any eliminations made by this extension.
	Breakable	Roll back any eliminations made by this extension that have a danger point value of TRUE (non zero). Routing modules will first use the danger defined for each individual decision. However, one of the following may occur instead.
	If:	Then:

		<ul style="list-style-type: none"> No danger point has been configured for the decision. 	The site-wide danger point will be used.
		<ul style="list-style-type: none"> This function block is used in the decision output logic. 	The site-wide danger point will be used.
		<ul style="list-style-type: none"> No danger point can be found for a decision. 	The decision remains eliminated.
Item Class Array	Name of the array point that stores the name of the item class to be checked for the corresponding extended attribute. Blank will specify all items at the region.		
Rule Type Array	Type of elimination to be performed.		
	G	Grouping	Eliminate if the count exceeds or equals the Rule.
	S	Spacing	Eliminate if the count is less then the rule.
	P	Percentage	
Attribute Name Array	Name of the array point that stores the names of the standard and/or extended attributes to be compared. Note: The values of corresponding element in the Attribute Value array is compared with the value for that attribute (if found) of the specified item, i.e. the value of the corresponding element in the Attribute Value Array point is compared to the extended attribute value of the head item of the source region(s).		
Attribute Value Array	Name of array points that lists the Attribute value(s). Each element can contain one or multiple values with a comma separator. Do not include quotes. Example Element 0 = 1 ton,3/4 ton Element 1 = K		
Rule Array	(Grouping and Spacing Rules) Name of the array that stores the number of jobs for each rule.		
Count Array	(Grouping and Spacing Rules) Name of the array that stores the number of jobs released.		
Upper Release % Target	(Percentage Rule Only) The upper percentage limit, of an acceptable range, for items that should be released with the corresponding attribute value.		
Lower Release % Target	(Percentage Rule Only) The lower percentage limit, of an acceptable range, for items that should be released with the corresponding attribute value.		
Attribute Release Count Array	(Percentage Rule Only) Name of the array that stores the actual number of items released with the corresponding attribute value.		
Total Released Count	(Percentage Rule) Integer (analog) point. Counter point stores and continues to increment the total number of items released.		
Spacing Breakability	Name of the array point. If breakable (element = 1) and all sources are eliminated, roll back the decisions eliminated during the comparison by one set of related elements.		

 **Note:** All the function blocks that deal with extended attributes can be provided `INTERNALHOLD` text to read the value of an internal hold.

Values are:

0	Internal Hold is not set.
1	Internal Hold is set.

However, you cannot use any of the extended attribute function blocks to modify system attributes.

Eliminate Decision by Attribute

Eliminate Decision by Attribute is obsolete. Use the current function blocks:

- Eliminate Decision by Attribute attribute.
- Eliminate Decision by Attribute Point.
- **Eliminate Decision by Attribute Value.**

Eliminate Decision by Extended Attribute

Description

`Eliminate Decision by Extended Attribute` eliminates a named decision based on the comparison of an extended attribute of an item and a specified extended attribute, value, or point.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	The region where the item is.
Region Location	Location, within the decision source region, of the item attribute.
Item Class	Class of the item.
Attribute	Name of the extended attribute to compare.
Attribute Start Byte	Starting character within extended attribute.
Attribute Length	Specifies the number of characters in attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.
Decision to Eliminate	Name of decision to eliminate.
Condition	Comparison operator, Match, Mismatch, Greater or Lessor.

Comparison	Type of compare, either numeric comparison or alphabetic comparison.
Compare Against	Specifies where to get a comparison value: Extended Attribute, Value or Point.
Value to Compare	Specifies a value to compare against. Used if Compare Against is Value.
Point Name	ID of point to use in a comparison. Used if Compare Against is Point.
Region ID	The region where the item with an attribute to compare is. Used if Compare Against is Attribute.
Region Location	Location holding the item with an attribute to compare. Used if Compare Against is Attribute.
Item Class	Class of the item to compare. Used if Compare Against is Attribute.
Attribute Id	Name of the extended attribute to compare. Used if Compare Against is Attribute.
Attribute Start Byte	Starting character within extended attribute. Used if Compare Against is Attribute.
Attribute Length	Specifies the number of characters in attribute to compare. Used if Compare Against is Attribute. Enter zero in both the Start Character and Attribute Length fields to use all characters.

Eliminate Decision by Attribute Attribute

Eliminate Decision by Attribute Attribute

Description

Eliminate Decision by Attribute Attribute eliminates a decision based on the comparison of two attribute values.

Parameters

This function block has the following parameters:

Parameter	Description
Decision ID	ID of decision to be eliminated.
Region ID	Region which contains the first item to compare.
Region Location	Location of first item in the region.
Item Class	Class of item to compare. Optional
Attribute ID	Attribute of first item to compare.
Start Character	Starting character of the attribute to be used.
Attribute Length	Specifies the number of characters in the attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.

Condition	Type of Comparison - Greater Than, Less Than, Equals.
Comparison	Specifies the comparison type – numeric, alphabetic.
Region ID	Region which contains the second item to compare.
Region Location	Location of second item in the region.
Item Class	Class of the second item.
Attribute ID	Attribute of second item to compare.
Start Character	Starting character of the attribute to be used.
Length	Number of characters to compare.

Eliminate Decision by Attribute Attribute Example

Site is triggered and a logic module with a single extension, Eliminate Decision By Attribute Attribute, is run. The parameters are as follows:

Parameters

This function block has the following parameters:

Parameter Name	Parameter Value
Decision ID	SRC1 to DST1
Region ID	SRC1
Region Location	1
Item Class	VEHICLE
Attribute ID	COLOR
Start Character	0
Attribute Length	0
Condition	Equals
Comparison Type	Alphanumeric
Region ID	DST1
Region Location	PRT_LAST
Item Class	
Attribute ID	COLOR
Start Character	4
Length	3

1. The value of the Color attribute of the item in the head location of region SRC1 is YEL.
2. The value of characters 4 to 7 of the Color attribute of the item in the last location of region DST1 is **YEL**.
3. The attributes match so decisions SRC1 to DST1 is eliminated.

Eliminate Decision by Attribute Attribute Pseudo Code

Comparison is case sensitive.

```

Get the First Attribute value
Get the second attribute value
Switch comparison type
  Case Greater than
  If (attribute1 is > attribute2)
    Eliminate the decision
  End if
  Case Less Than:
  If (attribute 1 is < attribute 2)
    Eliminate the decision
  End if
  Case Equal To
  If (attribute 1 = = attribute 2)
    Eliminate the decision
  End if
End switch

```

Eliminate Decision by Attribute Point

Eliminate Decision by Attribute Point

Description

Eliminate Decision by Attribute Point eliminates a decision based on the comparison of an attribute to a point value. If the comparison is true, the decision is eliminated.

Parameters

This function block has the following parameters:

Parameter	Description
Decision ID	ID of decision to be eliminated.
Region ID	Region which contains the item to compare.
Region Location	Location of item in the region.
Item Class	Class of item to compare.

Attribute ID	Attribute of first item to compare.
Start Character	Starting character of the attribute to be used.
Attribute Length	Specifies the number of characters in the attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.
Condition	Type of Comparison - Greater Than, Less Than, Equals.
Comparison Type	Numeric or alphanumeric comparison. If numeric comparison is selected, values are converted to integers before comparison.
Point ID	ID of point whose value is to be compared.

Eliminate Decision by Attribute Point Example

Site is triggered and a logic module with a single extension, Eliminate Decision by Attribute Point, is run. The parameters are as follows:

Parameters

This function block has the following parameters:

Parameter Name	Parameter Value
Decisions ID	SRC1 to DST1
Region ID	SRC1
Region Location	1
Item Class	VEHICLE
Attribute ID	COLOR
Start Character	0
Attribute Length	0
Comparison Operator	Equals
Comparison Type	Numeric
Point ID	COMPARE_POINT

1. The value of the Color attribute of the item in the head location of region SRC1 is **YEL**.
2. The value of the point is **YEL**.
3. The attribute and the point value match, so decisions SRC1 to DST1 is eliminated.


Eliminate Decision By Attribute Point Pseudo Code

```
Get the First Attribute value
Get the Point Value
Switch comparison operator
```

```

Case Greater than
If (attribute1 is > Point Value)
    Eliminate the decision
End if
Case Less Than:
    If (attribute 1 is < Point Value)
        Eliminate the decision
    End if
Case Equal To
    If (attribute 1 = = Point Value)
        Eliminate the decision
    End if
End switch

```

 **Note:** Alphanumeric comparison is case sensitive.

Eliminate Decision by Attribute Value

Eliminate Decision by Attribute Value

Description

Eliminate Decision by Attribute Value eliminates a decision based on the comparison of an attribute value to a RCO variable or hard coded value.

Parameters

This function block has the following parameters:

Parameter	Description
Decision ID	ID of decision to be eliminated.
Region ID	Region which contains the item to compare.
Region Location	Location of item in the region.
Item Class	Class of item to compare.
Attribute ID	Attribute of first item to compare.
Start Character	Starting character of the attribute to be used.
Attribute Length	Specifies the number of characters in the attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.
Condition	Type of Comparison - Greater Than, Less Than, Equals.
Comparison Type	Numeric or Alphanumeric comparison. If numeric comparison is selected, attribute and value are converted to integers before comparison.

Value	Select an RCO Variable name or type a value.
-------	--

Eliminate Decision by Attribute Value Example

Site is triggered and a logic module with a single extension, Eliminate Decision By Attribute Value, is run.

Parameters

This function block has the following parameters:

Parameter Name	Parameter Value
Decision ID	SRC1 to DST1
Region ID	SRC1
Region Location	1
Item Class	VEHICLE
Attribute ID	COLOR
Start Character	0
Attribute Length	0
Comparison Operator	Equals
Comparison Type	Alphanumeric
Value	YEL

1. The value of the Color attribute of the item in the head location of region SRC1 is **YEL**.
2. The value is **YEL**.
3. The attributes and the value match, decisions SRC1 to DST1 is eliminated.


Eliminate Decision by Attribute Value Pseudo Code

```

Get the Attribute
Switch comparison type
Case Greater than
  If (attribute1 is > Value)
    Eliminate the decision
  End if
Case Less Than:
  If (attribute 1 is < Value)
    Eliminate the decision
  End if
Case Equal To
  If (attribute 1 = = Value)
    Eliminate the decision

```

```
End if
End switch
```

 **Note:** Alphanumeric comparison is case sensitive.

Eliminate Decision by Ext. Attribute Attribute

Description

Eliminate Decision by Ext. Attribute Attribute eliminates a decision based on the comparison of two attribute values.

Parameters

This function block has the following parameters:

Parameter	Description
Decision ID	ID of decision to be eliminated.
Region ID	Region which contains the first item to compare.
Region Location	Location of first item in the region.
Item Class	Class of item to compare. Optional
Attribute ID	Attribute of first item to compare.
Start Character	Starting character of the attribute to be used.
Attribute Length	Specifies the number of characters in the attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.
Condition	Type of Comparison - Greater Than, Less Than, Equals.
Comparison	Specifies the comparison type – numeric, alphabetic.
Region ID	Region which contains the second item to compare.
Region Location	Location of second item in the region.
Item Class	Class of the second item.
Attribute ID	Attribute of second item to compare.
Start Character	Starting character of the attribute to be used.
Length	Number of characters to compare.

Eliminate Decision by Ext Attribute Point

Description

Eliminate Decision by Ext. Attribute Point eliminates a decision based on the comparison of an attribute to a point value. If the comparison is true, the decision is eliminated.

Parameters

This function block has the following parameters:

Parameter	Description
Decision ID	ID of decision to be eliminated.
Region ID	Region which contains the item to compare.
Region Location	Location of item in the region.
Item Class	Class of item to compare.
Attribute ID	Attribute of first item to compare.
Start Character	Starting character of the attribute to be used.
Attribute Length	Specifies the number of characters in the attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.
Condition	Type of Comparison - Greater Than, Less Than, Equals.
Comparison Type	Numeric or alphanumeric comparison. If numeric comparison is selected, values are converted to integers before comparison.
Point ID	ID of point whose value is to be compared.s

Eliminate Decision by Ext. Attribute Value

Description

Eliminate Decision by Ext. Attribute Value eliminates a decision based on the comparison of an extended attribute value to a RCO variable or hard coded value.

Parameters

This function block has the following parameters:

Parameter	Description
Decision ID	ID of decision to be eliminated.
Region ID	Region which contains the item to compare.
Region Location	Location of the item in the region.

Item Class	Class of the item to compare.
Attribute ID	Attribute of first item to compare.
Start Character	Starting character of the attribute to be used.
Attribute Length	Specifies the number of characters in the attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.
Condition	Type of Comparison - Greater Than, Less Than, Equals.
Comparison Type	Numeric or Alphanumeric comparison. If numeric comparison is selected, attribute and value are converted to integers before comparison.
Value	Select an RCO Variable name or type a value.

Eliminate Decision by Increasing Weight Percentages

Eliminate Decision by Increasing Weight Percentages

Description

Eliminate a Decision by Increasing Weight Percentages will adjust a decision's weight, and is usually run in conjunction with **the Select Decision with Highest Order & Weight** function block. The weight change only has effect during the current control cycle.

Modifications

This function block has an additional parameter, an array Point ID that lists the item class for each attribute. Only items that belong to the specified item class will be considered. Remainder of this function block is unchanged.

Parameters

This function block has the following parameters:

Parameter		Description
Always Evaluate		Hard, Soft or Breakable. This parameter controls what happens when the extension eliminates all available decisions.
	Hard	All decisions will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, None , is executed to handle this condition.
	Soft	Roll back any decisions eliminated by this extension.
	Breakable	Roll back any decisions eliminated by this extension that have a danger point value of TRUE (non zero). The routing modules will first use the danger defined for each individual decision and, if no danger point has been configured for the decision, then the site-wide danger point will be used. If no danger point can be found for a decision, then the decision remain eliminated.

Item Class	Name of array point that lists item class of the item that contains the attribute. Specify the class of the items to be considered or leave blank for all items.
Attribute Name	Name of the array point that lists the attribute names to be compared.
Attribute Value	Name of the array point that contains the Attribute Value. Each Element can contain one or multiple values separated by a comma. Do not include quotes. Example element 0 = Blue,Green element 1 = K
Upper Release % Target	The upper percentage limit, of an acceptable range, for items that should be released with the corresponding attribute value.
Lower Release % Target	The lower percentage limit, of an acceptable range, for items that should be released with the corresponding attribute value.
Attribute Released Count	The actual number of times released with the corresponding attribute value.
Total Released Count	Integer (analog) point. Counter point stores and continues to increment the total number of items released.

Eliminate Decision by Increasing Weight Percentages Example

Offset	Item Class	Attribute Name	Attribute Value	Upper %	Lower %	Attribute Count	Total Count
0	VEHICLE	COLOR	YEL,GRN	75	25	10	100
1	VEHICLE	SIZE	M	50	10	90	1

Array Offset 0

SRC1.

Head Item with class of VEHICLE has a color attribute of YEL. Possible attribute values are YEL and GRN. We have a match so the Attribute Percentage has the following formula:

$$\text{Attribute Percent} = \text{Attribute Release Count} / \text{total Count} * 100$$

$$10 = 10/100 * 100$$

Since Attribute Percent is lower than 10 %, the decision weight is increased as follows:

$$\text{Weight Delta} = \text{Upper \%} - \text{Attrib \%}$$

$$65 = 75 - 10$$

So the weight will now be 75 for the decision.

SRC2.

Head Item with class of VEHICLE has a color attribute of GRN. Possible attribute values are YEL and GRN. We have a match so the Attribute Percentage has the following formula:

$$\text{Attribute Percent} = \text{Attribute Release Count} / \text{Total Count} * 100$$

$$10 = 10/100 * 100$$

Since Attribute Percent is lower than 10 %, the decision weight is increased as follows:

$$\text{Weight Delta} = \text{Upper \%} - \text{Attrib \%}$$

$$65 = 75 - 10$$

So the weight will now be 70 for the decision.

Array Offset 1

SRC1.

Head Item with class of VEHICLE has a size attribute of M. Possible attribute values are S M L XL. We have a match so the Attribute Percentage has the following formula:


$$70 = 70/100 * 100$$

Since this value is greater than the Upper %, the decision is eliminated.

SRC2.

Head Item with class of VEHICLE has a size attribute of S. There is no match so the decision is skipped.

Thus SRC2 is the only decision remaining, so it will be selected.

 **Note:** In the event there are multiple decisions remaining that have had their weights increased, the Select Decision with Highest Order and Weight Function Block can be run to select the appropriate decision.

Eliminate Decision by Increasing Weight Percentages Pseudo Code

```

For each attribute name
  For each decision
    For each Item in head location of source region
      If the attribute is found in the Item
        Attrib Percent = 0
        If Total Released Count !=0
          Attribute Percent = Attribute Release Count / Total
          Released Count * 100
        If Attrib Percent > Upper Release Target %
          Elminate Decision
  
```

```

Alarm if no more decisions else continue
If Attrib Percent < Lower Release Target %
SetWeight (current decision weight + Upper Release Target
% - Attrib Percent)

```

Eliminate Decision by Increasing Weight Percentages (Ext. Attr.)

Description

Eliminate a Decision by Increasing Weight Percentages (Ext. Attr.) will adjust a decision's weight, and is usually run in conjunction with **the Select Decision with Highest Order & Weight** function block. The weight change only has effect during the current control cycle.

Modifications


This function block has an additional parameter, an array Point ID that lists the item class for each standard or extended attribute. Only items that belong to the specified item class will be considered. Remainder of this function block is unchanged.

Parameters

This function block has the following parameters:

Parameter	Description		
Always Evaluate	Hard, Soft or Breakable. Controls what happens when the extension eliminates all items in the range source.		
	Hard	All items will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.	
	Soft	Roll back any eliminations made by this extension.	
	Breakable	Roll back any eliminations made by this extension that have a danger point value of TRUE (non zero). Routing modules will first use the danger defined for each individual decision. However, one of the following may occur instead.	
		If:	Then:
		<ul style="list-style-type: none"> No danger point has been configured for the decision. 	The site-wide danger point will be used.
		<ul style="list-style-type: none"> This function block is used in the decision output logic. 	The site-wide danger point will be used.
		<ul style="list-style-type: none"> No danger point can be found for a decision. 	The decision remains eliminated.

Item Class Array	Name of array point that lists item class of the item that contains the extended attribute. Specify the class of the items to be considered or leave blank for all items.
Attribute Name Array	Name of the array point that lists the attribute names to be compared.
Attribute Value Array	Name of the array point that contains the Attribute Value. Each Element can contain one or multiple values separated by a comma. Do not include quotes. Example element 0 = Blue,Green element 1 = K
Upper Release % Target Array	The upper percentage limit, of an acceptable range, for items that should be released with the corresponding standard or extended attribute value.
Lower Release % Target Array	The lower percentage limit, of an acceptable range, for items that should be released with the corresponding standard or extended attribute value.
Attribute Released Array	The actual count of items released with the corresponding standard or extended attribute value.
Total Released Count	Integer (analog) point. Counter point stores and continues to increment the total number of items released.

 **Note:** All the function blocks that deal with extended attributes can be provided `INTERNALHOLD` text to read the value of an internal hold.

Values are:

0	Internal Hold is not set.
1	Internal Hold is set.

However, you cannot use any of the extended attribute function blocks to modify system attributes.

Eliminate Decision by Point

Eliminate Decision by Point is obsolete. Use the current function blocks:

Eliminate Decision by Point Point
Eliminate Decision by Point Value.
Routing function blocks list.

Eliminate Decision by Point (Ext. Attr.)


Description

Eliminate Decision by Point (Ext. Attr.) eliminates a decision based on the comparison of a standard or extended attribute and a point value.

Parameters

This function block has the following parameters:

Parameter	Description
Point Name	ID of point to use in a comparison.
Decision to Eliminate	Name of decision to eliminate.
Comparison Type	Comparison operator, Match, Mismatch, Greater or Lessor.
Value Type	Type of compare, either numeric comparison or alphabetic comparison.
Compare Against	Specifies where to get a comparison value: Extended Attribute, Value or Point.
Value to Compare	Specifies a value to compare against. Used if Compare Against is Value.
Point Name	ID of point to use in a comparison. Used if Compare Against is Point.
Region ID	The region where the item with an attribute to compare is. Used if Compare Against is Attribute.
Region Location	Location holding the item with an attribute to compare. Used if Compare Against is Attribute.
Item Class	Class of the item to compare. Used if Compare Against is Attribute.
Attribute Id	Name of the standard or extended attribute to compare. Used if Compare Against is Attribute.
Attribute Start Byte	Starting character within standard or extended attribute. Used if Compare Against is Attribute.
Attribute Length	Specifies the number of characters in attribute to compare. Used if Compare Against is Attribute. Enter zero in both the Start Character and Attribute Length fields to use all characters.

 **Note:** All the function blocks that deal with extended attributes can be provided `INTERNALHOLD` text to read the value of an internal hold.

Values are:

0	Internal Hold is not set.
1	Internal Hold is set.

However, you cannot use any of the extended attribute function blocks to modify system attributes.

Eliminate Decision by Point PointEliminate Decision by Point Point**Description**

Eliminate Decision by Point Point eliminates a decision based on the comparison of two point values.

Parameters

This function block has the following parameters:

Parameter	Description
Decision ID	ID of the decision to be eliminated.
Point ID 1	ID of the point whose value is to be compared.
Comparison Operator	Type of Comparison - Greater Than, Less Than, Equals.
Point ID 2	ID of second point whose value is to be compared.
Comparison Type	Numeric or alphanumeric comparison. If numeric comparison is selected, values are converted to integers before comparison.

Eliminate Decision by Point Point Example

Site is triggered and a logic module with a single extension, Eliminate Decision by Point Point, is run.

Parameters

This function block has the following parameters:

Parameter Name	Parameter Value
Decision ID	SRC1 to DST1
Point ID 1	COMPARE_1
Comparison Operator	Equals
Point ID 2	COMPARE_2
Comparison Type	Alphanumeric

1. The value of point COMPARE_1 is **YEL**.
2. The value is point COMPARE_2 is YEL.


3. The point values match, decisions SRC1 to DST1 is eliminated.

Eliminate Decision by Point Point Pseudo Code

```

Get value of point 1
GetValue of Point 2
Switch comparison type
  Case Greater than
    If (Value 1 is > Value 2)
      Eliminate the decision
    End if
  Case Less Than:
    If (Value 1 is < Value 2)
      Eliminate the decision
    End if
  Case Equal To
    If (Value 1 = = Value 2)
      Eliminate the decision
    End if
End switch

```

 **Note:** Alphanumeric comparison is case sensitive.

Eliminate Decision by Point Value

Eliminate Decision by Point Value

Description

Eliminate Decision by Point Value eliminates a decision based on the comparison of a point to a value.

Parameters

This function block has the following parameters:

Parameter	Description
Decision ID	ID of the decision to be eliminated.
Point ID 1	ID of the point whose value is to be compared.
Comparison Operator	Type of Comparison - Greater Than, Less Than, Equals.
Value	Drop List combo listing the available RCO variables. Operator may select a RCO Variable from the list or type any value.
Comparison Type	Numeric or alphanumeric comparison. If numeric comparison is selected, values are converted to integers before comparison.

Eliminate Decision by Point Value Example

Site is triggered and a logic module with a single extension, Eliminate Decision by Point Value, is run.

Parameters

This function block has the following parameters:

Parameter Name	Parameter Value
Decision ID	SRC1 to DST1
Point ID 1	COMPARE_1
Comparison Operator	Equals
Value	GRN
Comparison Type	Alphanumeric


1. The value of point COMPARE_1 is **YEL**.
2. The point value does not match the value, so decision SRC1 to DST1 is not eliminated.

Eliminate Decision by Point Value Pseudo Code

```

Get value of point 1
Switch comparison type
  Case Greater than
    If (Point Value is > Value)
      Eliminate the decision
    End if
  Case Less Than:
    If (Point Value is < Value)
      Eliminate the decision
    End if
  Case Equal To
    If (Point Value = = Value)
      Eliminate the decision
    End if
End switch

```

 **Note:** Alphanumeric comparison is case sensitive.

Eliminate Decision by Solve

(Available with Order Execution Mgt.)

Description

Eliminate Decision by Solve eliminates a decision using specified query expression.

Parameters

This function block has the following parameters:

Parameter	Description	
Solve Expression	Expression used in the comparison. Do one of the following. <ul style="list-style-type: none"> Enter an existing (valid) query expression. Click the Query Browse button to open the Query Expression Browser and browse for or create a new expression. 	
Decision To Eliminate	ID of the decision to be eliminated, if the specified query expression evaluates to True.	
Region ID	Region that contains the first item to run the query against.	
Region Location	Region location that contains the item.	
	Use	For the
	PRT_FIRST	First item in the region.
	PRT_LAST	Last item in the region.
Item Class	Item class (page 161) to run the query against, e.g. vehicle.	

Eliminate Decision if not only Decision

Eliminate Decision if not only Decision

Description

Eliminate Decision if not only Decision eliminates the decision if there is more than one decision not yet eliminated.

Parameters

This function block has the following parameter:

Parameter	Description
Decision ID	The decision to be eliminated.

Eliminate Decision if not only Decision Example

Site is triggered and a logic module with a multiple Function Blocks is run. The last Function Block, Eliminate Decision If Not Only Decision, is executed as in the example.

Parameters

This function block has the following parameters:

Parameter Name	Parameter Value
Decision ID	SRC1 to DST1

1. Get the decision list.
2. The decision count is 1 because decision SRC2 to DST1 has been eliminated.
3. Decision SRC1 to DST1 has been eliminated.

Eliminate Decision if not only Decision Pseudo Code

```
Get the list of still possible decisions
If (the list count is greater than 1
  or ( decision count == 1 and decision[1] == decision id) then
  Eliminate decision
End If
```

Eliminate Decisions Based on Region Status

Eliminate Decisions Based on Region Status

Description

Eliminate Decisions Based on Region Status eliminates a decision based region status data of the source and destination region id for each decision.

INPUT

The input for this function block is a list of TRUE/FALSE combo boxes. In all cases select TRUE for the criteria to be selected.

1. If TRUE, eliminates decisions whose source region's head item is not validated.
2. If TRUE< eliminates decisions whose source region has a decision.
3. If TRUE, eliminates decisions whose source region is out-locked or destination region is in-locked.
4. If TRUE, eliminates decisions whose destination region is full.

OUTPUT

Decisions that meet the criteria are eliminated.

Eliminate Decisions Based on Region Status Pseudo Code

```

For each decision in the decision list.
  Get data on source region
  If (first parameter is true and head item not valid then
    Eliminate the decision.
  Else if (second parameter is true and region is waiting on a decision
  then
    Eliminate Decision
  Else if (third parameter is true and source region is out0locked then
    Eliminate Decision
  End if
Next
For each decision in the decision list.
  Get data on destination region
  If (third parameter is true and the destination region is in-locked then
    Eliminate Decision
  Else if (fourth parameter is true And the destination region is full
    Eliminate Decision
  End if
Next

```

Eliminate Decisions with no Items in Source

Eliminate Decisions with no Items in Source

Description

Eliminate Decisions with no Items in Source eliminates all decisions with no items in the source region's head location.

Parameters

This function block has the following parameters:

Parameter	Description	
Always Evaluate	Hard, Soft or Breakable. This parameter controls what happens when the extension eliminates all available decisions.	
	Hard	All decisions will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, None , is executed to handle this condition.
	Soft	Roll back any eliminations made by this extension.
	Breakable	The routing modules will first use the danger defined for each individual decision. If no danger point has been configured for the decision, the site-wide danger point will be used.
	TRUE (1)	Roll back any decisions eliminated made by this extension.

		FALSE (0)	The decision remains eliminated.
--	--	-----------	----------------------------------

Eliminate Decisions with no Items in Source Example

Site is triggered and a logic module with a single extension, Eliminate Decisions with No Items in Source, is run.

Decisions SRC1 to DST1 is eliminated because there are no items in the head location of the source region.

Eliminate Decisions with no Items in Source Pseudo Code

```

For each decision still possible
  Get the number of items at the source region's head location.
  If head location is empty
    Eliminate the decision.
  End if
Next

```

Get Trigger Name

Description

Get Trigger Name will retrieve the name of the trigger that activated the site. If the trigger was timed or automatic, the trigger named will be "Automatic"; if the trigger was manual, then the trigger named will be "Manual."

Parameters

This function block has no parameters.

Example

Not available for this release.

Pseudo Code

Not available for this release.

Return Value

The name of the trigger can be stored in an RCO Variable. Once the trigger name is stored, it can be logged or used as needed.

Item Type Translation

Item Type Translation

Description

Item Type Translation translates the input code to Item Type ID or vice versa, Item Type ID to input code, depending on the value of the output.

Parameters

This function block has the following parameters:

Parameter	Description
Translation ID	RCO Variable: specify the Translation ID to reference from the prt_type_trn configuration file.
Input Code	RCO Variable: this value must be supplied if the output is set to Item Type ID. This will be populated with the translated input code if output is set to input code.
Output	Specify the output to be either input code or Item Type ID.
Item Type ID	RCO Variable: This value must be supplied if the output is set to input code. This will be populated with the translated Item Type ID if output is set to Item Type ID.

Item Type Translation Pseudo Code

```

If Output = Input Code
  Input Code = ItemTypeTranslation Translation ID, Item Type ID
Else
  Item Type ID = ItemTypeTranslation Translation ID, Input Code

```

Look Back Blocking

Look Back Blocking

Description

Look Back Blocking eliminates decisions where the specified attribute in the tail item of the destination region match the corresponding attribute in one or more items following the head item of the source region.

Parameters

This function block has the following parameters:

Parameter	Description	
Always Evaluate	Hard, Soft or Breakable. This parameter controls what happens when the extension eliminates all available decisions.	
	Hard	All decisions will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, None , is executed to handle this condition.
	Soft	Roll back any decisions eliminated by this extension.
	Breakable	The routing modules will first use the danger defined for each individual decision. If no danger point can be found for the decision the site-wide danger point is used.
	TRUE (1)	Roll back any decisions eliminated by this extension.
	FALSE (0)	The decision remains eliminated.
Region	Region to search. If blank, each decision's source region will be searched.	
Last Location	Number of locations to look back. Item Class Class of item to compare.	
Attribute ID	Attribute of first item to compare. Start Character Starting character of the attribute to be used.	
Attribute Length	Specifies the number of characters in attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.	

Look Back Blocking Pseudo Code

```

For nlocation = 2 to look_back_factor
  For each decision
    If the decision is still possible
      If user specified region is blank
        Get the items in nLocation from the decision source region
      else
        Get the items in nLocation from the user specified region
      End if
      If number of items in region > 0
        Get items in the tail location of the decision destination region
        If number of items in tail location > 0
          If attribute matches for items with user specified class
            Eliminate current decision
          End if
        End if
      End if
    End if
  Next Decision
Next Location

```

Look Back Blocking Example

Parameters

This function block has the following parameters:

Parameter	Value
Always Evaluate	Hard
Region	""
Look Back Factor	3
Item Class	VEHICLE
Attribute ID	COLOR
Start Character	0
Attribute Length	0

1. Decision SRC1 to DST1:

Location 2, RED does not match destination region tail color attribute, YEL. Decision is not eliminated.

Location 3, BLUE does not match destination region tail color attribute, YEL. Decision is not eliminated.

2. Decision SRC2 to DST1:

Location 2, RED does not match destination region tail color attribute, YEL. Decision is not eliminated.

Location 3, YEL does match destination region tail color attribute, YEL. Decision is eliminated.

Look Back Blocking (Ext. Attr.)

Description


Look Back Blocking eliminates decisions where the specified standard or extended attribute in the tail item of the destination region match the corresponding attribute in one or more items following the head item of the source region.

Parameters

This function block has the following parameters:

Parameter	Description
Always Evaluate	Hard, Soft or Breakable. Controls what happens when the extension eliminates all items in the range source.

	Hard	All items will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.	
	Soft	Roll back any eliminations made by this extension.	
	Breakable	Roll back any eliminations made by this extension that have a danger point value of TRUE (non zero). Routing modules will first use the danger defined for each individual decision. However, one of the following may occur instead.	
		If:	Then:
		<ul style="list-style-type: none"> No danger point has been configured for the decision. 	The site-wide danger point will be used.
		<ul style="list-style-type: none"> This function block is used in the decision output logic. 	The site-wide danger point will be used.
		<ul style="list-style-type: none"> No danger point can be found for a decision. 	The decision remains eliminated.
Region Id	Region to search. If blank, each decision's source region will be searched.		
Last Location	Number of locations to look back. Item Class Class of item to compare.		
Item Class	Class of the item.		
Attribute	Name of the standard or extended attribute to compare.		
Attribute Start Byte	Starting character within the standard or extended attribute.		
Attribute Length	Specifies the number of characters in the attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.		

 **Note:** All the function blocks that deal with extended attributes can be provided `INTERNALHOLD` text to read the value of an internal hold.

Values are:

0	Internal Hold is not set.
1	Internal Hold is set.

However, you cannot use any of the extended attribute function blocks to modify system attributes.

Select Decision

Description

Select Decision checks to make sure that the decision made is valid.

Parameters

This function block has the following parameters:

Parameter	Description
Decision Name	ID of the decision to be selected.

Select Decision By Attribute

Select Decision By Attribute is obsolete. Use the current function blocks:

- Select Decision by Attribute Point
- Select Decision by Attribute Value
- Select Decision by Point Value

Select Decision by Attribute Point

Select Decision by Attribute Point

Description

Select Decision By Attribute Point selects a named decision based on the comparison of a point and an attribute.

Parameters

This function block has the following parameters:

Parameter	Description
Location	Location within the decision source region, of the item attribute to compare.
Item Class	Class of the item to compare.
Attribute ID	ID of attribute to compare.
Start Character	Starting character within attribute.
Attribute Length	Specifies the number of characters in attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.
Point ID	ID of point to use in comparison.
Decision	Name of decision to select.

Condition	Comparison operator, Match, Mismatch, Greater or Lessor.
Type of Comparison	Type of compare, either numeric comparison or alphabetic comparison.

Select Decision by Attribute Point Example

Parameters

This function block has the following parameters:

Parameter	Value
Location	PRT_FIRST
Item Class	VEHICLE
Attribute ID	COLOR
Start Character	0
Attribute Length	0
Point ID	COMPARE_POINT
Decision	SRC1_TO_DST1
Comparison	MISMATCH
Type of Comparison	Numeric

1. Value of point COMPARE_POINT is "BLUE".
2. Value of attribute of first item at the head location of region SRC1 is "YEL".
3. The values do not match, the decision is selected.

Select Decision by Attribute Point Pseudo Code

```


Get the value of the point
For each decision
  Get item at user specified location and region
  Get the value of the attribute
  Switch comparison type
    Case Match:
      If point value = attribute value
        Select the decision
      Exit sub
    End if
    Case MisMatch:
      If point value != attribute value
        Select the decision
      Exit sub
    End if
  Case Greater:

```

```

    If point value > attribute value
    Select the decision
    Exit sub
  End if
  Case Lessor:
    If point value < attribute value
    select the decision
    exit sub
  End if
End switch
Next Decision

```

 **Note:** If the specified decision is already eliminated, the function block will not re-select the decision or eliminate any other decisions.

If the decision was selected, the logic script will immediately exit.

Select Decision by Attribute Value

Select Decision by Attribute Value

Description

Select Decision by Attribute Value confirms that a decision is valid by making an attribute value comparison.

Parameters

This function block has the following parameters:

Parameter	Description
Location	Location within the decision source region, of the item attribute to compare.
Item Class	Class of the item to compare.
Attribute ID	ID of attribute to compare.
Start Character	Starting character within attribute.
Attribute Length	Specifies number of characters in attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.
Value	Value to use in comparison.
Decision	Name of decision to select.
Condition	Comparison operator, Match, Mismatch, Greater or Lessor.
Type of Comparison	Type of compare, either numeric comparison or alphabetic comparison.

Select Decision by Attribute Value Example

Parameters

This function block has the following parameters:

Parameter	Value
Location	PRT_FIRST
Item Class	VEHICLE
Attribute ID	COLOR
Start Character	0
Attribute Length	0
Value	BLUE
Decision	SRC1_TO_DST1
Condition	MISMATCH
Type of Comparison	Alpha

1. Compare Value "BLUE."
2. Value of attribute of first item at the head location of region SRC1 is "YEL."
3. The values do not match, the decision is selected.
4. The logic script exits.

Select Decision by Attribute Value Pseudo Code

```

For each decision
  Get item at user specified location and region
  Get the value of the attribute
  Switch comparison type
    Case Match:
      If point value = attribute value
        Select the decision
        Exit sub
      End if
    Case MisMatch:
      If point value != attribute value
        Select the decision
        Exit sub
      End if
    Case Greater:
      If point value > attribute value
        Select the decision
        Exit sub
      End if


```



```

Case Lessor:
  If point value < attribute value
    select the decision
    exit sub
  End if
End Switch
Next Decision

```

 **Note:** If the specified decision is already eliminated, the function block will not re-select the decision or eliminate any other decisions.

If the decision was selected, the logic script will immediately exit.

Select Decision By Extended Attribute

Description

Select Decision By Extended Attribute selects a named decision based on the comparison of an extended attribute of an item and a specified extended attribute, value, or point.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	The region where the item is.
Region Location	Location, within the decision source region, of the item attribute.
Item Class	Class of the item.
Attribute	Name of the extended attribute to compare.
Attribute Start Byte	Starting character within extended attribute.
Attribute Length	Specifies the number of characters in attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.
Decision to Select	Name of decision to select.
Condition	Comparison operator, Match, Mismatch, Greater or Lessor.
Comparison	Type of compare, either numeric comparison or alphabetic comparison.
Compare Against	Specifies where to get a comparison value: Extended Attribute, Value or Point.
Value to Compare	Specifies a value to compare against. Used if Compare Against is Value.
Point Name	ID of point to use in a comparison. Used if Compare Against is Point.

Region ID	The region where the item with an attribute to compare is. Used if Compare Against is Attribute.
Region Location	Location holding the item with an attribute to compare. Used if Compare Against is Attribute.
Item Class	Class of the item to compare. Used if Compare Against is Attribute.
Attribute Id	Name of the extended attribute to compare. Used if Compare Against is Attribute.
Attribute Start Byte	Starting character within extended attribute. Used if Compare Against is Attribute.
Attribute Length	Specifies the number of characters in attribute to compare. Used if Compare Against is Attribute. Enter zero in both the Start Character and Attribute Length fields to use all characters.

Select Decision By Ext. Attribute Point

Description

Select Decision By Ext. Attribute Point selects a named decision based on the comparison of an extended attribute of an item and a specified extended attribute, value, or point.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	The region where the item is.
Region Location	Location, within the decision source region, of the item attribute.
Item Class	Class of the item.
Attribute	Name of the extended attribute to compare.
Attribute Start Byte	Starting character within extended attribute.
Attribute Length	Specifies the number of characters in attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.
Decision to Select	Name of decision to select.
Condition	Comparison operator, Match, Mismatch, Greater or Lessor.
Comparison	Type of compare, either numeric comparison or alphabetic comparison.
Compare Against	Specifies where to get a comparison value: Extended Attribute, Value or Point.
Value to Compare	Specifies a value to compare against. Used if Compare Against is Value.
Point Name	ID of point to use in a comparison. Used if Compare Against is Point.

Region ID	The region where the item with an attribute to compare is. Used if Compare Against is Attribute.
Region Location	Location holding the item with an attribute to compare. Used if Compare Against is Attribute.
Item Class	Class of the item to compare. Used if Compare Against is Attribute.
Attribute Id	Name of the extended attribute to compare. Used if Compare Against is Attribute.
Attribute Start Byte	Starting character within extended attribute. Used if Compare Against is Attribute.
Attribute Length	Specifies the number of characters in attribute to compare. Used if Compare Against is Attribute. Enter zero in both the Start Character and Attribute Length fields to use all characters.

Select Decision by Ext. Attribute Value

Description

Select Decision by Attribute Value confirms that a decision is valid by making an attribute value comparison.

Parameters

This function block has the following parameters:

Parameter	Description
Location	Location within the decision source region, of the item attribute to compare.
Item Class	Class of the item to compare.
Attribute ID	ID of attribute to compare.
Start Character	Starting character within attribute.
Attribute Length	Specifies number of characters in attribute to compare. Enter zero in both the Start Character and Attribute Length fields to use all characters.
Value	Value to use in comparison.
Decision	Name of decision to select.
Condition	Comparison operator, Match, Mismatch, Greater or Lessor.
Type of Comparison	Type of compare, either numeric comparison or alphabetic comparison.

Select Decision by Point Value

Select Decision by Point Value

Description

Select Decision by Point Value confirms that a user specified decision is valid if the specified value matches a specified point.

Parameters

This function block has the following parameters:

Parameter	Description
Point ID	ID of point value to compare.
Value	Value to compare. May be hard coded or a RCO Variable.
Decision	ID of decision to select.
Condition	Type of comparison, Match, Mismatch, Greater or Lessor.
Type of Comparison	Type of compare, either numeric comparison or alphabetic comparison.

Select Decision by Point Value Example

Parameters

This function block has the following parameters:

Parameter	Value
Point ID	COMPARE_POINT
Value	3
Decision	SRC1_TO_DST1
Comparison operator	Greater
Comparison Type	Numeric

1. Compare value of point COMPARE_POINT to Compare Value of "3."
2. Specified value of "3" is not greater then point value of 3. Decision is not selected.

Select Decision by Point Value Pseudo Code


```

Get the value of the point
Get item at user specified location and region
Get the value of the attribute
Switch comparison type
  Case Match:
    If point value = value
      Select the decision
  
```

```

Exit sub
End if
Case MisMatch:
  If point value != value
    Select the decision
    Exit sub
  End if
Case Greater:
  If point value > value
    Select the decision
    Exit sub
  End if
Case Lessor:
  If point value < value
    select the decision
    exit sub
  end if
end switch

```

 **Note:** If the specified decision is already eliminated, the function block will not re-select the decisions or eliminate any other decisions.

If decision is selected, the logic script exits.

Select Decision by Solve

(Available with Order Execution Mgt.)

Description

Select Decision by Solve selects a decision using specified query expression.

Parameters

This function block has the following parameters:

Field	Description	
Solve Expression	Specifies the expression used in the comparison. Do one of the following. <ul style="list-style-type: none"> Type an existing (valid) query expression. Click the Query Browse button to open the Query Expression Browser to browser for or create a new expression. 	
Decision To Eliminate	ID of the decision to be selected, if the specified query expression evaluates to True.	
Region ID	ID of the Region that contains the first item to run the query against.	
Region Location	Specifies the region location.	
	Use	For the

	PRT_FIRST	First item in the region.
	PRT_LAST	Last item in the region.
Item Class	Class of item to run the query against.	

Select Decision Extended

Description

Select Decision Extended checks to make sure that the decision made is valid.

Parameters

This function block has the following parameters:

Parameter	Description
Decision Name	ID of the decision to be selected.

Select Decision with Highest Order & Weight

Select Decision with Highest Order & Weight

Description

Select Decision with Highest Order & Weight confirms that the decision with the largest weight is valid, using order as a tie breaker.

Parameters


This function block has no parameters.

Select Decision with Highest Order & Weight Pseudo Code

```

For each decision
  If decision is still possible
    If decision has highest weight
      Store decision offset
    End if
  End if
End if
Next decision
If any decisions found
  Select the decision with the highest weight and order
End if

```

 **Note:** The order of the decision is determined by its position in the Routing Control Configuration tree. The first decision in the tree has the lowest order.

If all decisions have already been eliminated, the logic script will not terminate.

Select Decision with Highest Order & Weight Example

1. Decision SRC1 to DST1, has weight of 10 and order of 1. This decision is tied for the highest weight, but not the highest order. This decision is eliminated.
2. Decision SRC2 to DST1, has a weight of 10 and an order of 2. This decision is tied for the highest weight, and also has the highest order. This decision is not eliminated.

Select Highest Order Decision

Select Highest Order Decision

Description

Select Highest Order Decision confirms that a decision made based on the highest order is valid.

Parameters

This function block has no parameters.

Select Highest Order Decision Example

1. Decision SRC1 to DST1, has an order of 1. This decision is not the highest order decision, so it is eliminated.
2. Decision SRC2 to DST1, has an order of 2. This decision has the highest order, so it is not eliminated.

Select Highest Order Decision Pseudo Code

```

For each decisions
  If the decision is still possible
    If the decision has the highest order
      Save the decision offset
    End if
  End if
Next decision
Select the decisions with the highest order

```

 **Note:** The decision order is determined by its position in the Routing Control Configuration tree.

After this decision executes, the logic script stops.

If all decisions have already been eliminated, the logic script will not terminate.

Select Lowest Order Decision

Select Lowest Order Decision

Description

Select Lowest Order Decision confirms that a decision made based on the lowest order is valid.

Parameters


This function block has no parameters.

Select Lowest Order Decision Pseudo Code

```

For each decisions
  If the decision is still possible
    If the decision has the lowest order
      Save the decision offset
    End if
  End if
Next decision
Select the decisions with the Lowest order
Exit Sub

```

 **Note:** The decision order is determined by its position in the Routing Control Configuration tree.

After this decision executes, the logic script stops.

If all decisions have already been eliminated, the logic script will not terminate.

Select Lowest Order Decision Example

1. Decision SRC1 to DST1, has an order of 1. This decision is the lowest order decision that is still possible and therefore is not eliminated.
2. Decision SRC2 to DST1, has an order of 2. This decision is not the lowest order decision and therefore is eliminated.

Select Most Empty Region

Select Most Empty Region

Description

Select Most Empty Region confirms that a decision is valid based on the most empty destination region.

Parameters

This function block has no parameters.

Select Most Empty Region Example


1. For decision SRC1 to DST1, destination region has a quantity of 2. Since this destination region has the lowest quantity, the decision is not eliminated.
2. For Decision SRC1 to DST2, the destination region has a quantity of 3. Since this destination region does not have the lowest quantity, the decision is eliminated.

Select Most Empty Region Pseudo Code

```

For each decisions
  If the decision is still possible
    If the decision source region has a Quantity point in Region
      Maintenance
        Get the value of the quantity point
        If the region quantity is less than the min
          Store the decision as the min quantity
        End if
      End if
    End if
  End if
Next decision
If found a decision with min quantity
  Select the decision
End if
Exit Sub

```

 **Note:** In the case where multiple decisions have the same destination regions, containing the minimum quantity or when multiple destination regions have the same quantity, the decision with the lowest order is selected.

If all decisions have already been eliminated, the logic script will not terminate.

Select Most Full RegionSelect Most Full Region**Description**

Select Most Full Region confirms that a decision is valid based on the most full source region.

Parameters

This function block has no parameters.

Select Most Full Region Example


1. Decision SRC1 to DST1, has a quantity of 2, in its source region.. This is not the source region containing the highest quantity, so eliminate the decision.
2. Decision SRC2 to DST1, has a quantity of 3 in its source region. Since this source region has the highest quantity, the decision is not eliminated.

Select Most Full Region Pseudo Code

```

For each decisions
  If the decision is still possible
    If the decision source region has a Quantity point in Region
      Maintenance
        Get the value of the quantity point
        If the region quantity is greater than the max
          Store the decision as the max quantity
        End if
      End if
    End if
  End if
Next decision
If found a decision with max quantity
  Select the decision
End if
Exit Sub

```

 **Note:** In the case where multiple decisions have the same source region with a maximum quantity or when multiple source regions have the same quantity, the decision with the lowest order is selected.

If all decisions have already been eliminated, the logic script will not terminate.

Select Next Highest Order Decision

Select Next Highest Order Decision

Description

Select Next Highest Order Decision confirms that a decision is valid based on the next highest order after the last executed decision.

Parameters

This function block has no parameters.

Select Next Highest Order Decision Example


1. The last decision executed was SRC2 to DST1.
2. Since there is no decision with a higher order, so the all decisions but the first decision are eliminated.

Select Next Highest Order Decision Pseudo Code

```

If all decisions have not already been eliminated
  Get order of last decision selected
  If the last decision selected had the highest order
    Set the current decision to the first
  Else
    Set the current decision to the next order
  End if
While not done
  If current decision has been eliminated
    If the current decision has the highest order
      Set the current decision to the first decision
    Else
      Set the current decision to the next order
    End if
  Else
    Select the current decision
    Set done flag
  End If
End while
Exit Sub

```

 **Note:** After this block, the logic module terminates.

*Select Oldest Item**Select Oldest Item***Description**

Select Oldest Item confirms that the decision is valid based on the lowest standard attribute value in its source region.

This does not eliminate any other decision.

Parameters

This function block has the following parameters:

Parameter	Description
Item Class	Class of Item to be used in the comparison.
Attribute Name	Name of attribute to be used in comparison.
Start Character	Start character within the attribute value.
Number of Characters	Number of attribute value characters used in the comparison. This allows the operator to use a subsection of the attribute. To use the entire attribute value in the comparison, set this value to 0 .

*Select Oldest Item Example***Parameters**

This function block has the following parameters:

Parameter	Value
Item Class	VEHICLE
Attribute Name	COLOR
Start Character	0
Number of Characters	0


1. Decision SRC1 to DST1 has a COLOR Attribute value of **YEL**. Since this is not the lowest attribute value, the decision is eliminated.
2. Decision SRC2 to DST1 has a COLOR Attribute value of **GRN**. Since this is the lowest attribute value, the decision is not eliminated.

Select Oldest Item Pseudo Code

```

For each decision
  If the decision is still possible
    Get the specified attribute item
    If item found that matches specified item class
      Get the attribute value
      If attribute was found
        If the present attribute value < the lowest yet
          Set lowest yet to present attribute value
          Store decision offset
        End if
      End if
    End if
  End if
Next decision
If a decision was found with a lowest attribute
  Eliminate all decisions but the one with the lowest attribute
Else
  If we didn't find any items with specified item class
    Eliminate all decisions
  End if
End if
Exit sub

```

 **Note:** If no items of the specified item class are found in any of the decisions source regions, then all decisions will be eliminated.

The Logic module will exit after this block.

Select Oldest Item (Ext. Attr.)

Description

Select Oldest Item (Ext. Attr.) confirms that the decision is valid based on the lowest standard or extended attribute value in its source region.

This does not eliminate any other decision.

Parameters

This function block has the following parameters:

Parameter	Description
Item Class	Class of Item to be used in the comparison.
Attribute Name	Name of attribute to be used in comparison.

Start Character	Start character within the attribute value.
Number of Characters	Number of attribute value characters used in the comparison. This allows the operator to use a subsection of the attribute. To use the entire attribute value in the comparison, set this value to 0.

Range Source Function Blocks

RSA Function Block List

Range Source Architecture function blocks include:

About Attribute Value Wildcards	Record Attribute Rule
Advance in Order	Record Attribute Spacing
Alarm On No Sources Remaining	Record Violated Rules
Apply Hold To Item	Release Hold From Item
Apply Hold To Range	Release Hold For Range
Begin Spread Range Block Module	Reorder Region by Rotation Number
Eliminate Attribute Mismatch	Resequence Mode Begin
Eliminate Attribute Value Mismatch	Resequence Range
Eliminate By Solve	Rotation Pull Ahead
Eliminate By Weight (this block also appears in Routing Function Blocks)	Select Lowest Weight
Eliminate Capability Mismatch	Select Source By Age
Eliminate Source By Pattern	Set Attribute (Range Source)
Eliminate Sources Based On Attribute Spacing	Set Block Weights
Eliminate Sources Based On Rule	Set Point With Attribute
Eliminate Source by Percentages	Set Rotation Number to Attribute
Eliminate Source By Ratio	Set Source Criteria Set
End Spread Range Block Module	Set Source Range
Move Item from Range	Set Source Solve
Range Source Eliminate By Expression Mismatch	Set Super-Priority Weight
Record Attribute Pattern	Set Variable With Attribute
Record Attribute Percentages	Substitute Attributes And Status
Record Attribute Ratios	Substitute Order

About Attribute Value Wild Cards

RSA function blocks that have Attribute Value as a parameter support the following wild cards and characters for matching special values for standard or extended attribute values.

- Wild cards
- Match special characters

Wild cards		Example	
Wild card	Description	Wild card	Possible Results
?	Any single character.	T?m	Tam Tim Tom
		12?0	1200 1220 12A0*
*	Zero or more characters.	Ma*	Ma Mark Manilla
		10*	10 1
#	Any single digit (0-9).	200#	2000 2001 2003

Match Special Characters		Example		
Character		Group	Match	
[]	Encloses characters, including wild cards (page 533) , to match.	abc[*]	abc*	
[Can be enclosed in a match character group. Note: The right bracket,], cannot be used within a group to match itself, but it can be used outside of any brackets as an individual character.	[[]abc]	[abc]	
[charlist]	Any character that matches one or more enclosed characters.	t[io]p	Yes	tip top
	Note: The character sequence [] is considered a zero-length string ("").		No	tap
[!charlist]	Any character that is not enclosed in brackets.	t[!io]p	Yes	tap
			No	tip top
[char-list]	(Hyphen) separates the upper and lower bounds of a charlist to identify a range.	[0-5]AB	Yes	0AB 1AB
	Important: When a range of characters is specified, they must appear in ascending sort order (from lowest to highest), [A-Z] is a valid pattern. [Z-A] is not a valid pattern.		No	6AB
[char-list] [char-list]	Matches multiple ranges.	[A-M] [0-5]	Yes	A1 M5
	Note: There are no delimiters between ranges.		No	N6 A9 Z0
[-charlist]	Matches itself when it is:	[-201]	Yes	-21

	<ul style="list-style-type: none"> • At the beginning of a charlist. 		No	33
	<ul style="list-style-type: none"> • After an exclamation point. 	[ABCD-]	Yes	ABC-
	<ul style="list-style-type: none"> • At the end of charlist. 		No	DEF

Advance in Order


(Available with Order Execution Mgt.)

Description

Advance in Order assigns the lowest rotation number in the range source to the item at a particular location in a specified region. The rotation numbers of other items in the range source are then adjusted to maintain their original order after the advanced item.

Note:

- Rotation numbers must already be assigned to items in the range source by a [Resequence Mode Block \(page 571\)](#) module earlier in the routing or output logic module. The Advance in Order function block should be added somewhere after the Resequence Range function block to advance a particular item.
- The entire range source is searched for a the lowest rotation number, however, only items local to the current project are modified. You can apply this function block across multiple projects by entering a comma-delimited list of project names into the **Project ID** field and recompiling.

 **Note:** if an item has been physically pulled ahead to the first location in a region, use this function block to adjust the rotation numbers accordingly.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	ID of Region where the item is located.
Region Location	An integer value representing the location of the item in the Region. Note: The reserved names PRT_FIRST or PRT_LAST can be entered here to specify the first or last location, respectively, within the region. The first location is at the head of the region, the last location is at its tail.
Item Class	Class of the item to be assigned a rotation number.

Rotation Number Attribute	The name of the attribute that stores the rotation number.
Rotation Number Point ID	(Optional) Point that contains an attribute name. Supercedes Rotation Number Attribute.
Project ID	(Optional) Names of one or more projects in which the selected region may be found. Note: Leave this field blank to limit the search for regions to the local project only.
Project Point ID	(Optional) Text point. Stores a list of project(s). Supercedes Project ID.

Alarm on No Sources Remaining

(Available with Order Execution Mgt.)

Description

Alarm on No Sources Remaining generates an alarm if there are no items left in the range source. An option is provided to terminate the routing or output logic when the alarm is generated.

Parameters

This function block has the following parameters:

Parameter	Description
Alarm Name	ID of the alarm to generate. Must be a valid alarm of type \$CIMBASIC.
Project Id	Project to generate the alarm on. An empty string "" indicates the current project.
Resource Id	Resource to generate the alarm against. Used to control routing of the alarm.
Userld\$	String (optional). A User ID for the generated alarm. This information appears in the alarm viewer if a custom button or field is configured to display it (%USER).
Reference ID	String (optional). A Reference ID used to distinguish identical alarms. Enter a name for the reference ID, or select an RCO variable that has been assigned this string value.
Alarm Message	Message for the alarm. Use "%s" for any of the user parameters below that you have configured Example Alert! RCO variable 1 is: %s UserValue1 is substituted for % during and instance of the alarm message. Alert! RCO variables are: %s %s %s UserValue1, UserValue2 and UserValue3 are substituted for the %'s with a space between the values.
UserValue1	User defined value to be inserted into the alarm message.
UserValue2	User defined value to be inserted into the alarm message.
UserValue3	User defined value to be inserted into the alarm message.
UserValue4	User defined value to be inserted into the alarm message.
Exit Logic?	Whether to terminate logic processing if no sources remain.


	YES	(Default) Terminate the logic.
	NO	Do not terminate the logic.

Apply Hold to Item

(Available with Order Execution Mgt.)

Description

Apply Hold to Item places a named hold on the item at a particular location in a specified region.

 **Note:** Only items local to the current project are modified. You can apply this function block across multiple projects by entering a comma-delimited list of project names into the **Project ID** field and recompiling.


Note:

A named hold is a special type of attribute that specifies the range in which the hold is active. The item will actually be in hold state only when it physically enters the range where the hold is active..

The range is retrieved from the given Hold Active Range Point name.

Parameters

This function block has the following parameters:

Parameter	Description
Region Name	Name of the Region where the item is currently located.
Region Location	Location of the item in the Region. Select a location, or enter a value greater than zero that specifies a location. Note: The first location, i.e. 1, is at the head of the region, the last location is at its tail. Items generally move from tail to head.
Item Class	Class of the item to place on hold.
Hold Attribute Name	Name of the standard attribute that stores an item's hold name. Note: You can configure the standard attributes that appear in this list by selecting Tools>Attribute Maintenance on the TrackerCfg_UI window menu bar.
Hold Expiration Period	Number of hours until the hold expires, starting at the time the function block executes. Enter a numeric value.  Important: The numeric value must be manually entered; it cannot be retrieved from an analog point. Enter a value of 0 to specify an unlimited hold.
Hold Exp. Time Point	(Optional) Date-time real double analog point Time at which the hold will expire. Selecting a point for this parameter overrides the Hold Expiration Period, which it supercedes.


Hold Description	(Optional) Description saved with the hold. The description is stored in the extended attributes for items on hold.
Hold Attribute Point	(Optional) Text point Contains dynamically changeable hold attribute name Supercedes Hold Attribute Name.
Hold Active Range Point	Range where the hold is active. The range is retrieved from the given range point. Note: The hold placed on the specified item does not take effect until it enters the named range or region. Supercedes Hold Active Range.
Project ID	(Optional) Names of one or more projects in which the selected region may be found. Note: Leave this field blank to limit the search for regions to the local project only.
Project Point ID	(Optional) Text point. Stores a list of project(s). Supercedes Project ID.

Apply Hold to Range

(Available with Order Execution Mgt.)

Description

Apply Hold to Range places a named hold on all items that remain in the range source after any possible eliminations. This hold becomes active when these items enter another range, the [Hold Active Range \(page 537\)](#).

 **Note:** A named hold is a special type of attribute that specifies the range in which the hold is active. An item will actually be in hold state only when it physically enters the Hold Active Range.

Parameters

This function block has the following parameters:

Parameter	Description
Hold Attribute Name	Extended attribute to store the hold status.
Hold Active Range	Named range in which the hold will be active. Enter the name of a defined range or region. Note: The hold placed on the specified item does not take effect until it enters the named range or region.
Hold Expiration Period	Number of hours until the hold expires. Enter a value or the name of an analog (numeric) point to define the hold period. Enter a value of 0 to specify an unlimited hold.
Hold Exp. Time Point	(Optional) Analog (real double) Date-time point. Defines the time at which the hold will expire. Select a point for this parameter overrides the Hold Expiration Period.
Hold Description	(Optional) Description saved with the hold. The description is stored in the extended attributes for items on hold.

Hold Attribute Point	Text point Holds an attribute name. Overrides Hold Attribute Name.
Hold Active Range Point	Text point Holds the name of a range. Overrides Hold Active Range.

Begin Spread Range Block Module

(Available with Order Execution Mgt.)

Description

Begin Spread Range Block Module marks the start of a multiple-pass Spread Range Block Module in the routing or output logic module.

1. Whenever a range is used the logic module must begin with two blocks in the following order.

Block	Module Options	Purpose
1	Can be either:	
	Set Source Range	Defines the range that the block evaluates.
	Set Source Solve	
2	Must be:	
	Set Block Weights	Sets the static importance of each of the subsequent blocks by assigning a weight to each block.

2. An [End Spread Range Block Module \(page 558\)](#) block is required at the end of the Spread Range Block Module.

Other RSA function blocks, within the Spread Range Block Module, perform an evaluation on an initial number of items within a named range.

If none of these items remain unweighted after the evaluation (or if no items remain) the:

- a. Number of items included in the evaluation set is incremented
- b. Expanded set is re-evaluated by a second pass by the Spread Range Block Module logic.
- c. Evaluation repeats multiple times, up to the number of passes you specify, until at least one unweighted item remains.

Note: Tips

3. Use a Spread Range Block Module to define a dynamic range that eliminates unsuitable items but keeps expanding until at least one item suitable for subsequent processing remains.

4. Add an [Alarm on No Sources Remaining \(page 535\)](#) function block after your Spread Range Block module to address the possibility that no unweighted items remain in the range source list after the final pass.

Parameters

This function block has the following parameters:

Parameter	Description
Range Name	Name of a range of sources from which a spreading range of items are evaluated. . a. Double-click Range Name to display the browse dialog for ranges. b. Do one of the following: <ul style="list-style-type: none"> • Enter a existing range or region name • Click the Browse button to select from a list of existing ranges. • Click the Pop-up button and: <ul style="list-style-type: none"> • Select Browse to select from a list of existing ranges, or • Select New to define a new range. The Range Viewer dialog box appears.
Item Class	Class of item to include within the range source.
Initial Range Source Length	Number of items to be evaluated in the first pass.
Source Increment	Number of source items to add to the evaluation set in each subsequent pass.
Passes	Maximum number of passes to be attempted.
Range Point ID	(Optional) Text point. Stores the name of a range. Supercedes Range Name.
Length Point ID	(Optional) Integer point. Stores the number of items evaluated in the first pass. Supercedes Initial Range Source Length.
Increment Point ID	(Optional) Integer point Stores the increment value. Supercedes Source Increment.
Passes Point ID	(Optional) Integer point. Stores the number of passes. Supercedes Passes.

5. Double-click **Range Name** to display the browse dialog for ranges.
6. Do one of the following:
- Enter a existing range or region name
 - Click the Browse button to select from a list of existing ranges.
 - Click the Pop-up button and:
 - Select Browse to select from a list of existing ranges, or
 - Select New to define a new range. The Range Viewer dialog box appears.

Eliminate Attribute Mismatch


(Available with Order Execution Mgt.)

Description

Range Source Eliminate Attribute Mismatch applies a weight to items with an attribute value that does not match the corresponding attribute value of a particular item, specified by that item's region, location, and class.

The applied weight is a value that:

- You entered in the [Set Block Weights \(page 575\)](#) function block's corresponding parameter.
- Set Block Weights assigns to this function block.


 **Note:** A block that applies a zero weight eliminates items that fail from the range source.

Parameters

This function block has the following parameters:

Parameter	Description		
Always Evaluate	Hard, Soft or Breakable. Controls what happens when the extension eliminates all items in the range source.		
	Hard	All items will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.	
	Soft	Roll back any eliminations made by this extension.	
	Breakable	Roll back any eliminations made by this extension that have a danger point value of TRUE (non zero). Routing modules will first use the danger defined for each individual decision. However, one of the following may occur instead.	
		<ul style="list-style-type: none"> • No danger point has been configured for the decision. 	The site-wide danger point will be used.
		<ul style="list-style-type: none"> • This function block is used in the decision output logic. 	The site-wide danger point will be used.
		<ul style="list-style-type: none"> • No danger point can be found for a decision. 	The decision remains eliminated.
Attribute Name	The name of the attribute to compare.		
Attribute Start Byte	(Optional) An integer specifying the byte address (offset) of the first byte within the attribute to be included in the value comparison. Leave the Attribute Start Byte and Attribute Length fields unspecified to compare actual attribute values.		
Attribute Length	(Optional) An integer specifying the number of bytes, beginning with the Attribute Start Character, to include in the value comparison.		

Region Name	The region to check.		
Region Location	Select a region location or enter a value greater than one.		
Item Class	The item class whose attribute is to be checked.		
Attribute Point ID	(Optional) Text point Stores the name of an attribute. Supercedes Attribute Name.		
Region Point ID	(Optional) Text point Stores the name of a PRT Region. Supercedes Region Name.		

 **Note:** You can compare the value of an attribute, or the value of a specified number of bytes within an attribute.

Option	Attribute Start Character is:		Attribute Length is:	Result
1	Specified	and	Specified	The value of the specified bytes, taken as a whole, is compared.
2	Specified	and	Not specified	A one byte value is compared.
3	Not specified	and	Not specified	The binary value of the attribute is compared.

Eliminate Attribute Value Mismatch

(Available with Order Execution Mgt.)

Description

Range Source Eliminate Attribute Value Mismatch applies a weight to items with an attribute value that does not match the value of a specified RCO variable.

The applied weight is a value that:

- You entered in the [Set Block Weights \(page 575\)](#) function block's corresponding parameter.
- Set Block Weights assigns to this function block.

Important:

- A block that applies a weight of zero directly eliminate items that fail from the range source.
- This block is designed to work only when a valid danger point is configured.

A danger point:

- Must be either of two types:

- Integer, long.
- Boolean.

All other types throw an exception.

- Can be device or virtual point.
- Can have a point value of zero or non zero.

The danger point value is considered in elimination decision, when the value is non-zero and if the evaluation type is Breakable (neither Soft nor Hard) type.


If an invalid danger point or no danger point is configured, the function block will not work and throws CORLOG warning.

Parameters

This function block has the following parameters:

Parameter	Description									
Always Evaluate	Hard, Soft or Breakable. Controls what happens when the extension eliminates all items in the range source.									
	Hard	All items will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.								
	Soft	Roll back any eliminations made by this extension.								
	Breakable	Roll back any eliminations made by this extension that have a danger point value of TRUE (non zero). Routing modules first use the danger defined for each individual decision. However, one of the following may occur instead.								
		<table border="1"> <thead> <tr> <th>If:</th> <th>Then:</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> • No danger point has been configured for the decision. </td> <td>The site-wide danger point will be used.</td> </tr> <tr> <td> <ul style="list-style-type: none"> • This function block is used in the decision output logic, </td> <td>The site-wide danger point will be used.</td> </tr> <tr> <td> <ul style="list-style-type: none"> • No danger point can be found for a decision. </td> <td>The decision remains eliminated.</td> </tr> </tbody> </table>	If:	Then:	<ul style="list-style-type: none"> • No danger point has been configured for the decision. 	The site-wide danger point will be used.	<ul style="list-style-type: none"> • This function block is used in the decision output logic, 	The site-wide danger point will be used.	<ul style="list-style-type: none"> • No danger point can be found for a decision. 	The decision remains eliminated.
If:	Then:									
<ul style="list-style-type: none"> • No danger point has been configured for the decision. 	The site-wide danger point will be used.									
<ul style="list-style-type: none"> • This function block is used in the decision output logic, 	The site-wide danger point will be used.									
<ul style="list-style-type: none"> • No danger point can be found for a decision. 	The decision remains eliminated.									
Attribute Name	The ID of the attribute to base the comparison on.									
Attribute Start Byte	(Optional) An integer that specifies the byte address (offset) of the first byte within the attribute that is included in the value comparison.									
Attribute Length	(Optional) An integer that specifies the number of bytes, beginning with the Attribute Start Character, to include in the value comparison.									

Attribute Value	The value to match. Do one of the following. <ul style="list-style-type: none"> • Select an assigned RCO Variable. • Enter a constant value. <p>Note: Supports attribute value wild cards (page 532).</p>		
Attribute Point ID	(Optional) Text point Stores the name of an attribute. Supercedes Attribute Name.		

 **Note:** You can compare the value of an attribute, or the value of a specified number of bytes within an attribute.

Option	Attribute Start Character is:		Attribute Length is:	Result
1	Specified	and	Specified	The value of the specified bytes, taken as a whole, is compared.
2	Specified	and	Not specified	A one byte value is compared.
3	Not specified	and	Not specified	The binary value of the attribute is compared.

Eliminate By Solve


(Available with Order Execution Mgt.)

Description

Range Source Eliminate By Solve applies a weight to items that do not match the criteria of a named query or query expression.

The applied weight is a value that:

- You entered in the [Set Block Weights \(page 575\)](#) function block's corresponding parameter.
- Set Block Weights assigns to this function block.

 **Note:** A block that applies a zero weight directly eliminates items that fail from the range source.

Parameters

This function block has the following parameters:

Parameter	Description	
Always Evaluate	Hard, Soft or Breakable. Controls what happens when the extension eliminates all items in the range source.	
	Hard	All items will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.

	Soft	Roll back any eliminations made by this extension.	
	Breakable	Roll back any eliminations made by this extension that have a danger point value of TRUE (non zero). Routing modules will first use the danger defined for each individual decision. However, one of the following may occur instead.	
		If:	Then:
		<ul style="list-style-type: none"> No danger point has been configured for the decision. 	The site-wide danger point will be used.
		<ul style="list-style-type: none"> This function block is used in the decision output logic, 	The site-wide danger point will be used.
		<ul style="list-style-type: none"> No danger point can be found for a decision 	The decision remains eliminated.
Solve Id/ Expression	Enter one of the following to compare items against. <ul style="list-style-type: none"> The identifier of a named query. A query expression, Items that do not satisfy the query criteria fail the comparison.		
Solve Point ID	(Optional) A text point that stores the name of a query. Supercedes Solve Id/Expression.		
Constant Solve Expression	(Optional) Valid BASIC expression to generate a solve or solve ID. Supercedes Solve ID and Solve Point ID.		
	Default	<none>	

Eliminate By Weight

(Available with Order Execution Mgt.)

Description

Eliminate By Weight eliminates all items within the range source that have a weight greater than a specified threshold value.

Eliminate By Weight

- Does not apply weights. No broken rule history is created for the items this block eliminates.
- Can be added to the routing or output logic after other RSA function blocks that eliminate or weight items.

The weights that function blocks apply are determined by the [Set Block Weights \(page 575\)](#) function block.

Parameters

This function block has the following parameters:

Parameter	Description		
Always Evaluate	Hard, Soft or Breakable. Controls what happens when the extension eliminates all items in the range source.		
	Hard	All items remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.	
	Soft	Roll back any eliminations made by this extension.	
	Breakable	Roll back any eliminations made by this extension that have a danger point value of TRUE (non zero). Routing Logic modules first try to use the danger defined for each individual decision. However, one of the following may occur instead:	
		If:	Then:
		<ul style="list-style-type: none"> No danger point has been configured for the decision. 	The site-wide danger point is used.
		<ul style="list-style-type: none"> This function block is used in the decision output logic. 	The site-wide danger point is used.
		<ul style="list-style-type: none"> No danger point can be found for a decision 	The decision remains eliminated.
Threshold Value	Items with a weight greater than this value are eliminated.		
Threshold Point ID	(Optional) A point configured as an integer that stores the threshold value. Supercedes Threshold Value.		

Eliminate Capability Mismatch


(Available with Order Execution Mgt.)

Description

Range Source Eliminate by Capability Mismatch applies a weight to items for which the value of a specified attribute does not match the value of any point within RCO's set of capability points.

The applied weight is a value that:


- You entered in the [Set Block Weights \(page 575\)](#) function block's corresponding parameter.
- Set Block Weights assigns to this function block.

 **Note:** A block that applies a zero weight directly eliminates items that fail from the range source.

Parameters

This function block has the following parameters:

Parameter	Description									
Always Evaluate	Hard, Soft or Breakable. Controls what happens when the extension eliminates all items in the range source.									
	Hard	All items will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.								
	Soft	Roll back any eliminations made by this extension.								
	Breakable	Roll back any eliminations made by this extension that have a danger point value of TRUE (non zero). Routing modules will first use the danger defined for each individual decision. However, one of the following may occur instead.								
		<table border="1"> <thead> <tr> <th>If:</th> <th>Then:</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> • No danger point has been configured for the decision. </td> <td>The site-wide danger point will be used.</td> </tr> <tr> <td> <ul style="list-style-type: none"> • This function block is used in the decision output logic. </td> <td>The site-wide danger point will be used.</td> </tr> <tr> <td> <ul style="list-style-type: none"> • No danger point can be found for a decision. </td> <td>The decision remains eliminated.</td> </tr> </tbody> </table>	If:	Then:	<ul style="list-style-type: none"> • No danger point has been configured for the decision. 	The site-wide danger point will be used.	<ul style="list-style-type: none"> • This function block is used in the decision output logic. 	The site-wide danger point will be used.	<ul style="list-style-type: none"> • No danger point can be found for a decision. 	The decision remains eliminated.
If:	Then:									
<ul style="list-style-type: none"> • No danger point has been configured for the decision. 	The site-wide danger point will be used.									
<ul style="list-style-type: none"> • This function block is used in the decision output logic. 	The site-wide danger point will be used.									
<ul style="list-style-type: none"> • No danger point can be found for a decision. 	The decision remains eliminated.									
Attribute Name	ID of the attribute to base the comparison on.									
Attribute Start Byte	(Optional) An integer that specifies the byte address (offset) of the first byte within the attribute is included in the value comparison.									
Attribute Length	(Optional) An integer that specifies the number of bytes, beginning with the Attribute Start Byte, to include in the value comparison.									
Region Name	Region where the capability point to be compared with an attribute is assigned.									
Capability Point Number	Capability point (Point #1, Point #2, or Point #3) from that region.									
Attribute Point ID	(Optional) Text point. Stores the name of an attribute. Supersedes Attribute Name.									

 **Note:** You can compare the value of an attribute, or the value of a specified number of bytes within an attribute.

Option	Attribute Start Character is:		Attribute Length is:	Result
1	Specified	and	Specified	The value of the specified bytes, taken as a whole, is compared.
2	Specified	and	Not specified	A one byte value is compared.
3	Not specified	and	Not specified	The binary value of the attribute is compared.

Eliminate Source By Pattern

Eliminate Source By Pattern

The function block compares the first Attribute Value element with the value for that attribute from the current item. If no matching Attribute Value is found:

1. The function block continues to look for a match by examining other items.
2. If still no match is found the:
 - Pattern History Point value is automatically incremented.
 - Function block attempts to find a match for the next attribute name/value pair until a match is found.
3. This process continues until either of the following occurs.
 - A matching value is found.
 - All items and attribute name/value pairs have been compared without a match.
4. The current item fails the comparison and is either weighted or eliminated. The failure is saved as history data.
5. The next item is compared to the same attribute name/value pair.
6. This process continues until a matching value is found or all items have been weighted or eliminated.
When a matching Attribute Value is found:
7. The function block compares the same Attribute Value element with the value of the corresponding attribute for the next item.
8. At the conclusion of the pass, the function block stores an index value that corresponds to the position of the next element after the array element for which a match was found. This value is stored to the Pattern History Point. **Note:** After reaching the last element in the array, the Pattern History Point is reinitialized; the end of the pattern wraps back to the beginning comparison with any subsequent items. **Notes**

- An Attribute Name element containing the reserved name ANY is a wildcard. The current item will always be selected as a match for this index into the pattern. This reserved name does not require a value; i.e. you can enter any value or no value for the respective Attribute Value element.
- To compare items that satisfy a named solve (query) or query expression, enter the reserved name SOLVE for this element in the Attribute Name array. (This is instead of the name of an attribute). You can then enter a named query or query expression as the respective element in the Attribute Value array.

Eliminate Source By Pattern Example

The table below shows the elements of the array points defined Eliminate Sources Based on Rule organized as a table.

Columns and rows represent the following.

Column	Elements of one of the array points
Row	Pattern that defines the order in which items (that have attributes matching those in the column) are selected.

The yellow row represents the index in the pattern that specifies the attribute name/value pair of the next item to select.

	Attribute Name Array	Attribute Value Array:	Pattern History Point:
	Color	Blue	
	Color	Green	
Next	Color	Red	2
	Any	---	
	Doors	4	
	Solve	\$IsRed4Door\$	
	Solve	\$FinishIs\$	

Eliminate Sources Based on Attribute Spacing

1. The function block attempts to match the respective Attribute Value element with a value for that attribute from the current item.
2. When an attribute value is matched, the function block compares the respective Spacing Rule and Spacing Count elements.
3. If selecting the current item (incrementing the Spacing Count) violates the rule, the:
 - a. Comparison fails for that attribute.
 - b. Item is either weighted or eliminated.
 - c. Failure is saved as history data.

Notes:

- An elimination may be rolled back depending on the setting of the respective danger flag in the Breakable array.
- To compare items that satisfy a named query or query expression, enter the reserved name SOLVE for this element in the Attribute Name array. (This is instead of the name of an attribute). You can then enter a named query or query expression as the respective element in the Attribute Value array.

*Eliminate Sources Based on Rule**Eliminate Sources Based on Rule*

(Available with Order Execution Mgt.)

Description

Eliminate Sources Based on Rule eliminates or applies a weight to items in the range source if their selection breaks a business rule.

The applied weight is a value that:

- You entered in the [Set Block Weights \(page 575\)](#) function block's corresponding parameter.
- Set Block Weights assigns to this function block.

Overview

A business rule assigns any of the following.

Rule Type	For items with the same selected attribute values the rule dictates the:
Group	Maximum number of consecutive items.
Spacing	Minimum separation between items.
Blocking	Minimum number of consecutive items.
Percent	Minimum or maximum percentage of the total number of items.

- If a business rule is violated, the Item is eliminated / weighted, or the Failure is saved as history data. Failed items are eliminated from the range source when a block applies a zero weight.
- An accurate historical record of the item selections is required.

When you use this block, insert a corresponding Record Attribute function block into your routing or output logic module.

Insert a Record Attribute function block where all item eliminations (by any and all methods) have been completed.

Example

If the `Eliminate Sources Based on Rule` function block is placed within a Resequencing Block Module, then the corresponding Record Attribute function block should be inserted immediately before the [Resequencing Range \(page 572\)](#) block where the selection is processed.

The `Eliminate Sources Based on Rule` and corresponding Record Attribute function blocks should be configured with the same points and array point names.

Rotation numbers can be assigned to items.

1. Apply a zero weight to `Eliminate Sources Based on Rule`.
2. Use `Eliminate Sources Based on Rule` within a multiple-pass Resequencing Block Module.

Parameters

This function block has the following parameters:

Parameter	Description	
Always Evaluate	Hard, soft or Breakable. Controls what happens when the extension eliminates all items in the range source.	
	Hard	All items will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.
	Soft	Roll back any eliminations made by this extension.
	Breakable	If the danger flag is set to:
	TRUE (1)	Roll back any eliminations made by this extension.
	FALSE (0)	The item remains eliminated even if all items in the range source are eliminated.
	Note: The danger flag is an element of the Breakable (page 553) array described below.	
Rule Type Array	A text array point. Each element in the array must contain a rule type.	
	Rule	Item is:
	G	Eliminated if the Count exceeds or equals the Rule.
	S	Eliminated if the Count is less than the Rule.
	B	Eliminated if the Count is less than the Rule.
Example (page 553)	P	Eliminated if: <ul style="list-style-type: none"> • The percentage of the total for this attribute would exceed Upper Release % Limit, or • The percentage of the total for another percentage rule (defined for a different attribute name/value pair) would fail to reach its Lower Release % Limit. (page 552)
	Note: Values in this array are the first letter of the rule. Example Use G for the element value if the rule is a Grouping rule.	

Parameter	Description		
Attribute Name Array Example (page 553)	<p>A text array point. Each element in this array contains the name of an attribute to be paired with an attribute value for evaluation by the respective business rule. For each Attribute Name element:</p> <ol style="list-style-type: none"> The function block attempts to match the respective Attribute Value element with a value for that attribute from the current item. For a matching Attribute Value, the function block compares the respective Count and Rule elements, or the respective release limits and release counts according to the respective Rule Type element (G, S, B or P). If selecting the current item (incrementing the Count) would violate the rule, the comparison fails for that attribute and the item is either weighted or eliminated. The failure is saved as history data. <p>If selecting the current item would not violate this rule, the function block attempts to match the next attribute name/value match.</p> <p>An elimination may be rolled back depending on the respective danger flag setting in the Breakable array. Items remain in the range source unless they are eliminated.</p> <p>To compare items that satisfy a named solve (query) or query expression, enter the reserved name SOLVE for this element in the Attribute Name array. (This is instead of the name of an attribute). You can then enter a named query or query expression as the respective element in the Attribute Value array.</p>		
Attribute Value Array Example (page 553)	<p>A text array point. Each element of this array contains a value to be paired with the respective element in the Attribute Name Array. An element of the array can contain:</p> <ul style="list-style-type: none"> The value of the respective attribute in the Attribute Name Array. Each element can contain one or multiple values with a comma separator. Do not include quotes. A named query, or query expression, returning a Boolean result. <p>Guidelines</p> <ul style="list-style-type: none"> For any element of the Attribute Name Array that contains the reserved name SOLVE you can enter a named query, or query expression, as the respective value element. In this case, the respective rule is not applied to an attribute name/value pair, but is instead applied to any items for which the query returns TRUE. Example A Boolean named query (page 918) <code>\$IsATruck\$</code> returns TRUE for items that have a heavy-duty chassis <code>\$IsATruck\$</code> is the query expression: <code>Vehicle.ChassisType='Heavyduty'</code> When <code>\$IsATruck\$</code> is entered as the value element, a spacing rule can be applied to determine the number of counts required between truck items in a resulting sequence. You can use a query that returns a set of attributes, instead of a Boolean result, by setting the query name or expression to a particular value. <p>Example: An attribute named query (page 918) <code>\$ChassisIs\$</code> is the query expression <code>Vehicle.ChassisType. \$ChassisIs\$</code> can be entered as the value element of this array in the form of an expression: <code>\$ChassisIs\$ = 'Heavyduty'</code></p>		
	<p>The parameters below define the business rules that evaluate an item with a matching attribute name/value pair.</p>		
Rule Array	<p>(Must be configured if a Grouping, Spacing or Blocking rule is used.) An integer (analog) array point. The Rule value has a different interpretation for each Rule Type.</p>		
	<table border="1"> <tr> <td data-bbox="423 1575 496 1625">Rule</td> <td data-bbox="496 1575 1425 1625">Dictates the:</td> </tr> </table>	Rule	Dictates the:
Rule	Dictates the:		
	<table border="1"> <tr> <td data-bbox="423 1625 496 1705">G</td> <td data-bbox="496 1625 1425 1705">Maximum count allowed of consecutively selected items that have the same specified attribute name/value.</td> </tr> </table>	G	Maximum count allowed of consecutively selected items that have the same specified attribute name/value.
G	Maximum count allowed of consecutively selected items that have the same specified attribute name/value.		
	<table border="1"> <tr> <td data-bbox="423 1705 496 1785">S</td> <td data-bbox="496 1705 1425 1785">Minimum item count required between items that have the same specified attribute name/value.</td> </tr> </table>	S	Minimum item count required between items that have the same specified attribute name/value.
S	Minimum item count required between items that have the same specified attribute name/value.		
	<table border="1"> <tr> <td data-bbox="423 1785 496 1860">B</td> <td data-bbox="496 1785 1425 1860">Minimum count requirement of consecutively selected items that have the same specified attribute name/value.</td> </tr> </table>	B	Minimum count requirement of consecutively selected items that have the same specified attribute name/value.
B	Minimum count requirement of consecutively selected items that have the same specified attribute name/value.		

Parameter	Description	
Example (page 553)	P	Ignored.
Count Array	(Must be configured if Grouping, Spacing or Blocking rules are used.) An array point configured as integer (analog). The Count value has a different interpretation for each Rule Type:	
	Rule	Each Element counts the number of selected items that have:
	G	The same respective attribute name/value that was consecutively selected immediately before the current item. The count is reset to zero as soon as an item with a different value for the respective Attribute Name is selected.
	S	A different value from the specified name/value. The count is reset to zero when the minimum count requirement (S in the Rule array) is reached and the item with the respective value can be selected again.
	B	The same respective attribute name/value that consecutively follow the current item. This count element is reset to the actual look-back count every pass or control cycle.
Example (page 553)	P	Ignored.
Upper Release % Limit	(Must be configured if Percentage rules are used.) A real or integer (analog) array point. This value is used for percentage rules only:	
	Rule	Each Element is
	G	Ignored.
	S	Ignored.
	B	Ignored.
Example (page 553)	P	The maximum percent of the Total Release Count that items with the respective attribute name/value are allowed reach.
Lower Release % Limit	(Must be configured if Percentage rules are used.) A real or integer (analog) array point. This value is used for percentage rules only:	
	Rule	Each Element is
	G	Ignored.
	S	Ignored.
	B	Ignored.
	P	The minimum percent of the Total Release Count that items with the respective attribute name/value must reach.
		<p>Example The current item is a red, two-door car.</p> <ol style="list-style-type: none"> The number of red cars selected has not yet reached the Upper Release % Limit defined for Color/Red. Selecting the red two-door would cause the percentage of four door cars selected to fall below its defined Lower Release % Limit The red, two-door car is eliminated instead of being selected.

Parameter	Description	
Example (page 553)		<p>Notes:</p> <ul style="list-style-type: none"> This evaluation checks the Lower Release % Limit for attributes of all items that remain in the range source for which a percentage rule has been defined. The Eliminate Sources Based on Rule function block is generally used within a multiple-pass Resequence Block Module where a rejected item may be selected by a subsequent pass.
Attribute Release Count Array	(Must be configured if Percentage rules are used.) An integer (analog) array point. Each element of the array contains the total number of selected items that have the same respective attribute name/value.	
Total Release Count point	(Must be configured if Percentage rules are used.) Integer (analog) point. Counter point stores and continues to increment the total number of items released.	
Breakable Array	A Boolean array point. Each element of the array contains a danger flag. Example	
	0 =	Nonbreakable
	1 =	Breakable
	The flag value 1 roll backs eliminations when the respective rule is broken.	
Example (page 553)	Note: The danger flag only has effect for a given rule when Always Evaluate is set to Breakable. A danger state can be set for any attribute name/value pair for which a rule is defined.	

Eliminate Sources Based on Rule Example

The table below shows the elements of the array points defined Eliminate Sources Based on Rule organized as a table.

Columns and rows represent the following.

Column	Elements of one of the array points
Row	Business rule for a particular attribute name/value pair (yellow-shaded elements).

0 = Ignored elements. In practice any value, or no value, can be assigned to an ignored array point element.

Rule Type (page 550)	Attribute Name (page 551)	Attribute Value (page 551)	Rule (page 551)	Upper Release % Limit (page 552)	Lower Release % Limit (page 552)	Count (page 552)	Attribute Release Count (page 553)	Total Release Count Point (page 553)	Breakable (page 553)
S (page 554)	Color	Red	3	0	0	2	32		0
G (page 554)	Color	Green	2	0	0	2	11		0

P (page 554)	Doors	Two	0	20	15	3	25	134	1
B	SOLVE	\$IsaTruck \$	4	0	0	5	17		0
P	Doors	Four	0	60	40	2	74		1

S. Array elements for the first row S (Spacing) rule keep count of the following:

1. At least three consecutive non-red items must be selected before another red item can be selected.
2. Two non-red items have been selected since the last red item was selected.
3. If the current item is

A.	Not red	The Spacing rule has not been broken. <ul style="list-style-type: none"> • The item will be selected. • The count will go to 3.
B.	Red	The Spacing rule has been broken. <ul style="list-style-type: none"> • The item will fail and will be either weighted or eliminated. • The count will stay at 2.

G. Array elements for the second row G (Grouping) rule provide the following:

4. Only two green items can be selected consecutively.
5. Two consecutive green items have been counted.
6. If the current item is

A.	Not green	The Spacing rule has not been broken. <ul style="list-style-type: none"> • The item will be selected. • The count will go to 0.
B.	Green	The Spacing rule has been broken. <ul style="list-style-type: none"> • The item will fail and will be either weighted or eliminated. • The count will stay at 2.

P. Array elements for the third row P (Percentage) rule provide the following:

7. The percent of selected two- door items must fall in the following range.

Maximum	20%
Minimum	15%

8. 25 2-door items have been selected.

9. The percent of the Total Release Count, 134, is **19%**.
10. The count is still within the bounds of the rules, but is close to breaking the maximum.

In this example, a warning alarm will be triggered to alert production.


Result based on the first three rules and values in this table:

- The next item can be neither red nor green.
- No more than 2 2-door cars can be selected before 4-door cars are selected.
- The additional rules provide further filters for selecting items.

Eliminate Source By Percentages

Text array point. Each rule contains an attribute name/value that the function block compares to an item's corresponding attribute names/values. **Note:** The name is defined in the Attribute Name Array. An element in the array can contain:

- The value of the respective attribute in the Attribute Name Array.
- Each element can contain one or multiple values with a comma separator. Do not include quotes.
- Attribute values can be [wild cards \(page 532\)](#).
- A named query, or query expression, returning a Boolean result.

 **guide: Guidelines**

- For any element in the Attribute Name Array that contains the reserved name SOLVE you can enter a named query, or query expression, as the respective value element.

In this case, the percentage rule is not applied to an attribute name/value, but is instead applied to any items for which the query returns TRUE. Example A Boolean [named query \(page 918\)](#) `$IsStandardHatchback$` returns TRUE for items that have a standard transmission and have a hatchback rear door. `$IsStandardHatchback$` is the query expression: `Vehicle.Transmission='Standard'&Vehicle.DoorType='Hatchback'` When `$IsStandardHatchback$` is entered as the value element, the applied percentage rule determines the percent of selected vehicles that can be standard hatchbacks. However, interaction with other rules can cause a hatchback item to fail even if it complies with the hatchback percent rule. Example

1. The current hatchback item is red.
2. Selecting the red hatchback causes the percentage rule defined for a different attribute (or query) e.g. COLOR Blue, to fail to meet its Lower Release % Target,
3. The current red hatchback item is eliminated.
 - You can use a query that returns a set of attributes, instead of a Boolean result, by setting the query name or expression to a particular value.

ExampleA attribute named query `$TransmissionIs$` is the expression `Vehicle.TransmissionType. $TransmissionIs$` can be entered as the value element of this array in the form of an expression: `$TransmissionIs$ = 'Standard'`

4. The number of red cars selected has not yet reached the Upper Release % Limit defined for Color/Red.
5. Selecting the red two-door would cause the percentage of four door cars selected to fall below its defined Lower Release % Limit
6. The red, two-door car is eliminated instead of being selected.

Notes:

- This evaluation checks the Lower Release % Limit for attributes of all items that remain in the range source for which a percentage rule has been defined.
- Eliminate Sources Percentages function block is generally used within a multiple-pass Resequencing Block Module. Therefore, a rejected item may be selected by a subsequent pass in that case.

Eliminate Source By Ratio

(Available with Order Execution Mgt.)

Description

Eliminate Source By Ratio eliminates or applies a weight to items that do not fit ratio constraints.

The applied weight is a value that:

- You entered in the [Set Block Weights \(page 575\)](#) function block's corresponding parameter.
- Set Block Weights assigns to this function block.

Any item whose selection would push any attribute name/value ratio outside its configured limits fails.

Notes

- An accurate historical record of the item selections is required.

When you use this block, insert a corresponding Record Attribute function block into your routing or output logic module.

- Insert a Record Attribute function block where all item eliminations (by any and all methods) have been completed.

Example

If the Eliminate Source function block is placed within a Resequence Block Module, then the corresponding Record Attribute function block should be inserted immediately before the [Resequence Range \(page 572\)](#) block where the selection is processed.

- The `Eliminate Source By Ratio` and corresponding Record Attribute function blocks should be configured with the same points and array point names.
- A block that applies a zero weight eliminates failed items from the range source.
- Rotation numbers that observe a particular ratio can be assigned to items.

Use Eliminate Source by Ratio within a multiple-pass Resequence Block Module.

Parameters

This function block has the following parameters:

Parameter	Description		
Always Evaluate	Hard, Soft or Breakable. Controls what happens when the extension eliminates all items in the range source.		
	Hard	All items will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.	
	Soft	Roll back any eliminations made by this extension.	
	Breakable	Roll back any eliminations made by this extension that have a danger point value of TRUE (non zero). Routing modules will first use the danger defined for each individual decision However, one of the following may occur instead.	
		If:	Then:
		<ul style="list-style-type: none"> • No danger point has been configured for the decision. 	The site-wide danger point is used.
		<ul style="list-style-type: none"> • This function block is used in the decision output logic. 	The site-wide danger point is used.
		<ul style="list-style-type: none"> • No danger point can be found for a decision. 	The decision remains eliminated.
Attribute Name Array	Text array point. Each element of this array contains the name of an attribute to be paired with an attribute value for comparison. The function block attempts to match the array element's Attribute Value the current item's value for the same attribute. Results can be either of the following.		
		If:	Then:
		<ul style="list-style-type: none"> • The attribute values match. 	The function block compares the respective Block Size and Attribute History elements.


	<ul style="list-style-type: none"> • Selecting the current item (incrementing the Spacing Count) would exceed the ratio defined by <code>Block Size:Block Limit</code>. 	<p>The comparison fails for that attribute and the:</p> <ul style="list-style-type: none"> • Item is either weighted or eliminated. • Failure is saved as history data.
	<p>Note: To compare items that satisfy a named query or query expression, enter the reserved name SOLVE for this element in the Attribute Name array. (This is instead of the name of an attribute). You can then enter a named query or query expression as the respective element in the Attribute Value array.</p>	
Block Size Array	<p>Integer (analog) array point. Each element in the array contains the size of the block for its respective attribute name/value, i.e. the denominator of the ratio. The number of items released that have the attribute are recorded to the elements of this array.</p>	
Block Limit Array	<p>Real or integer (analog) array point. Each element of the array contains the maximum number of matching attribute values for the respective attribute name/value in a block; i.e. the numerator of the ratio.</p>	
Attribute History Array	<p>Integer (analog) array point. Each element of the array contains the actual history of released items for the respective attribute name/value.</p>	

End Spread Range Block Module

(Available with Order Execution Mgt.)

Description

End Spread Range Block Module marks the end of a multiple-pass Spread Range Block Module that starts with a [Begin Spread Range Block Module \(page 538\)](#).

 **Note:** An End Spread Range Block Module block is required when using a Spread Range Block Module in the routing or output logic module.

Parameters

This function block has no parameters.

Move Item From Range

(Available with Order Execution Mgt.)

Description

Move Item From Range moves a PRT item from a range to a region, possibly on a different project.

Parameters

This function block has the following parameters:

Parameter	Description

Item Id Variable	Item ID. Do one of the following. <ul style="list-style-type: none"> • Enter an Item ID. • Select an RCO variable that was previously set to an Item ID value.
Source Range Name	(Optional) Range from which the item will be moved.
Destination Project	(Optional) Project that has the destination region.
Dest. Region Id	Region to which the item will be moved.
Dest. Region Location	Location in the region to which the item will be moved.
Insert Flag	How the item should be added to the destination region location. Options are: <ul style="list-style-type: none"> • ADD • INSERT

Range Source Eliminate By Expression Mismatch

(Available with Order Execution Mgt.)

Description

Range Source Eliminate By Expression Mismatch evaluates values in expressions.

Any value that does not adhere to a stated value is considered a "mismatch" and is eliminated.

Example

An expression that is evaluated is:

```
'color' = ""BLUE""
```

Any item that is not BLUE is eliminated.

Important:

This block is designed to work only when a valid danger point is configured.

A danger point:

- Must be either of two types:
 - Integer, long.
 - Boolean.

All other types throw an exception.

- Can be device or virtual point.

- Can have a point value of zero or non zero.

The danger point value is considered in elimination decision, when the value is non-zero and if the evaluation type is Breakable (neither Soft nor Hard) type.

If an invalid danger point or no danger point is configured, the function block will not work and throws CORLOG warning.

Parameters

This function block has the following parameters:

Parameter	Description		
Always Evaluate	Hard, Soft or Breakable. Controls what happens when the extension eliminates all items in the range source.		
	Hard	All items will remain eliminated. Note: When all decisions are eliminated the RCO Output Module, <code>None</code> , is executed to handle this condition.	
	Soft	Roll back any eliminations made by this extension.	
	Breakable	Roll back any eliminations made by this extension that have a danger point value of TRUE (non zero). Routing modules first use the danger defined for each individual decision. However, one of the following may occur instead.	
		If:	Then:
		<ul style="list-style-type: none"> • No danger point has been configured for the decision. 	The site-wide danger point will be used.
		<ul style="list-style-type: none"> • This function block is used in the decision output logic, 	The site-wide danger point will be used.
		<ul style="list-style-type: none"> • No danger point can be found for a decision. 	The decision remains eliminated.
Expression	Attribute expression that sources will be evaluated against.		
	Operators		
	Basic	>, >=, <, <=, = and !=	
		Where	
		>	Greater than
		>=	Greater or equal

		<	Less than
		<=	Less or equal
		=	Equals
		!=	Not
		<>	Not (greater than or less than)
	Modifiers	#, %	
		Where	
		#	Integer
		%	Floating (decimal)
	Boolean	and, or, (combine sub-conditions)	
	Exists value	= "\$Exists"	Attribute exists
		<> "\$Exists"	Attribute does not exist
	Note: This evaluation is similar to the SQL Where condition.		
	Examples		
	'var' = "value"		
	('var' = "value") AND ('var < "value")		
	'COLOR' = \$EXISTS		
	Syntax Notes		
	<ul style="list-style-type: none"> • A variable is surrounded by single-quotes. • A value is surrounded by double-quotes. • At least one space is required around an operator. • Parentheses can be placed around sub-expressions to include more than one sub-expression in the expression. • An integer value must be started with a pound sign #. • A float must be started with a % sign %. 		
	Note: Case is not important; any entry will be treated as upper case.		
Expression Point ID	(Optional) point containing a dynamically changeable expression. Supercedes Expression.		

Record Attribute Pattern

(Available with Order Execution Mgt.)

Description

Record Attribute Pattern adds history data for the currently selected source item to pattern the history array points that you create or select.

 **Note:**

- Add Record Attribute Pattern to the end of any module where you use the [Eliminate Source by Pattern \(page 547\)](#) function block to store information about the item that remains after that elimination.

Example

Eliminate Source by Pattern is in a Resequencing Block Module,

Place a Record Attribute Pattern block between that block and the [Resequencing Range \(page 572\)](#) block at the end of that module.

- Record Attribute Pattern records only one item's history data to the specified points the item at the range source head.
- Where all item eliminations (by any and all methods) have been completed.

1. Insert another function block such as

	Select Lowest Weight
	Select Source By Age


before Record Attribute Pattern to insure that the most appropriate item(s) are selected.

If you have criteria to determine which item should be selected in case of a tie, specify that criteria in an RSA function block inserted before the Select block.

2. Insert Record Attribute Pattern.

Parameters

This function block has the following parameters:

 **Note:** Select the same array points previously configured for the corresponding Eliminate Source by Pattern block you inserted earlier in this logic or output module.

Parameter	Description
Attribute Name Array	Text array point. The names of the selected item's attributes are recorded to elements of this array.
Attribute Value Array	Text array point. The values of the selected item's respective attributes are recorded to elements of this array.

Pattern History Point	Integer (analog) point. The actual history of the items selected (one per pass or control cycle) are recorded to this point.
Attribute Name History Array	Text array point. The history attribute names are recorded to the elements of this array.
Attribute Value History Array	Text array point. The history attribute values are recorded to the elements of this array.

Record Attribute Percentages

(Available with Order Execution Mgt.)

Description

Record Attribute Percentages adds history data for the currently selected source item to percentage history array points that you create or select.

Note:

- Add Record Attribute Percentages to the end of any module where you use the [Eliminate Source by Percentages \(page 555\)](#) function block to store information about the item that remains after that elimination.

Example

Eliminate Source by Percentages is in a Resequence Block Module,

Place a Record Attribute Percentages block between that block and the [Resequence Range \(page 572\)](#) block at the end of that module.

- Record Attribute Percentages records only one item's history data to the specified points the item at the range source head.
- Where all item eliminations (by any and all methods) have been completed.

1. Insert another function block such as

- Select Lowest Weight
- Select Source By Age


before Record Attribute Percentages to insure that the most appropriate item(s) are selected.

If you have criteria to determine which item should be selected in case of a tie, specify that criteria in an RSA function block inserted before the Select block.

2. Insert Record Attribute Percentages.

Parameters

This function block has the following parameters:

 **Note:** Select the same array points previously configured for the corresponding Eliminate Source by Percentage block you inserted earlier in this logic or output module.

Parameter	Description
Attribute Name Array	Text array point. The names of the selected item's attributes are recorded to elements of this array.
Attribute Value Array	Text array point. The values of the selected item's respective attributes are recorded to elements of this array.
Attribute Release Array	Integer (analog) array point. The actual number of items released that have the respective attribute are recorded to the elements of this array.
Total Release Count	Integer (analog) point. Counter point stores and continues to increment the total number of items released.

Record Attribute Ratios

(Available with Order Execution Mgt.)

Description

Record Attribute Ratios adds history data for the currently selected source item to ratio history array points you create or select.

Note:

- Add Record Attribute Ratios to the end of any module where you use the [Eliminate Source by Ratio \(page 556\)](#) function block to store information about the item that remains after that elimination.

Example

Eliminate Source by Ratio is in a Resequence Block Module,

Place a Record Attribute Ratios block between that block and the [Resequence Range \(page 572\)](#) block at the end of that module.

- Record Attribute Ratio records only one item's history data to the specified points the item at the range source head.
- Where all item eliminations (by any and all methods) have been completed.

1. Insert another function block such as

- Select Lowest Weight
- Select Source By Age


before Record Attribute Ratio to insure that the most appropriate item(s) are selected.

If you have criteria to determine which item should be selected in case of a tie, specify that criteria in an RSA function block inserted before the Select block.

2. Insert Record Attribute Ratio.

Parameters

This function block has the following parameters:

 **Note:** Select the same array points previously configured for the corresponding Eliminate Source by Ratio block you inserted earlier in this logic or output module.

Parameter	Description
Attribute Name Array	Text array point. The names of the selected item's attributes are recorded to elements of this array.
Attribute Value Array	Text array point. The values of the selected item's respective attributes are recorded to elements of this array.
Block Size Array	Integer (analog) array point. The current number of items selected in a block are recorded to the elements of this array.
Attribute History Array	Integer (analog) array point. The current number of items selected for each attribute name/value pair are recorded to this point.

Record Attribute Rule

(Available with Order Execution Mgt.)

Description

Record Attribute Rule adds the history data saved for the currently selected source item to Rule History array points you create or select.

Note:

- Add Record Attribute Rule to the end of any module where you use the [Eliminate Sources Based on Rule \(page 549\)](#) function block to store information about the item that remains after that elimination.

Example

Eliminate Source by Rule is in a Resequencing Block Module,

Place a Record Attribute Rule block between that block and the [Resequencing Range \(page 572\)](#) block at the end of that module.

- Record Attribute Rule records only one item's history data to the specified points the item at the range source head.
- Where all item eliminations (by any and all methods) have been completed.

1. Insert another function block such as

- Select Lowest Weight
- Select Source By Age


before Record Attribute Rule to insure that the most appropriate item(s) are selected.

If you have criteria to determine which item should be selected in case of a tie, specify that criteria in an RSA function block inserted before the Select block.

2. Insert Record Attribute Rule.

Parameters

This function block has the following parameters:

 **Note:** Select the same array points previously configured for the corresponding Eliminate Sources Based on Rule block you inserted earlier in this logic or output module.

Parameter	Description
Rule Type Array	Text array point. Each element of the array contains one of the following rule types.
	G Grouping
	S Spacing
	B Blocking
	P Percentage
Attribute Name Array	Text array point. The names of the selected item's attributes are recorded to elements of this array.
Attribute Value Array	Text array point. The values of the selected item's respective attributes are recorded to elements of this array.
Count Array (page 552)	(Grouping, Spacing and Blocking rules only) Integer (analog) array point. The actual number of items consecutively selected that have the same, or different, attribute name/value (depending on the grouping, spacing or blocking rule) is recorded to the respective element of this array. Note: Items selected using a percentage rule are also recorded, however this data is not currently used.
Attribute Release Count Array (page 553)	(Grouping, Spacing and Blocking rules only) Integer (analog) array point. The actual number of matching items selected for the respective attribute name/value pair is recorded to the respective element of this array.

Total Release Count	(Percentage rules only) Integer (analog) point. Counter point stores and continues to increment the total number of items released.
---------------------	---

Record Attribute Spacing

(Available with Order Execution Mgt.)

Description

Record Attribute Spacing adds history data for the currently selected source item to spacing history array points you create or select.

Add this block to the end of any module where you use the [Eliminate Sources Based on Attribute Spacing \(page 548\)](#) function

Note:

- Add Record Attribute Spacing to the end of any module where you use the [Eliminate Sources Based on Attribute Spacing \(page 548\)](#) function block to store information about the item that remains after that elimination.

Example

Eliminate Source by Spacing is in a Resequence Block Module,

Place a Record Attribute Spacing block between that block and the [Resequence Range \(page 572\)](#) block at the end of that module.

- Record Attribute Spacing records only one item's history data to the specified points the item at the range source head.
- Where all item eliminations (by any and all methods) have been completed.

1. Insert another function block such as

Select Lowest Weight
Select Source By Age


before Record Attribute Spacing to insure that the most appropriate item(s) are selected.

If you have criteria to determine which item should be selected in case of a tie, specify that criteria in an RSA function block inserted before the Select block.

2. Insert Record Attribute Spacing.

Parameters

This function block has the following parameters:

 **Note:** Select the same array points previously configured for the corresponding Eliminate Sources block you inserted earlier in this logic or output module.

Parameter	Description
Attribute Name Array	Text array point. The names of the selected item's attributes are recorded to elements of this array.
Attribute Value Array	Text array point. The values of the selected item's respective attributes are recorded to elements of this array.
Spacing Count Array	Integer (analog) array point. The current spacing counts for the respective attributes are recorded to the elements of this array.

Record Violated Rules

(Available with Order Execution Mgt.)

Description

Record Violated Rules commits broken rule records to the currently selected items' extended attributes.

Note:

Where all item eliminations (by any and all methods) have been completed.

1. Insert another function block such as

- Select Lowest Weight
- Select Source By Age

before Record Violated Rules to insure that the most appropriate item(s) are selected.

If you have criteria to determine which item should be selected in case of a tie, specify that criteria in an RSA function block inserted before the Select block.

2. Insert Record Violated Rules.

Parameters


This function block has no parameters.

Release Hold For Item

(Available with Order Execution Mgt.)

Description

Release Hold From Item removes a named hold on the item at a particular location in a specified region.

 **Note:** Only items local to the current project are modified.

Apply this function block across multiple projects as follows.

1. Enter a comma-delimited list of project names into the **Project ID** field.
2. Recompile the block.

Parameters

This function block has the following parameters:

Parameter	Description
Region Name	Name of the Region where the item is located.
Region Location	Location of the item in the Region. Select a region or enter a value greater than zero that specifies a region.
Item Class	Class of the item whose hold will be released.
Hold Attribute Name	Name of the attribute that determines the hold status.
Hold Attribute Point ID	(Optional) Point ID that provides the value for the hold attribute. Supercedes Hold Attribute Name.
Project ID	(Optional) Names of one or more projects in which the selected region may be found. Note: Leave this field blank to limit the search for regions to the local project only.
Project Point ID	(Optional) Text point. Stores a list of project(s). Supercedes Project ID.

Release Hold for Range

(Available with Order Execution Mgt.)

Description

Release Hold for Range removes a named hold on all items that remain in the range source after any possible eliminations.

Parameters

This function block has the following parameters:

Parameter	Description
Hold Attribute Name	Name of the hold attribute that determines the release status.
Hold Attribute Point ID	(Optional) Point ID that provides the value for the hold attribute. Supercedes Hold Attribute Name.

Reorder Region by Rotation Number

(Available with Order Execution Mgt.)

Description

Reorder Region by Rotation Number reassigns location numbers of all items in the specified region to match the numerical sequence of their rotation numbers.

Note:

- Items that have no attribute by the name specified, or that have a null or non-numeric value for the rotation, will:
- Be moved to the end of the sequence, i.e assigned the highest rotation and moved to the end of the region
- Retain their static order with respect to each other.
- Reorder Region by Rotation Number is intended for reordering non physical items, such as Order items. Use [Advance in Order \(page 534\)](#) to change the rotation numbers of physical items.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	Region where you want to reassign location numbers.
Item Class	Class of the item assigned a rotation number.
Rotation Number Attribute	Name of the attribute that stores the rotation number.
Region Point ID	(Optional) Text point. Contains the name of a region. Supercedes Region ID.
Rotation Attribute Point ID	(Optional) Text point. Stores the name of an attribute. Supercedes Rotation Number Attribute.

Resequence Mode Begin

(Available with Order Execution Mgt.)

Description

Resequence Mode Begin marks the start of a multiple-pass Resequence Block Module that assigns rotation numbers to items in the range source.


Note:

A [Resequence Range \(page 572\)](#) block is required to mark the end of the Resequence Block Module.

- The sequence occurs as follows.
 1. Resequence Mode Begin starts the logic.
 2. Other RSA function blocks, inserted between Resequence Mode Begin and Resequence Range, perform an evaluation on items and temporarily weight or eliminate the items for the duration of the current pass.
 3. Resequence Range then selects the item with the lowest weight and assigns it the next rotation number.
 4. The next pass evaluates the remaining unselected items, including any items eliminated by previous passes.
 5. After the final pass, all items that have been assigned a rotation number become available to the subsequent function blocks in your routing or output logic module.
 - A block that applies a zero weight directly eliminates items that fail from the range source.
 - A block that applies a non-zero weight only weights a particular item once (per pass).

For example, an item that fails on several accounts, because more than one attribute name/value pair fail the comparison, is not weighted more heavily by a block than an item that fails on just one account.

- Weights are not accumulated by items from pass to pass. Each pass starts fresh with unweighted items in the range source.

 **Important:** Use only RSA blocks between Resequence Mode Begin and Resequence Range.

Parameters

This function block has no parameters.

Resequence Range


(Available with Order Execution Mgt.)

Description

Resequence Range marks the end of a multiple-pass Resequence Block Module and defines the way rotation numbers are assigned. After each pass, Resequence Range selects the item with the lowest weight and assigns it the next rotation number.

The block module's internal logic performs multiple passes of remaining items until:

- All items in the range source have been assigned a rotation number, or
- No items have been selected (a pass has eliminated all items not yet assigned).

 **Note:** If a pass eliminates all of the remaining items, the resequencing logic concludes. A rotation number will not be assigned to those unselected items. As a precaution, insert an [Alarm on No Sources Remaining \(page 535\)](#) block immediately before the Resequence Range block.

After the final pass, all items that have been assigned a rotation number are available to the subsequent function blocks in your routing or output logic module.

Parameters

This function block has the following parameters:

Parameter	Description
Attribute	Name of the standard attribute that holds the item's rotation number. Tip: Create a standard attribute, for example, SEQNUM in the Attribute Maintenance (page 350) box.
Rotation Number Base Value	First rotation number, where resequencing begins. Do one of the following. <ul style="list-style-type: none"> • Enter an integer value. • Select an RCO Variable assigned this value.
Rotation Number Increment Value	Numeric value to add to each successive rotation number. Do one of the following. <ul style="list-style-type: none"> • Enter an integer value. • Select an RCO Variable assigned this value.
Rotation Number Attribute Point ID	(Optional) Text point. Stores the attribute name. Supercedes Attribute.

Rotation Pull Ahead


1. Double-click this field to display the browse dialog for ranges.
2. Do one of the following:
 - Enter a existing range or region name
 - Click the Browse button to select from a list of existing ranges.
 - Click the Pop-up button and:
 - Select Browse to select from a list of existing ranges, or
 - Select New to define a new range. The Range Viewer dialog box appears.


Select Lowest Weight

(Available with Order Execution Mgt.)

Description

Select Lowest Weight eliminates all items within the source except for the item with the lowest weight.

 **Note:** In the case of a tie, i.e. if more than one item has the lowest weight, only one item is selected.

 **Note:** If you have criteria to determine which item should be selected in case of a tie, specify that criteria in an RSA function block inserted before the Select Lowest Weight block.

Parameters

This function block has no parameters:

Select Source By Age

ID of the attribute to base the comparison on. **Note:** The selection order is:

1. Items that have a value for this attribute (either the oldest or youngest, depending on the Age parameter).
2. Items with no value for this attribute.
3. Items that do not have this attribute at all.

Set Attribute (Range Source)

(Available with Order Execution Mgt.)

Description

(Range Source) Set Attribute sets the value of a specified attribute on the currently selected item in the Range Source list.

Note: In Range Source, the implicit selection is the first item in the Range Source list.

Parameters

Parameter	Description										
Attribute ID	(Required) Target attribute ID. Listed attributes are defined in the Attribute Maintenance dialog box. Tip: Click Tools>Attribute Maintenance on the Tracker Configuration window menu bar to open the Attribute Maintenance dialog box.										
From	(Required) Specifies the supplier of the value. Selections are:										
	<table border="1"> <tr> <td>Selection</td> <td>Parameter entry required with selection</td> </tr> <tr> <td>Attribute</td> <td>AttributeID (page 575)</td> </tr> <tr> <td>Expression</td> <td>Value Expression (page 575)</td> </tr> <tr> <td>Point</td> <td>Point Name (page 574)</td> </tr> <tr> <td>Value</td> <td>Attribute Value (page 574)</td> </tr> </table>	Selection	Parameter entry required with selection	Attribute	AttributeID (page 575)	Expression	Value Expression (page 575)	Point	Point Name (page 574)	Value	Attribute Value (page 574)
Selection	Parameter entry required with selection										
Attribute	AttributeID (page 575)										
Expression	Value Expression (page 575)										
Point	Point Name (page 574)										
Value	Attribute Value (page 574)										
Point Name	(Required if the From parameter is set to Point) Any single point ID. The point's value is assigned to the attribute.										
Attribute Value	(Required if the From parameter is set to Value) Absolute value that will be assigned to the target attribute. The value can contain any character; all values are passed as characters, including number characters.										
Project List	(Optional and used only when the From parameter is set to Attribute) One or more projects separated by commas. Note: If the Project List parameter is used the function block will only look in those projects for the value source.										
RegionID	(Optional and used only when the From parameter is set to Attribute) One or more regions separated by commas. Note: If the RegionID parameter is used, the function block will only look in those regions for the value source.										
Region Location	<p>(Optional and used only when the From parameter is set to Attribute) A region location can be any of the following:</p> <ul style="list-style-type: none"> • First. • Last. • All. • A specific location ID. <p>Note: If the Region Location parameter is used, Range Source will only look in the specified location(s) for the value source. The location cannot be a range.</p>										
ItemClass	(Optional and used only when the From parameter is set to Attribute) Class of the items in the source list.										


AttributeID	(Required if the From parameter is set to Attribute) ID of the From (page 574) attribute whose value will be assigned to the target attribute. Listed attributes are defined in the Attribute Maintenance dialog box. Tip: Click Tools>Attribute Maintenance on the Tracker Configuration window menu bar to open the Attribute Maintenance dialog box.
Item Id	(Optional and used only when the From parameter is set to Attribute) Item from which the attribute value will be copied.
Value Expression	(Required if the From parameter is set to Expression) Any CIMPLICITY Basic expression will be calculated; the value will be assigned to the target attribute.

Set Block Weights

(Available with Order Execution Mgt.)

Description

Set Block Weights assigns a weight of the subsequent RSA function blocks that you add to an output or logic module.

 **Note:** Whenever a range is used the logic module must begin with two blocks in the following order.

Block	Module Options	Purpose
1	Can be either: <code>Set Source Range</code> or <code>Set Source Solve</code>	Defines the range that the block evaluates.
2	Must be <code>Set Block Weights</code>	Sets the static importance of each of the subsequent blocks by assigning a weight to each block.

 **Note:**

- The **zero** or **non-zero** weight block does the following.

Weight	The block:
Zero	Directly eliminates items that do not satisfy its criteria.
	If a zero weighted block eliminates all items in range, that block's eliminations may be rolled back depending on the block's Always Evaluate parameter setting.
Non-zero	Applies this weight to any item that fails to satisfy its criteria; it does not eliminate items directly.
	Other subsequent blocks, e.g. Eliminate by Weigh (page 478) , can be added to eliminate items on the basis of their accumulated weight. Note: CIMPLICITY also creates a broken rule record for failed items.

- After execution of the output logic is complete items do not retain their weight that is assigned by the RSA function blocks

Parameters

This function block has the following parameters:

Parameter	Description	
Block Weights	Block 1 Weight	Weight for the first RSA block after the Set Block Weights block.
	Block 2 Weight	Weight for the second RSA block.
	...	
	Block 10 Weight	Weight for the tenth RSA block.
	<ul style="list-style-type: none"> • When items are evaluated, the higher the weight, the more likely they are to fail. • Use a numbering system that is least likely to end in a tie, for example, prime numbers. 	
Weight Point ID	(Optional) Integer array point with up to 100 elements. Each element defines the weight of the respective RSA block. Supercedes the other parameters.	

Set Point With Attribute

(Available with Order Execution Mgt.)

Description

Range Source Set Point With Attribute writes an attribute value of the current item to a CIMPPLICITY point.

Note:

- Range Source Set Point With Attribute writes only one item's attribute value to the specified point the item at the range source head.
- Where all item eliminations (by any and all methods) have been completed.

1. Insert another function block such as

- Select Lowest Weight
- Select Source By Age

before Range Source Set Point With Attribute to insure that the most appropriate item(s) are selected.

If you have criteria to determine which item should be selected in case of a tie, specify that criteria in an RSA function block inserted before the Select block.

2. Insert Range Source Set Point With Attribute.

Parameters

This function block has the following parameters:

Parameter	Description
Point Name	ID of the point to be updated with the attribute value.
Attribute	ID of the attribute whose value the point is set to. The name of the attribute to copy.


Set Rotation Number to Attribute

(Available with Order Execution Mgt.)

Description

Set Rotation Number to Attribute assigns a rotation number to the item at a particular location in a specified region, based on either the:

- Numeric value or
- Value assigned RCO variable that you specify.

 **Note:** Only items local to the current project are modified. Apply this function block across multiple projects:

1. Enter a comma-delimited list of project names in the **Project ID** field
2. Recompile the function block.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	ID of Region where the item is located.
Region Location	Location of the item in the Region. Note: The reserved names PRT_FIRST and PRT_LAST can be entered here to specify the first or last location, respectively, within the region.
Item Class	Class of the item to be assigned a rotation number.
Rotation Number Attribute	Name of the attribute to store the rotation number.
Variable	Value to assign as the rotation number specified as follows. <ul style="list-style-type: none"> • Enter a numerical value or • Select an assigned RCO Variable.
Rotation Number Attribute Point ID	(Optional) Text point. Stores the attribute name. Supercedes Attribute.

Project ID	(Optional) Names of one or more projects in which the selected region may be found. Note: Leave this field blank to limit the search for regions to the local project only.
Project Point ID	(Optional) Text point. Stores a list of project(s). Supercedes Project ID.

Set Source Criteria Set

1. Double-click **Criteria Set Name** to display the P1 dialog box for criteria sets.
2. Enter an existing criteria set name, or
 - Click the Browse button to select from a list of existing criteria sets, or
 - Click the Pop-up button and select:
 - Browse to select from a list of existing criteria sets, or
 - New to define a new criteria set. The Criteria Set Editor dialog box appears.

Set Source Range

Name of a range of sources.

1. Double-click **Range Name** to display the browse dialog for ranges.
2. Do one of the following.
 - Enter an existing range or region name.
 - Click the Browse button to select from a list of existing ranges.
 - Click the Pop-up button and:
 - Click Browse to select from a list of existing ranges, or
 - Click New to define a new range. The Range Viewer dialog box appears.

Set Source Solve

(Available with Order Execution Mgt.)

Description

Set Source Solve uses a named query to define a range of sources used by subsequent RSA function blocks in the output or logic module.

Note:

- Whenever a range is used the logic module must begin with two blocks in the following order.

Block	Module Options	Purpose
1	Can be either:	
	Set Source Range	Defines the range that the block evaluates.

	Set Source Solve	
2	Must be:	
	Set Block Weights	Sets the static importance of each of the subsequent blocks by assigning a weight to each block.

- Before you use Set Source Solve in a multi-project environment:

1. Configure a text point named TRACKER_PROJECTS in the current project.
2. Set the point to a comma-delimited list of projects names where the query will search for items.

Parameters

This function block has the following parameters:

Parameter	Description
Solve Name	Name of a query expression that returns sources. Enter the name of a defined query. <ol style="list-style-type: none"> Double-click Solve Name to open the Expression Browser. Do one of the following. <ul style="list-style-type: none"> • Enter an existing query name • Click the Browse button to display the Expression Browser (page 894). • Click the Pop-up Menu button and: <ul style="list-style-type: none"> • Select Browse to display the Expression Browser. (page 894) or • Select New to define a new query expression. The New Expression dialog box appears.
Item Type	TADB Item Type ID to which the query applies.
Solve Point ID	(Optional) Text point Stores the name of a query. Supercedes Solve Name.

3. Double-click **Solve Name** to open the Expression Browser.
4. Do one of the following.
 - Enter an existing query name
 - Click the Browse button to display the [Expression Browser \(page 894\)](#).
 - Click the Pop-up Menu button and:
 - Select Browse to display the [Expression Browser. \(page 894\)](#) or
 - Select New to define a new query expression. The New Expression dialog box appears.

Set Super Priority Weight

(Available with Order Execution Mgt.)

Description

Set Super Priority Weight overrides the weight value assigned to a block by a [Set Block Weights \(page 575\)](#) block that is the second block in the output or logic module.

 **Note:**

The **zero** or **non-zero** weight block does the following.

Parameters

Weight	The block:
Zero	Directly eliminates items that do not satisfy its criteria.
	If a zero weighted block eliminates all items in range, that block's eliminations may be rolled back depending on the block's Always Evaluate parameter setting.
Non-zero	Applies this weight to any item that fails to satisfy its criteria; it does not eliminate items directly.
	Note: CIMPPLICITY also creates a broken rule record for failed items.

This function block has the following parameters:

Parameter	Description
Block Number	Function block to which you want to apply a weight based on a point value.
Point ID	Point used for the super priority weight value. The selected point must be configured as an analog (numeric) data type. The value of this point overrides the weight value assigned to this block by a previous Set Block Weights block.

Set Variable With Attribute

(Available with Order Execution Mgt.)

Description

Range Source Set Variable With Attribute sets an RCO variable with an attribute value from the current item.

 **Note:**

- Range Source Set Variable With Attribute writes only one item's attribute value to the specified variable the item at the range source head.
- Where all item eliminations (by any and all methods) have been completed.

1. Insert another function block such as

- Select Lowest Weight
- Select Source By Age

before Range Source Set Variable With Attribute to insure that the most appropriate item(s) are selected.

If you have criteria to determine which item should be selected in case of a tie, specify that criteria in an RSA function block inserted before the Select block.

2. Insert Range Source Set Variable With Attribute.

Parameters

This function block has the following parameters:


Parameter	Description
Variable Id	The RCO Variable to be updated.
Attribute	ID of the attribute whose value the RCO Variable is set to.

Substitute Attributes and Status

(Available with Order Execution Mgt.)

Description

Substitute Attributes and Status exchanges all attributes and status bits of the current product item, including its sales order association (its [ASSOCIATE \(page 136\)](#) extended attribute value), with the attributes and status bits of a product item in a specified region/location.

 **Note:** For substitution to work, the selected item must have an extended attribute named ASSOCIATE that identifies the Item Type of its associated item.

Example

A PRT model is set up to track a BODY Item Type assembly.

- The BODY Item Type has an extended attribute ASSOCIATE.
- The ASSOCIATE extended attribute value is, literally, the value of another Item Type, ORDER.
- ORDER may be defined in the same, or another model.

ASSOCIATE attribute values are as follows.

ASSOCIATE attribute of the:	Is the Value Assigned to the:
Currently selected body item	ORDER item's attribute in the specified region\location.
ORDER item in the specified region\location	Currently selected body.

All other attributes and status bits are similarly swapped between the currently selected body and the body at the specified region location.

While the example above shows substitution of orders for different bodies, it is also possible to substitute bodies for different orders. You can only substitute the Item Type that is associated with the Item Type you have currently selected.

 **Note:**

- Both sales orders and products are items.
- You can use this function block if:
- Product items exist at some region defined in the current project's RCO database and
- Order items exist in either the current project or another running project to which you have access.
- You can search for the specified region across multiple projects. Enter a comma-delimited list of project names where the region may be found into the **Project ID** field and recompile.
- Where all item eliminations (by any and all methods) have been completed.


1. Insert another function block such as

Select Lowest Weight
Select Source By Age

before Substitute Attributes and Status to insure that the most appropriate item(s) are selected.

If you have criteria to determine which item should be selected in case of a tie, specify that criteria in an RSA function block inserted before the Select block.

2. Insert Substitute Attributes and Status.

 **CAUTION:** Attributes and status bits can be used for a wide variety of purposes. Use this function block with care and only after investigating the application of all attributes and status bits that belong to current item and its associate item.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	ID of Region where the item is located.
Region Location	Location of the item in the Region. Note: The reserved names PRT_FIRST and PRT_LAST can be entered in Region Location to specify the first or last location, respectively, within the region.
Item Class	Class of the item whose sales order will be exchanged.
Project ID	(Optional) Names of one or more projects in which the selected region may be found. Note: Leave this field blank to limit the search for regions to the local project only.

Project Point ID	(Optional) Text point. Stores a list of project(s). Supercedes Project ID.
------------------	--

Substitute Order

(Available with Order Execution Mgt.)

Substitute Order exchanges the current product item's sales order with the sales order of a product item at a particular location in a specified region. For substitution to work, the selected item must have an extended attribute named ASSOCIATE that identifies the Item Type of its associated item.

A PRT model is set up to track an assembly that is represented as the Item Type BODY.

- The BODY Item Type has an extended attribute ASSOCIATE.
- The value of ASSOCIATE is the value of another Item Type, ORDER.
- ORDER may be defined in the same, or another model.

ASSOCIATE attribute values are as follows.

ASSOCIATE attribute of the:	Is assigned the Value of the Order ID of:
Currently selected body item	A body's ASSOCIATE attribute in the specified region location.
Body item in the specified region location	The currently selected body.

All other attributes and status bits are similarly swapped between the currently selected body and the body at the specified region location.

While the example above shows substitution of orders for different bodies, it is also possible to substitute bodies for different orders. You can only substitute the Item Type that is associated with the Item Type you have currently selected.

- Both sales orders and products are items.
- You can use this function block if product items exist at some region defined in the current project's RCO database, and order items exist in either the current project or another running project to which you have access.
- You can search for the specified region across multiple projects. Enter a comma-delimited list of project names where the region may be found into the **Project ID** field and recompile.
- Where all item eliminations (by any and all methods) have been completed.

1. Insert another function block such as:

- Select Lowest Weight
- Select Source By Age

before Substitute Attributes Order to insure that the most appropriate item(s) are selected.

If you have criteria to determine which item should be selected in case of a tie, specify that criteria in an RSA function block inserted before the Select block.

2. Insert Substitute Order.

Parameters

This function block has the following parameters:

Parameter	Description
Region ID	ID of Region where the item is located.
Region Location	Location of the item in the Region. Note: The reserved names PRT_FIRST and PRT_LAST can be entered here to specify the first or last location, respectively, within the region.
Item Class	Class of the item at this region and location whose associated item (generally of a different class) will be exchanged
Project ID	(Optional) Names of one or more projects in which the selected region may be found. Note: Leave this field blank to limit the search for regions to the local project only.
Project Point ID	(Optional) Text point. Stores a list of project(s). Supercedes Project ID.

TADB Function Blocks List

TADB Function Blocks List

(Available with Order Execution Mgt.)

- Get TADB Item Attribute
- Modify TADB Item Attribute
- TADB Commit
- TADB Initilize
- TADB SetAttribute

Get TADB Item Attribute

(Available with Order Execution Mgt.)

Description

`Get TADB Item Attribute` returns the specified attribute value from the TADB Database

Parameters

This function block has the following parameters.

Parameter	Description
Item Type	Item Type
Item ID	Item ID to get the attribute value that belongs to this Item ID.
Item Group	Item Type Group defined in the TADB Database.
Attribute ID	Attribute ID whose value is to be set in the RCO Variable.
Attribute Value	RCO variable where value is to be returned.

Modify TADB Item Attribute

(Available with Order Execution Mgt.)

Description

`Modify TADB Item Attribute` modifies the attribute value of specified attribute ID from the TADB Database

Parameters

This function block has the following parameters.

Parameter	Description
Item Type	Item type
Item ID	Item ID to get the attribute value that belongs to this Item ID.
Item Group	Item Type Group defined in the TADB Database.
Attribute ID	Attribute ID whose value will be modified.
Attribute Value	New value for this attribute.

TADB Commit

(Available with Order Execution Mgt.)

Description

`TADB Commit` saves the attribute value in the TADB Database.

Parameters

This function block has no parameters.

 **Note:**

1. When you want to set the value of more than one attribute in the database you need to call the following function blocks in the following order.

Function Block	Procedure
1 <code>TADB Initilize</code>	When called <code>TADB Initilize</code> initializes the TADB Root object.
2 <code>TADB Set Attribute</code>	Call consecutively <code>TADB Set Attribute</code> function blocks for each attribute that belongs to that item type group.
3 <code>TADB Commit</code>	When you have completed setting the attributes immediately call <code>TADB Commit</code> to update the value in TADB Database.

Example

```

TADB Initilize
TADB Set Attribute //For attribute 1 Set Item
ID.
TADB Set Attribute //For attribute
2
TADB Set Attribute //For attribute
3
TADB Set Attribute //For attribute
4
TADB Commit //this will save all the attribute value in
database.

```

2. If you are setting attributes in the TADB Database where:
 - Tables have a relationship with Items and
 - The Item ID is the foreign key in the respective record that is set

Then you need to set the:

- Attribute ID as id. and
- Attribute value as Item ID.

TADB Initilize

(Available with Order Execution Mgt.)

Description

`TADB Initilize` initializes the TADB Root object.

Parameters

This function block has the following parameters.

Parameter	Description
Item ID	Item ID to get the attribute value that belongs this Item ID.
Item Type	Item type
Item Group	Item Type Group defined in the TADB Database.

 **Note:**

1. When you want to set the value of more than one attribute in the database you need to call the following function blocks in the following order.

Function Block	Procedure
1 <code>TADB Initilize</code>	When called <code>TADB Initilize</code> initializes the TADB Root object.
2 <code>TADB Set Attribute</code>	Call consecutively <code>TADB Set Attribute</code> function blocks for each attribute that belongs to that item type group.
3 <code>TADB Commit</code>	When you have completed setting the attributes immediately call <code>TADB Commit</code> to update the value in TADB Database.

Example

```

TADB Initilize
TADB Set Attribute //For attribute 1 Set Item
ID.
TADB Set Attribute //For attribute
2
TADB Set Attribute //For attribute
3
TADB Set Attribute //For attribute
4
TADB Commit //this will save all the attribute value in
database.

```

2. If you are setting attributes in the TADB Database where:
 - Tables have a relationship with Items and
 - The Item ID is the foreign key in the respective record that is set

Then you need to set the:

- Attribute ID as id. and
- Attribute value as Item ID.

TADB Set Attribute

(Available with Order Execution Mgt.)

Description

`TADB Set Attribute` sets the value of specified attribute ID in memory (e.g. in the `TADB Interface` object) and not yet saved in the TADB database.

Parameters

This function block has the following parameters.

Parameter	Description
Attribute ID	Attribute ID whose value will be set into the RCO Variable.
Attribute Value	RCO Variable where the value is to be returned.

Note:

1. When you want to set the value of more than one attribute in the database you need to call the following function blocks in the following order.

Function Block	Procedure
1 <code>TADB Initilize</code>	When called <code>TADB Initilize</code> initializes the TADB Root object.
2 <code>TADB Set Attribute</code>	Call consecutively <code>TADB Set Attribute</code> function blocks for each attribute that belongs to that item type group.
3 <code>TADB Commit</code>	When you have completed setting the attributes immediately call <code>TADB Commit</code> to update the value in TADB Database.

Example

```

TADB Initilize
TADB Set Attribute //For attribute 1 Set Item
ID.
TADB Set Attribute //For attribute
2
TADB Set Attribute //For attribute
3
TADB Set Attribute //For attribute
4
TADB Commit //this will save all the attribute value in
database.

```

2. If you are setting attributes in the TADB Database where:
 - Tables have a relationship with Items and
 - The Item ID is the foreign key in the respective record that is set

Then you need to set the:

- Attribute ID as id. and
- Attribute value as Item ID.

Tracker Data Logging

Tracker Data Logging

- Tracker data logging files overview.
- Common Tracker data logging tables.
- Important PRT logging tables naming requirement.
- Steps to edit Tracker data logging files.
- RCO Site logging summary overview.

Tracker data logging files overview

- Two files contain configuration information about PRT and RCO data logging.

Datalog.idt	Lists the tables that PRT and RCO log to.
Data_field.idt	Defines the fields in each table.

- Configuration information is passed back and forth between the Database Logger and these configuration files.

If you change a logged field that has other fields after it then you need to update the offset field to reflect the change in size. So if you change the ATTR_NAME record to a field_len of 32 you need to change the offset field of the ATTR_VALUE to 68.

Example

Datalog.idt

A	Tables are listed in the Database Logger.
B	The same list displays in the datalog.idt.

Data_Field.idt

Fields listed for the ALARM_LOG in the **data_field.idt** are the columns in the CIMPLICITY alarm logging file, ALARM_LOG.

Common Tracker data logging tables

Tables that will almost always be listed and defined include:

Table Name	Logs:
ALARM_LOG	Alarms
EVENT_LOG	Alarms for events
PRT_GRP	PRT groups
PRT_ITEM	PRT items
PRT_REG	PRT regions
SPLIT	RCO information
\$DEFAULT	Point data
PRT_ITEM_ATTR	Extended attributes
RCO_SITE1	An RCO site. Note: The RCO site name is used instead of RCO SITE1.

Important PRT logging tables naming requirement

Note:

- Three tables that are automatically included in the datalog.idt and data_field.idt are:

PRT_GRP

PRT_ITEM

PRT_REG

The PRT in these file names refers to the [PRT service ID \(page 177\)](#).

If your project uses a different service ID, replace the PRT with your service ID.

Example

1. You used the [POMS Wizard \(page 800\)](#) to create a POMS project.
2. The service ID that POMS uses is OMPRT_DC.
3. The three PRT log table names need to be changed to:

OMPRT_GRP

OMPRT_ITEM

OMPRT_REG

- Each RCO site that should be logged must be added to both the `datalog.idt` and `data_field.idt` files.

Steps to edit Tracker data logging files

Step 1 (page 591)	Edit the <code>datalog.idt</code> .
Step 2 (page 593)	Edit the <code>data_field.idt</code> .

Step 1. Edit the Datalog.idt

Step 1. Edit the Datalog.idt

Datalog.idt lists the tables that PRT and RCO log data to.

1. Log file names/specifications that you add to the `datalog.idt` create tables in the Database Logger.
2. Tables that you create in the Database Logger display in the `datalog.idt`. Add as many tables as you need to log data.

Option 1.1 (page 591)	Open and Save Datalog.idt.
Option 1.2 (page 592)	Use a Datalog.idt sample file

Option 1.1. Open and Save the DataLog.idt

Click **Tools>Command Prompt** on the Workbench menu bar.

The Command window opens.

Do the following.

1	Type <code>cd master</code> ; press Enter.
2	Type <code>idtpop Datalog</code> ; press Enter. The Datalog record definition is loaded.
3	Type <code>Notepad datalog.idt</code> , where Notepad is a text editor. Notepad opens with the Datalog.idt file.
4	Make necessary additions, save and close the file. The command window displays.

5 | Type `scpop datalog;` press Enter.

Result: Tracker incorporates your changes.

Option 1.2. Use a Datalog.idt Sample File

The following is a sample `datalog.idt` file.

```

|-* IDT file generated by IDTPOP utility v1.0
* RECORD: DATALOG PARAMETERS FOR DATA LOG FILES
*
* 0 FILE_ID          log file name
* 1 SERVICE_ID      Owner service id for the file
* 2 max_rec_size    max size of a record in bytes
* 3 max_recs        max number of records before rollover
* 4 alarm_rec       threshold for alarm generation
* 5 rollover_recs   number of records to keep on rollover
* 6 auto_rollover   TRUE=rollover on number of records logged
* 7 daily_rollover  TRUE=rollover on time_of_day (sync time)
* 8 add_field       Bit 0=add time stamp, Bit 1=add project
* 9 rollover_cmd    DOS cmd to execute on rollover
* 10 timed_rollover TRUE=rollover on timed basis (periodic)
* 11 archive_id     two character archive id
* 12 compress_arc   TRUE=compress archive file
* 13 batch_pt_id    batch point id associated with log file
* 14 dbms_id        Database in DBMS_DEF file
* 15 table_type     one of A/D/G/X indicating log type
* 16 space_rollover TRUE if rollover on disk full
* 17 gated_rollover TRUE if rollover on digital point
* 18 redund_rollover TRUE if rollover on redundant transition
* 19 rollover_dev   device to check for disk full
* 20 rollover_space limit (in Kb) for disk full rollover
* 21 gate_point     point to gate rollover
* 22 gate_condition 1=HIGH, 2=LOW, 3=TRANSITION
* 23 rollover_units Ok5=DAYS, 4=HOURS, 3=MINUTES
* 24 rollover_period frequency for periodic rollover
* 25 rollover_time  time for synch rollover in min from 00:0
* 26 rollover_export 0=none, 1=all, 2=count, 3=time
* 27 rollover_purge  0=none, 1=all, 2=count, 3=time, 5=exp+al
* 28 rollover_exec   0=none, 1=SQL, 2=DOS, 3=SQL+DOS, 4=Pack,
* 29 export_file     export target file
* 30 purge_file      purge export target file

* 31 export_format   'C'=CSV
* 32 purge_format    'C'=CSV
* 33 export_range    time/# of records to export
* 34 purge_range     time/# of records not to purge
* 35 sql_command     SQL cmd to execute on rollover
* 36 queue_size      Size of the table's write (or insert sta
* 37 is_enabled      Table is enabled (or active) for logging
* 38 is_bulk_ins_enabled Table is using bulk insertion (must use)
* 39 bulk_ins_cache_size Bulk insertion cache size
* 40 bulk_ins_rate    Bulk insertion write rate
*
ALARM_LOG|DL_RP|228|500|0|100|0|1|0||1|A|1|0||$LOGGING|A|0|0|0|0|0|0|5|1|0|0|3|0|||C|C|0|86400||-1|0|0|1000|
60
EVENT_LOG|DL_RP|228|500|0|100|0|1|0||1|E|1|0||$LOGGING|A|0|0|0|0|0|0|5|1|0|0|3|0|||C|C|0|86400||-1|0|0|1000|
60
PRT_GRP|DL_RP|200|500|0|100|1|0|1||0|PG|0||$LOGGING|X|0|0|0|0|0|0|5|1|0|0|3|0|||C|C|0|86400||-1|1|0|1000|60
PRT_ITEM|DL_RP|1700|1000|0|100|1|0|1||0|PI|0||$LOGGING|X|0|0|0|0|0|0|5|1|0|0|3|0|||C|C|0|86400||-1|1|0|
1000|60
PRT_REG|DL_RP|130|500|0|100|1|0|1||0|PR|0||$LOGGING|X|0|0|0|0|0|0|5|1|0|0|3|0|||C|C|0|86400||-1|1|0|1000|60
SPLIT|DL_RP|368|500|0|100|1|0|1||0|PR|0||$LOGGING|X|0|0|0|0|0|0|5|1|0|0|3|0|||C|C|0|86400||-1|1|0|1000|60
$DEFAULT|PTDL_RP|200|500|0|100|0|1|0||1|0||$PTLOG|G|0|0|0|0|0|0|5|1|0|0|3|0|||C|C|0|86400||-1|1|0|1000|60
PRT_ITEM_ATTR|DL_RP|2000|2000|0|100|0|1|1|1||0||$LOGGING|X|0|0|0|0|0|0|5|1|0|0|3|0||PRT_ITEM_ATTR|C|C|0|
86400||-1|1|0|100|10
RCO_SITE1|DL_RP|350|500|0|100|1|0|1||0|PR|0||DB31467|X|0|0|0|0|0|0|5|1|0|0|3|0|||
C|C|0|86400||-1|1|0|1000|60

```

Step 2. Edit the Data_field.idt

Step 2. Edit the Data_field.idt

Data_field.idt defines the fields in each logging table.

The following three tables are the basic tables that are used to hold logged PRT data. Fields for these tables are already defined in the data_field.idt file.

The tables are:

Table	Contains Data Logged for any PRT:
PRT_GRP <i>(page 593)</i>	Group.
PRT_ITEM <i>(page 593)</i>	Item or Item attribute.
PRT_REG <i>(page 594)</i>	Region.
RCO_SITE <i>(page 596)</i>	RCO site

! **Important:** [Change the names \(page 590\)](#) of these three files if the service ID is not PRT_DC.

Fields that are listed for the tables are:


PRT_GRP Fields	Description
View fields in sample file.	
Event_Type	Text description of the type of event which triggered logging.
Group_Id	Name of the group for which the event occurred.
Ext_Hold_Active	External hold state of the group.
Ext_Hold_Reason	Text description of reason for external hold activation.
Ext_Service_Id	Name of the external process, which modified the group's status mask.
Comment	Extra data sent to PRT to describe the changes made.
User_Id	Identifies the agent, which modified the group's status mask.
PRT_ITEM Fields	Description
View fields in sample file.	

Event_Type	Text description of the type of event which triggered logging.
Entry_Time	Time the Item entered the current region.
Modify_Bitmask	Mask whose bits identify which tracking fields for this Item were modified.
Region_1_Id	Name of exited region, in the case of a move, otherwise name of region, which contains modified Item.
Region_1_Loc	Location of Item in exited region, in the case of a move, otherwise location in region of modified Item.
Region_2_Id	Name of entered region, in the case of a move.
Region_2_Loc	Location of Item in entered region, in the case of a move.
Item_Tracking_Type	Types: Serialized, Non-serialized.
Item_Id	Item modified.
Reference_Id	RefId of the modified Item.
Item_Type_Id	Item type of the modified Item.
Assoc_Item_Id	Id of serialized item with which this Item is associated.
Item_Status	Items status mask.
Int_Hold_Active	Internal Hold State.
Int_Hold_Reason	Text Description of reason for internal hold activation.
Ext_Hold_Active	External hold state.
Group_Id	Name of group associated with activating external hold.
Num_Valid_Atts	Number of valid Item attributes.
Att_<x>_Name	<x> = 1 to Num_Valid_Atts. Names of each valid attribute.
Att_<x>_Value	<x> = 1 to Num_Valid_Atts. Values of each valid attribute.
Comment	Extra data sent to PRT to describe the changes made.
User_Id	User who modified the Item.
PRT_REG Fields	Description
View fields in sample file.	
Event_Type	Text description of the type of event which triggered logging.
Region_Id	Name of modified region.
Region_Status	Region status mask.
Comment	Extra data sent to PRT to describe the changes made.
User_Id	Identifies the agent which modified the Region.
PRT_GRP Fields	Description
View fields in sample file.	

Event_Type	Text description of the type of event which triggered logging.
Group_Id	Name of the group for which the event occurred.
Ext_Hold_Active	External hold state of the group.
Ext_Hold_Reason	Text description of reason for external hold activation.
Ext_Service_Id	Name of the external process, which modified the group's status mask.
Comment	Extra data sent to PRT to describe the changes made.
User_Id	Identifies the agent, which modified the group's status mask.
PRT_ITEM Fields	Description
View fields in sample file.	
Event_Type	Text description of the type of event which triggered logging.
Entry_Time	Time the Item entered the current region.
Modify_Bitmask	Mask whose bits identify which tracking fields for this Item were modified.
Region_1_Id	Name of exited region, in the case of a move, otherwise name of region, which contains modified Item.
Region_1_Loc	Location of Item in exited region, in the case of a move, otherwise location in region of modified Item.
Region_2_Id	Name of entered region, in the case of a move.
Region_2_Loc	Location of Item in entered region, in the case of a move.
Item_Tracking_Type	Types: Serialized, Non-serialized.
Item_Id	Item modified.
Reference_Id	RefId of the modified Item.
Item_Type_Id	Item type of the modified Item.
Assoc_Item_Id	Id of serialized item with which this Item is associated.
Item_Status	Items status mask.
Int_Hold_Active	Internal Hold State.
Int_Hold_Reason	Text Description of reason for internal hold activation.
Ext_Hold_Active	External hold state.
Group_Id	Name of group associated with activating external hold.
Num_Valid_Atts	Number of valid Item attributes.
Att_<x>_Name	<x> = 1 to Num_Valid_Atts. Names of each valid attribute.
Att_<x>_Value	<x> = 1 to Num_Valid_Atts. Values of each valid attribute.
Comment	Extra data sent to PRT to describe the changes made.
User_Id	User who modified the Item.

RCO_SITE Fields	Description
	View fields in sample file.
	<p><RCO_SITE1> is the name of each RCO site for which logging should be enabled.</p> <ul style="list-style-type: none"> • RCO_SITE1 is replaced with the name of the actual RCO site. • Each RCO site is listed separately.
operation	<p>Has the value the kind of operation that has happened in the RCOSITE. Example</p> <ul style="list-style-type: none"> • RCO Start • Point Read • TRIGGER • POINT_UNAVAILABLE • SCRIPT RUN • USER_TRANS • SCRIPT END • DECISION_SELECT • DECISION_SUCCESS
data_name	<p>Holds the value of the item for which the above operation is performed. Examples</p> <ul style="list-style-type: none"> • operation is "Point Read " <p>then data_name holds "<POINTNAME>"</p> <ul style="list-style-type: none"> • operation is "DECISION_SELECT" <p>data_name holds "<DECISIONNAME>"</p> <ul style="list-style-type: none"> • operation is "SCRIPT RUN" <p>data_name holds "<SCRIPTNAME>"</p>
old_value	<p>Holds the value of the data_item after execution of the associated operation. Is used when the following happens.</p> <ul style="list-style-type: none"> • Change in Site status. • An attribute was updated of a Item.
new_value	<p>Holds the value of the data_item after execution of the associated operation. Is used/logged when the following happens.</p> <ul style="list-style-type: none"> • Change in Site status. • A point was updated. • A point was Read. • An attribute was updated of a Item. • An attribute was accessed of a Item.
comments	Description of action
n1	Used only for points with array elements.
n2	Never used, reserved for future use.

Optional	Any other pertinent information such as: <ul style="list-style-type: none"> • Point value • Process ID • Item ID • Reason decision eliminated.
----------	--

 **Note:** Logging configuration does not require specific fields. Therefore, you can use only the fields that your application requires. However, if you change a logged field that has other fields after it you need to update the offset field to reflect the change in size.

Example

You:

- Change the [ATTR_NAME r \(page 599\)](#) record to a field_len of 32.
- Need to change the offset field of the [ATTR_VALUE \(page 599\)](#) to 68.

Option 2.1. Open and Save the Data_field.idt

Click **Tools>Command Prompt** on the Workbench menu bar.

The Command window opens.

Do the following.

1	Type <code>cd master</code> ; press Enter.
2	Type <code>idtpop Data_field</code> ; press Enter. The Datalog record definition is loaded.
3	Type <code>Notepad data_field.idt</code> , where Notepad is a text editor. Notepad opens with the Datalog.idt file.
4	Make necessary additions, save and close the file. The command window displays.
5	Type <code>scpop data_field</code> ; press Enter.

Result: Tracker incorporates your changes.

Option 2.2. Use a Data_field.idt Sample File

The following is a sample data_field.idt file:

Important: When you open the data_field.idt file, you will see the *_LOG entries. Copy the remaining entries into the file, starting with `PRT_GRP (page 598)|event_type|12|14|1|14|1`.

```
|-* IDT file generated by IDTPOP utility v1.0
* RECORD: DATA_FIELD DLRP Field Definitions
*
* 0 TABLE_ID          DLRP Table Identifier
```

```

* 1 field_id          DLRP Field to be logged
* 2 cimp_type         CIMPLICITY Data Type of Field
* 3 field_len         Size of Field, in cimp_type units for merging /
creation
* 4 offset           Offset of field in DLAP buffer for merging /
creation
* 5 ins_fld_len       Size of Field, in cimp_type units for reading
* 6 ins_offset        Offset of field in DLAP buffer for reading
*

```

```

ALARM_LOG|alarm_id|22|479|4|255|4 ALARM_LOG|alarm_class|12|5|483|5|259 ALARM_LOG|
resource|3|16|488|16|264 ALARM_LOG|logged_by|12|32|504|32|280 ALARM_LOG|reference|
12|256|536|256|312 ALARM_LOG|prev_state|12|1|792|1|568 ALARM_LOG|log_action|12|1|793|
1|569 ALARM_LOG|final_state|12|1|794|1|570 ALARM_LOG|alarm_message|12|512|795|512|
571 ALARM_LOG|timestamp|14|8|1312|8|1088 ALARM_LOG|generation_time|14|8|1320|8|1096
ALARM_LOG|timestamp_utc|14|8|1328|8|1104 ALARM_LOG|generation_time_utc|14|8|1336|
8|1112 EM_LOG|time|12|22|0|22|0 EM_LOG|time_utc|12|22|23|22|23 EM_LOG|event_type|12|
17|45|17|45 EM_LOG|event_source|12|300|64|300|64 EM_LOG|action_type|12|17|364|17|364
EM_LOG|action_target|12|305|383|305|383 EVENT_LOG|alarm_id|22|479|4|255|4 EVENT_LOG|
alarm_class|12|5|483|5|259 EVENT_LOG|resource|3|16|488|16|264 EVENT_LOG|logged_by|12|
32|504|32|280 EVENT_LOG|reference|12|256|536|256|312 EVENT_LOG|alarm_message|12|512|
795|512|571 EVENT_LOG|timestamp|14|8|1312|8|1088 EVENT_LOG|generation_time|14|8|1320|
8|1096 EVENT_LOG|timestamp_utc|14|8|1328|8|1104 EVENT_LOG|generation_time_utc|14|8|
1336|8|1112 PRT_GRP|event_type|12|14|1|14|1 PRT_GRP|group_id|12|16|15|16|15 PRT_GRP|
ext_hold_active|12|1|31|1|31 PRT_GRP|ext_hold_reason|12|40|32|40|32 PRT_GRP|ext_service_id|12|
32|72|32|72 PRT_GRP|comment|12|40|104|40|104 PRT_GRP|user_id|12|32|144|32|144 PRT_ITEM|
event_type|12|14|1|14|1 PRT_ITEM|entry_time|12|21|15|21|15 PRT_ITEM|modify_bitmask|12|4|
36|4|36 PRT_ITEM|region_1_id|12|16|40|16|40 PRT_ITEM|region_1_loc|12|5|56|5|56 PRT_ITEM|
region_2_id|12|16|61|16|61 PRT_ITEM|region_2_loc|12|5|77|5|77 PRT_ITEM|tracking_type|12|3|
82|3|82 PRT_ITEM|item_id|12|20|85|20|85 PRT_ITEM|reference_id|12|20|105|20|105 PRT_ITEM|
item_type_id|12|16|125|16|125 PRT_ITEM|assoc_item_id|12|20|141|20|141 PRT_ITEM|item_status|
12|8|161|8|161 PRT_ITEM|int_hold_active|12|1|169|1|169 PRT_ITEM|int_hold_reason|12|40|170|40|
170 PRT_ITEM|ext_hold_active|12|1|210|1|210 PRT_ITEM|group_id|12|16|211|16|211 PRT_ITEM|
num_valid_atts|12|3|227|3|227 PRT_ITEM|att_1_name|12|16|230|16|230 PRT_ITEM|att_1_value|
12|16|246|16|246 PRT_ITEM|att_2_name|12|16|262|16|262 PRT_ITEM|att_2_value|12|16|278|16|
278 PRT_ITEM|att_3_name|12|16|294|16|294 PRT_ITEM|att_3_value|12|16|310|16|310 PRT_ITEM|
att_4_name|12|16|326|16|326 PRT_ITEM|att_4_value|12|16|342|16|342 PRT_ITEM|att_5_name|12|
16|358|16|358 PRT_ITEM|att_5_value|12|16|374|16|374 PRT_ITEM|att_6_name|12|16|390|16|390
PRT_ITEM|att_6_value|12|16|406|16|406 PRT_ITEM|att_7_name|12|16|422|16|422 PRT_ITEM|
att_7_value|12|16|438|16|438 PRT_ITEM|att_8_name|12|16|454|16|454 PRT_ITEM|att_8_value|12|
16|470|16|470 PRT_ITEM|att_9_name|12|16|486|16|486 PRT_ITEM|att_9_value|12|16|502|16|502
PRT_ITEM|att_10_name|12|16|518|16|518 PRT_ITEM|att_10_value|12|16|534|16|534 PRT_ITEM|
att_11_name|12|16|550|16|550 PRT_ITEM|att_11_value|12|16|566|16|566 PRT_ITEM|att_12_name|
12|16|582|16|582 PRT_ITEM|att_12_value|12|16|598|16|598 PRT_ITEM|att_13_name|12|16|614|
16|614 PRT_ITEM|att_13_value|12|16|630|16|630 PRT_ITEM|att_14_name|12|16|646|16|646
PRT_ITEM|att_14_value|12|16|662|16|662 PRT_ITEM|att_15_name|12|16|678|16|678 PRT_ITEM|
att_15_value|12|16|694|16|694 PRT_ITEM|att_16_name|12|16|710|16|710 PRT_ITEM|att_16_value|

```


12|16|726|16|726 PRT_ITEM|att_17_name|12|16|742|16|742 PRT_ITEM|att_17_value|12|16|758|16|758 PRT_ITEM|att_18_name|12|16|774|16|774 PRT_ITEM|att_18_value|12|16|790|16|790 PRT_ITEM|att_19_name|12|16|806|16|806 PRT_ITEM|att_19_value|12|16|822|16|822 PRT_ITEM|att_20_name|12|16|838|16|838 PRT_ITEM|att_20_value|12|16|854|16|854 PRT_ITEM|att_21_name|12|16|870|16|870 PRT_ITEM|att_21_value|12|16|886|16|886 PRT_ITEM|att_22_name|12|16|902|16|902 PRT_ITEM|att_22_value|12|16|918|16|918 PRT_ITEM|att_23_name|12|16|934|16|934 PRT_ITEM|att_23_value|12|16|950|16|950 PRT_ITEM|att_24_name|12|16|966|16|966 PRT_ITEM|att_24_value|12|16|982|16|982 PRT_ITEM|att_25_name|12|16|998|16|998 PRT_ITEM|att_25_value|12|16|1014|16|1014 PRT_ITEM|att_26_name|12|16|1030|16|1030 PRT_ITEM|att_26_value|12|16|1046|16|1046 PRT_ITEM|att_27_name|12|16|1062|16|1062 PRT_ITEM|att_27_value|12|16|1078|16|1078 PRT_ITEM|att_28_name|12|16|1094|16|1094 PRT_ITEM|att_28_value|12|16|1110|16|1110 PRT_ITEM|att_29_name|12|16|1126|16|1126 PRT_ITEM|att_29_value|12|16|1142|16|1142 PRT_ITEM|att_30_name|12|16|1158|16|1158 PRT_ITEM|att_30_value|12|16|1174|16|1174 PRT_ITEM|att_31_name|12|16|1190|16|1190 PRT_ITEM|att_31_value|12|16|1206|16|1206 PRT_ITEM|att_32_name|12|16|1222|16|1222 PRT_ITEM|att_32_value|12|16|1238|16|1238 PRT_ITEM|att_33_name|12|16|1254|16|1254 PRT_ITEM|att_33_value|12|16|1270|16|1270 PRT_ITEM|att_34_name|12|16|1286|16|1286 PRT_ITEM|att_34_value|12|16|1302|16|1302 PRT_ITEM|att_35_name|12|16|1318|16|1318 PRT_ITEM|att_35_value|12|16|1334|16|1334 PRT_ITEM|att_36_name|12|16|1350|16|1350 PRT_ITEM|att_36_value|12|16|1366|16|1366 PRT_ITEM|att_37_name|12|16|1382|16|1382 PRT_ITEM|att_37_value|12|16|1398|16|1398 PRT_ITEM|att_38_name|12|16|1414|16|1414 PRT_ITEM|att_38_value|12|16|1430|16|1430 PRT_ITEM|att_39_name|12|16|1446|16|1446 PRT_ITEM|att_39_value|12|16|1462|16|1462 PRT_ITEM|att_40_name|12|16|1478|16|1478 PRT_ITEM|att_40_value|12|16|1494|16|1494 PRT_ITEM|comment|12|40|1510|40|1510 PRT_ITEM|user_id|12|32|1550|32|1550 PRT_REG|event_type|12|14|1|14|1 PRT_REG|region_id|12|16|15|16|15 PRT_REG|region_status|12|8|31|8|31 PRT_REG|comment|12|40|39|40|39 PRT_REG|user_id|12|32|79|32|79 SPLIT|operation|12|20|0 SPLIT|data_name|12|55|20 SPLIT|old_value|12|32|75 SPLIT|new_value|12|32|107 SPLIT|comments|12|121|139 SPLIT|n1|7|4|260 SPLIT|n2|7|4|264 PRT_ITEM_ATTR|UUID|12|36|0|36|0 PRT_ITEM_ATTR|Attr_Name|12|256|36|256|36 PRT_ITEM_ATTR|Attr_Value|12|512|292|512|292 [RCO_SITE1 \(page 601\)](#) |operation|12|20|0|20|0 RCO_SITE1|data_name|12|280|20|280|20 RCO_SITE1|old_value|12|32|300|32|300 RCO_SITE1|new_value|12|32|332|32|332 RCO_SITE1|comments|12|1024|364|1024|364 RCO_SITE1|n1|7|4|1388|4|1388 RCO_SITE1|n2|7|4|1392|4|1392 RCO_SITE1|optional|12|80|1396|80|1396

RCO Site Logging Summary Overview

Several steps must be taken to enable logging for an RCO site, as follows.

1 (page 600)	Work in the TrackerCfg_UI.
2 (page 600)	Edit the datalog.idt file.

3 (page 601)	Edit the data_field.idt file.
4 (page 601)	Configure the data log source in the Database Logger.

Work in the TrackerCfg_UI.

1. Lock the RCO_SITE1.
2. Right-click the RCO_SITE1 object.
3. Select Alarming/Logging on the Popup menu.

The Alarming/Logging Options dialog box opens.

1. Check Data Logger.
2. Check the features that you want to log in the Logging list.

1. Repeat the process for each RCO site for which you want logging.
2. Click OK.

Edit the datalog.idt file

1. [Open \(page 591\)](#) the datalog.idt file.
2. Add the following at the end of the datalog.idt.

```
RCO_SITE1|DL_RP|1477|500|0|100|1|0|1||0|PR|0||DB31467|X|0|0|0||0|5|1|0|0|3|0||- C|C|0|
86400||-1|1|0|1000|60
```

Notes

- When you configure the RCO site data logging, the name of an actual RCO site will replace RCO_SITE1.

This will add the specified RCO site as a new table in the Database logger, e.g. RCO_SITE1.

- If you want logging for RCO_SITE2, add a new line as follows.

```
RCO_SITE2|DL_RP|1477|500|0|100|1|0|1||0|PR|0||DB31467|X|0|0|0||0|5|1|0|0|3|0||- C|C|0|
86400||-1|1|0|1000|60
```

! **Important:** The third value in the datalog addition is 278 a minimum of 268 is required for RCO data logging records.

1. [Save and `scpop` \(page 591\)](#) the changed datalog.idt file.

Edit the data_field.idt file

1. [Open \(page 597\)](#) the data_field.idt file.
2. Add the following at the end of data_field.idt.

```
RCO_SITE1|operation|12|20|0|20|0 RCO_SITE1|data_name|12|280|20|280|20 RCO_SITE1|
old_value|12|32|300|32|300 RCO_SITE1|new_value|12|32|332|32|332 RCO_SITE1|comments|12|
1024|364|1024|364 RCO_SITE1|n1|7|4|1388|4|1388 RCO_SITE1|n2|7|4|1392|4|1392 RCO_SITE1|
optional|12|80|1396|80|1396
```

1. [Save and `scpop` \(page 597\)](#) the data_field.idt file.
2. Configure the data log source in the Database Logger

1. Open the Database Logger in the project's Workbench.

The newly created RCO site(s) display as new tables.

By default the database associated with these new tables is CIMPPLICITY.

1. Change the ODBC source as per the configuration and requirement.
2. Close Database logger.
3. Run the project.

Result: Whenever any of the RCO sites (configured for data logging) are triggered, data will be logged.

Data is logged for every operation carried out by the triggered RCO site. Data is stored in the Operation column of the configured database table.

Operations include:

RCO Start

TRIGGER

POINT_UNAVAILABLE

Point Read

SCRIPT RUN

SCRIPT_ERROR

SCRIPT END

DECISION_SELECT

USER_TRANS

SCRIPT END

DECISION_SUCCESS

DECISION_FAILURE

VALIDATE_FAILURE

EXECUTE_FAILURE

CTRL_STATUS_CHG

MANUAL_DEC

ASYNCH_ERROR

VERIFY_ERROR

Chapter 4. Tracker Appendix

About the Tracker Appendix

- [Production tracking application interface. \(page 603\)](#)
- [PRT objects for use in the Basic control Engine. \(page 696\)](#)
- [RCO external decision. \(page 754\)](#)

Production Tracking Application Interface

Production Tracking Application Interface

CIMPLICITY software's Production Tracking API lets application programs access Production Tracking functions. These functions include: adding items to the Production Tracking system; deleting items; modifying items; moving items, and modifying Region and Group tracking information. Using the API requires you to do the following:

- Understand the subroutine interfaces and system capabilities provided by CIMPLICITY software's Production Tracking System. These functions are documented below.
- Understand the configuration requirements for the Production Tracking System. The Production Tracking configuration data is described in the previous chapter.
- Code appropriate applications programs.
- Compile and link the programs (as explained in this section).

Contents of the Application Subroutine Interface

These files make up the contents of the Production Tracking Application Program Interface. These files can be used to write applications that interface with the Production Tracking system, and can also be used to compile and link a demonstration program.

The following files are distributed with the Production Tracking Application Interface utilities.

%BSM_ROOT% is the directory where the CIMPLICITY software is installed.

Include Files

%BSM_ROOT%\include\PRT_API.H

Library Files

%BSM_ROOT%\EXE\PRT_API.DLL

%BSM_ROOT%\LIB\PRT_API.LIB

Include Files

The following header files contain definitions used by Production Tracking procedures; they therefore must be included in an application program that interfaces with Production Tracking.

```
#include <inc_path/cor.h>
#include <inc_path/cor_stat.h>
#include <inc_path/sc_recs.h>
#include <inc_path/netcom.h>
```

#include <inc_path/prt_api.h>	(structure definitions, etc. required by the application program.)
--	--

Runtime Notes

If you start making off-node PRT calls to a project on a particular Tracking server, then move the project from the Tracking server to another computer, the local project will no longer be able to find the off-node project. You must restart the local project to pick up the server project's new location.

Application Subroutines

Application Subroutines

prt_api_add_item
prt_api_advance_model
prt_api_check_route
prt_api_delete_item
prt_api_delete_items_at_loc
prt_api_dyn_register_intproc
prt_api_get_intproc_updates
prt_api_get_item_tracking_type
prt_api_get_item_types
prt_api_get_req
prt_api_group_hold
prt_api_init
prt_api_insert_item

prt_api_intproc_region
prt_api_intproc_retry_connect
prt_api_load_item
prt_api_load_item_hold
prt_api_locate_item
prt_api_modify_item
prt_api_modify_item_all
prt_api_modify_item_att
prt_api_modify_region
prt_api_move_item
prt_api_move_item_to_detain
prt_api_prod_start
prt_api_region_setpoints
prt_api_register_intproc
prt_api_reorder_region
prt_api_req_class_itemlist
prt_api_req_groupitemlist
prt_api_req_grouplist
prt_api_req_itemlist
prt_api_req_class_itemlist_Ex
prt_api_req_regionlist
prt_api_send_resp
prt_api_static_region_data
prt_api_term
prt_api_term_intproc
prt_api_unload_groupitemlist
prt_api_unload_grouplist
prt_api_unload_itemlist
prt_api_unload_regionlist
prti_get_RCMconnection
prti_term_RCMconnection

prt_api_add_item

Name

prt_api_add_item

Purpose

Add item to tracking region queue.

Description

This routine provides a mechanism to add an individual Item at the same location where another Item currently resides. Defined constants are provided as a way of indicating that the item is to be added either at the head or at the tail of the queue. The region location can be specified as FIRST (or also as EXIT) to add the item to the first (head) location in the queue, i.e. this item is being added to a location where one or more other items already currently reside (although no more than one serialized item may reside at a region location at any one time). These items will be the next items to transition out of the queue. LAST (and ENTRY) are provided to add the item to the last location in the region (the last location which currently has one or more items residing in it). These items will be the last items to transition out of the region (of all the items currently in the region).

Syntax

```

int prt_api_add_item ( timestamp, ext_proc_flag, region_id,
                      region_loc, item_id, reference_id,
                      item_type_id, parent_item_id,
                      item_status, ext_hold_active,
                      group_id, int_hold_active,
                      int_hold_reason, num_atts_valid,
                      item_att_list, comment, user_or_svc_id,
                      retstat)
cor_time_t          timestamp;
COR_BOOLEAN        ext_proc_flag;
char               region_id[PRT_REGION_ID_LEN + 1];
COR_I2            region_loc;
char               item_id[PRT_ITEM_ID_LEN + 1];
char               reference_id[PRT_ITEM_ID_LEN + 1];
char               item_type_id[PRT_ITEM_TYPE_ID_LEN + 1];
char               parent_item_id[PRT_ITEM_ID_LEN + 1];
COR_U4            item_status;
COR_BOOLEAN        ext_hold_active;
char               group_id[PRT_REGION_ID_LEN + 1];
COR_BOOLEAN        int_hold_active;
char               *int_hold_reason;
COR_I1            num_atts_valid;
PRT_ITEM_ATT      *item_att_list;
char               comment[SC_DESCRIPTION_LEN + 1];

```



```
char      user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS *retstat;
```

Data Structures

See Static Information Requirements

Arguments

Argument	Description
Input	
Serialized Items Only	
ext_proc_flag	External process flag; if TRUE, get all information from external process. (required).
item_id	Unique identifier of a Serialized Item (optional if reference_id specified)
reference_id	Secondary identifier of a Serialized Item (optional if item_id specified)
ext_hold_active	If TRUE, item held due to external hold. required)
group_id	Group identifier providing reason for hold (required if ext_hold_active = TRUE).
num_atts_valid	Number of item attributes valid. This specifies the number of array elements in the item attribute list that contain valid data. If item_att_list is a NULL pointer, the function sets this argument to zero regardless of what was actually passed. (required)
*item_att_list	Item attribute list. This argument is the pointer to an array of PRT_NUM_ATTRIBUTES elements. If the pointer is NULL, this clearly indicates that no item attributes are being provided along with the item being added. If the pointer is non-NULL, however, it must point to an array of exactly PRT_NUM_ATTRIBUTES elements. The num_atts_valid argument indicates how many of these elements contain valid data. (optional)
Non-Serialized Items	
parent_item_id	Associated serialized item identifier (optional).
Both	
Timestamp	The time the item was last moved or modified. The timestamp is used to ensure that, when attempting to add an item, the item(s) currently at the location where the new item is being added have not moved or been modified between the time when information was last obtained for that item(s) and the attempt to add a new item at the same location. (One of the fields in the structure that returns data on an item is the last_mod_time field, which indicates the time at which the item was last moved or modified.) If the timestamp argument provided to this function is older than the timestamp associated with the item(s) at the location being added to, then the 'add' function is not performed and a warning returned. If the timestamp argument passed to this function is zero, the operation is performed without any timestamp checking.
region_id	Region where the item is to be added to the queue (required).
region_loc	Specific location in the region where the Item is to be added (required).
item_type_id	Item type identifier (required).

Argument	Description
item_status	Item status (required).
int_hold_active	If TRUE, item held due to internal hold. (required)
int_hold_reason	Comment specifying reason for internal hold (required if int_hold_active = TRUE).
Comment	Comment to be recorded in PRT history log file. (optional)
user_or_svc_id	User or service identifier. This is used for logging purposes and is optional. If this pointer is NULL, the host process id is used. (optional)
Output	
* retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS or COR_FAILURE.

If the function returns COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

PRTC_NON_SER_ID	Non-serialized item has item or ref. id (Failure)
PRTC_SER_NO_ID	Serialized item missing both item & ref. id (Failure)
PRTC_UNKNOWN_REGION	Unknown REGION specified (Failure)
PRTC_BAD_REG_LOC_SPEC	Bad region location specified (Failure)
PRTC_ITEM_STAMP_TOO_NEW	An affected item has changed since the last display (Failure)
PRTC_ADD_LOC_TOO_BIG	Cannot add item to unoccupied location- inserting (Warning)
PRTC_ADD_MULTIPLE_SERIAL	Cannot have two serialized items at one location (Warning)
PRTI_ITEM_TYPE_NULL	Item type identifier NULL (Warning)
PRTI_ITEM_TYPE_NDEF	Item type identifier not valid (Warning)
PRTI_NS_EXT_PROC_INVALID	ext_proc_flag set for non-serialized item (Warning)
PRTI_NON_SER_ID	Non-serialized item has item id or reference id (Warning)
PRTI_SER_NO_ID	Serialized item missing both item id and reference id (Warning)
PRTI_SER_PARENT	Parent ID specified for serialized item (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_EXT_HOLD_NON_SER	External hold specified for non-serialized item (Warning)
PRTI_INT_HOLD_NO_REASON	Internal hold specified without reason (Warning)

PRTI_NON_SER_ATTS	Non-serialized item has attribute(s) (Warning)
PRTI_NUM_ATTS_INVALID	Invalid number of item attributes specified (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_INVALID_REG_LOC	Invalid region location (Warning)
PRTI_ITEM_TYPE_NULL	Item type identifier NULL (Warning)
PRTI_INVALID_DEST	Destination region is invalid for item of this type (Warning)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)

Plus all error codes returnable by `prt_api_send_cmd`

`prt_api_advance_model`

Name


`prt_api_advance_model`

Purpose

Advance Tracking Model

Description

This routine provides a mechanism for advancing the next Item Carrier to leave a Source Tracking Region to a Destination Tracking Region.

 **Note:** This call moves all items at the FIRST (EXIT) location in the source region to the LAST (ENTRY) location in the destination region.

Syntax

```
int prt_api_advance_model ( src_reg_id, dest_reg_id, comment,
                          user_or_svc_id, retstat )
char   src_reg_id[PRT_REGION_ID_LEN + 1];
char   dest_reg_id[PRT_REGION_ID_LEN + 1];
char   comment[SC_DESCRIPTION_LEN + 1];
char   user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS *retstat;
```

Data Structures

See Static Information Requirements

Arguments

Argument	Description
Input	
src_reg_id	Source region id; region the item is moving out of (required).
dest_reg_id	Destination region id; region the item is moving into (required).
comment	Comment to be recorded in PRT history log file (optional).
user_or_svc_id	User or service identifier. This is used for logging purposes and is optional. If this pointer is NULL, the host process id is used. (optional)
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS or COR_FAILURE.

If the function returns COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

All error codes returnable by `prt_api_move_item`.

`prt_api_check_route`

Name

`prt_api_check_route`

Purpose

Validate tracking route.

Description

This routine provides a mechanism for validating that an ITEM of a specific type is configured to move from a Source Tracking Region to a Destination Tracking Region. Route validation is performed locally by the API from configuration information that was read into the API's internal

schema during initialization. If the function returns `COR_SUCCESS`, the route is valid. If some other status is returned, the `err_code` field of the `COR_STATUS` structure will contain additional information. If a valid route does exist between the source and destination regions specified (according to the configuration data) but the item type specified is not allowed to transition along this route, `PRTI_INVALID_ROUTE` is returned in the `err_code` field. If there are no routes configured between the source and destination regions specified, the `err_code` field is set to `PRTI_NO_SUCH_ROUTE`.

Syntax

```
int prt_api_check_route ( src_region_id, dest_region_id,
                        item_type_id, retstat )
char   src_region_id[PRT_REGION_ID_LEN + 1];
char   dest_region_id[PRT_REGION_ID_LEN + 1];
char   item_type_id[PRT_ITEM_TYPE_ID_LEN + 1];
COR_STATUS *retstat;
```

Data Structures

See Static Information Requirements

Arguments

Argument	Description
Input	
<code>src_region_id</code>	Source region id; the starting region of the route (required).
<code>dest_region_id</code>	Destination region id; the destination region of the route (required).
<code>item_type_id</code>	Item type identifier (required).
Output	
<code>*retstat</code>	Pointer to <code>COR_STATUS</code> structure

Return Value

Either `COR_SUCCESS` or `COR_FAILURE`.

If the function returns `COR_FAILURE` additional error information can be found in the `COR_STATUS` structure.

Error Codes

<code>PRTI_REG_ID_NULL</code>	Region identifier NULL (Warning)
<code>PRTI_REG_ID_NDEF</code>	Region identifier not valid (Warning)
<code>PRTI_ITEM_TYPE_NULL</code>	Item type identifier NULL (Warning)

PRTI_ITEM_TYPE_NDEF	Item type identifier not valid (Warning)
PRTI_NO_SRC_OR_DEST	Route has no source and no destination region (Warning)
PRTI_INVALID_ROUTE	Invalid route for item type (Warning)
PRTI_NO_SUCH_ROUTE	No route exists between src/dest regions (Warning)

prt_api_delete_item

Name

`prt_api_delete_item`

Purpose

Delete/scrap item from tracking region queue.

Description

This function provides a mechanism for deleting or scrapping a single Item from a specific Tracking Region queue. The calling routine receives back a status indicating the success or failure of the request.

Syntax

```
int prt_api_delete_item ( timestamp, prt_svc_id, group_id,
                        region_id, region_loc, item_id,
                        reference_id, item_type_id,
                        disposition, comment, user_or_svc_id,
                        retstat)
COR_time_t      timestamp;
char            prt_svc_id[SERVICE_ID_LEN + 1];
char            group_id[PRT_REGION_ID_LEN + 1];
char            region_id[PRT_REGION_ID_LEN + 1];
COR_I2         region_loc;
char            item_id[PRT_ITEM_ID_LEN + 1];
char            reference_id[PRT_ITEM_ID_LEN + 1];
char            item_type_id[PRT_ITEM_TYPE_ID_LEN + 1];
COR_I1         disposition;
char            comment[SC_DESCRIPTION_LEN + 1];
char            user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS     *retstat;
```

Data Structures

None

Arguments

Argument	Description								
Input									
Serialized Items Only	(See SYNTAX for proper order)								
item_id	Unique identifier of a serialized item (optional if reference_id specified)								
reference_id	Secondary identifier of a serialized item (optional if item_id specified)								
Both									
Timestamp	The time the item was last moved or modified. The timestamp is used to ensure that, when attempting to delete an item, the item has not changed (or moved) between the time when the time when information was last obtained for that item and the attempt to delete it. (One of the fields in the structure which returns data on an item is the last_mod_time field, which indicates the time at which the item was last moved or modified.) If the timestamp argument provided to this function is older than the item's current timestamp, then the deletion is not performed and a warning returned. If the timestamp argument passed to this function is zero, the operation is performed without any timestamp checking.								
prt_svc_id	PRT service identifier (optional)								
group_id	Identifier of the Tracking Group (optional)								
region_id	Identifier of the Tracking Region where the Item is to be deleted from the queue (required for non-serialized items, optional for serialized items)								
	Note: For serialized items, at least one of the above three arguments must be specified.								
region_loc	Location in the region where the item resides (required). For non-serialized items, a specific region location must be specified. For serialized items, the constant ALL_LOCATIONS may be used. This indicates that the item may reside at any location in the region.								
item_type_id	Item type identifier (required for non-serialized items, optional for serialized items)								
Disposition	Disposition of item to be deleted. (required) Valid values are:								
	<table border="1"> <tr> <td>SCRAP</td> <td>Causes the item to be scrapped.</td> </tr> <tr> <td>NO_LOCK_CHECK_SCRAP</td> <td>Causes the item to be scrapped, overriding region locking status.</td> </tr> <tr> <td>DELETE</td> <td>Causes the item to be deleted.</td> </tr> <tr> <td>NO_LOCK_CHECK_DELETE</td> <td>Causes the item to be deleted, overriding region locking status.</td> </tr> </table>	SCRAP	Causes the item to be scrapped.	NO_LOCK_CHECK_SCRAP	Causes the item to be scrapped, overriding region locking status.	DELETE	Causes the item to be deleted.	NO_LOCK_CHECK_DELETE	Causes the item to be deleted, overriding region locking status.
SCRAP	Causes the item to be scrapped.								
NO_LOCK_CHECK_SCRAP	Causes the item to be scrapped, overriding region locking status.								
DELETE	Causes the item to be deleted.								
NO_LOCK_CHECK_DELETE	Causes the item to be deleted, overriding region locking status.								
	Note: It may be necessary to force the delete of the item, regardless of the region's lock status, while using the PRT API functions to delete items in a region. To do this, use the "NO_LOCK_CHECK_" versions of SCRAP and DELETE above.								
Comment	Comment to be recorded in PRT history log file (optional)								
user_or_svc_id	User or service identifier (optional)								
Output									
*retstat	Pointer to COR_STATUS structure								

Return Value

Either COR_SUCCESS, COR_WARNING, or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

Error Codes

PRTC_XREF_NOFOUND	Deleted, but not found in xref table (Warning)
PRTC_ITEM_NOT_FOUND	Item ID not found (Failure)
PRTC_REF_NOT_FOUND	Reference ID not found (Failure)
PRTC_BAD_REG_SPEC	Bad region specified (Failure)
PRTC_BAD_REG_LOC_SPEC	Bad region location specified (Failure)
PRTC_NO_ITEM_AT_LOC	Specified item not found specified location (Failure)
PRTC_NO_REG_LOC_SPEC	Must specify region location to delete non-serialized item (Failure)
PRTC_ITEM_STAMP_TOO_NEW	An affected item has changed since the last display (Failure)
PRTI_ITEM_TYPE_NULL	Item type identifier NULL (Warning)
PRTI_ITEM_TYPE_NDEF	Item type identifier not valid (Warning)
PRTI_NON_SER_ID	Non-serialized item has item id or reference id (Warning)
PRTI_SVC_ID_NULL	PRT service identifier NULL (Warning)
PRTI_SVC_ID_NDEF	PRT service identifier not valid (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_INVALID_REG_LOC	Invalid region location (Warning)
PRTI_INVALID_REG_LOC_MSG	Invalid region location (Warning)
PRTI_REG_NOT_IN_GRP	Region not in group specified (Warning)
PRTI_REG_NOT_IN_SVC	Region not managed by service specified (Warning)
PRTI_GRP_NOT_IN_SVC	Group ID inconsistent with service specified (Warning)
PRTI_NO_SVC_IMPLIED	No service specified (via svc_id/group_id/region_id) (Warning)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)

Plus all error codes returnable by `prt_api_send_cmd`.

`prt_api_delete_items_at_loc`

Name

`prt_api_delete_items_at_loc`

Purpose

Delete/scrap items from tracking region queue location.

Description

This function lets users delete or scrap all Items at a location in a specific Tracking Region queue. The calling routine receives back a status indicating the success or failure of the request.

Syntax

```
int prt_api_delete_items_at_loc (timestamp, region_id,
                                region_loc, disposition,
                                comment, user_or_svc_id,
                                retstat)

cor_time_t      timestamp;
char            region_id[PRT_REGION_ID_LEN + 1];
COR_I2         region_loc;
COR_I1         disposition;
char            comment[SC_DESCRIPTION_LEN + 1];
char            user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS     *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	(See SYNTAX for proper order)

Argument	Description
Timestamp	The time the item was last moved or modified. The timestamp is used to ensure that, when attempting to delete an item, the item has not changed (or moved) between the time when the time when information was last obtained for that item and the attempt to delete it. (One of the fields in the structure that returns data on an item is the last_mod_time field, which indicates the time at which the item was last moved or modified.) If the timestamp argument provided to this function is older than the item's current timestamp, then the deletion is not performed and a warning returned. If the timestamp argument passed to this function is zero, the operation is performed without any timestamp checking.
region_id	Identifier of the Tracking Region where the Items are to be deleted from the queue
	Note: For serialized items, at least one of the above three arguments must be specified.
region_loc	Location in the region where the items reside (required)
Disposition	Disposition of item to be deleted. Valid (required) values are:
	SCRAP Causes the item to be scrapped.
	NO_LOCK_CHECK_SCRAP Causes the item to be scrapped, overriding region locking status.
	DELETE Causes the item to be deleted.
	NO_LOCK_CHECK_DELETE Causes the item to be deleted, overriding region locking status.
	Note: It may be necessary to force the deletion of the item, regardless of the region's lock status, while using the PRT API functions to delete items in a region. To do this, use the "NO_LOCK_CHECK_" versions of SCRAP and DELETE above.
Comment	Comment to be recorded in prt history log file (optional)
user_or_svc_id	User or service identifier (optional)
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING, or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

Error Codes

PRTC_BAD_REG_SPEC	Bad region specified (Failure)
PRTC_BAD_REG_LOC_SPEC	Bad region location specified (Failure)
PRTC_NO_ITEM_AT_LOC	Specified item not found specified location (Failure)
PRTC_NO_REG_LOC_SPEC	Must specify region location to delete non-serialized item (Failure)
PRTC_ITEM_STAMP_TOO_NEW	An affected item has changed since the last display (Failure)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)

PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_INVALID_REG_LOC	Invalid region location (Warning)
PRTI_NO_SVC IMPLIED	No service specified (via region_id) (Warning)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)

Plus all error codes returnable by **prt_api_send_cmd**.

`prt_api_dyn_register_intproc`

Name

`prt_api_dyn_register_intproc`

Purpose

Register as a dynamically interested process.

Description

This function provides a mechanism for a process to request to register as an interested process to a PRT process. Either this initialization routine must be called by all PRT interested processes prior to calling any other PRT_API interested process utilities or **prt_api_register_intproc**. The calling routine receives back a status indicating the success or failure of the request. This routine is used when you have not configured your process as a service and is not included in the **prt_service.dat** file. Otherwise you must use **prt_api_register_intproc**.

Syntax

```
int prt_api_dyn_register_intproc (intproc_event_flag,
                                prt_system, cimp_sys, retstat)
COR_I4      intproc_event_flag;
char        *prt_system;
char        *cimp_sys;
COR_STATUS  *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	

Argument	Description
intproc_event_flag	Interested process event flag (required)
prt_system	Prefix of PRT data server(s) to connect to.
cimp_sys	Name of project or node where prt data server resides.
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS or COR_FAILURE.

If the function returns COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

PRTI_PROC_NOT_CONFIGURED	Process is not configured as network process (Failure)
PRTI_FILE_OPEN_ERR	Error opening one of the configuration files (Failure)
PRTI_SVC_ID_NDEF	PRT service identifier not valid (Failure)
PRTI_FILE_READ_ERR	Error reading configuration file (Failure)
PRTI_EMPTY_FILE	Configuration file has no records (Failure)
PRTI_AUX_NAM_ERR	Error in ipc_aux_nam (Failure)
PRTI_ADD_PORT_ERR	Error in ipc_add_port (Failure)
PRTI_READ_PORT_ERR	Error in ipc_read_port (Failure)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)
PRTI_BAD_MSG	Error in segment (Failure)
PRTI_BAD_SVC_NAME	Bad service name (no underscore) (Failure)
PRTI_XLATE_ERR	Error in ipc_xlate getting physical addr (Failure)
PRTI_WRITE_PORT_ERR	Error in ipc_write_port (Warn/Fail)

`prt_api_get_intproc_updates`

Name

`prt_api_get_intproc_updates`

Purpose

Get interested process item/region tracking data updates.

Description

This routine provides a mechanism for an application to receive interested process updates from all data servers which have been specified based on the **prt_system** argument in the **prt_api_init** and **prt_api_dyn_register_intproc** calls. The information received by the interested process includes updates for Serialized and Non-serialized Items as well as region-related updates.

There are currently two types of updates that an interested process may receive from PRT:

- An item has been modified or moved - the manifest constant for this update type is **PRT_INTPROC_ITEM_UPDATE**. The **event_code** field in the **PRT_ITEM_UPDATE_STRUCT** indicates the exact nature of the update.
- The items in a region have been reordered - the manifest constant for this update type is **PRT_INTPROC_REGION_UPDATE**.

With both types of update structure, the **action** field contains the same value as the **update_type** argument returned to the calling program.

The caller allocates storage for both item update and region update structures. When the call returns, **update_type** will be set to **PRT_INTPROC_ITEM_UPDATE** if the item update structure has been filled in with item information. If the region update structure has been filled in, **update_type** will be set to **PRT_INTPROC_REGION_UPDATE**.

If a Data Server process which the interested process is configured to receive interested process updates from terminates (or some other error occurs) the interested process will be notified by receiving a message with **update_type** set to **PRT_ERROR_RESP**. In this case the service id of the associated Data Collector process will be placed in the **user_or_svc_id** field of the **PRT_ITEM_UPDATE_STRUCT** structure.

When **prt_api_get_intproc_updates** is called, it empties the datagram receive buffer of all messages that have arrived since the last call to **prt_api_get_intproc_updates** (or since the process registered as an interested process) via successive reads, until a read is posted and the event flag remains low. The messages which are read are placed into one of two lists, one containing errors messages, and the other containing valid interested process update messages. If there are any messages in the error message list, these are always returned before the real updates.

Syntax

```
int prt_api_get_intproc_updates( update_type, item_info,
                               reg_info, retstat )
COR_U2                          *update_type;
PRT_ITEM_UPDATE_STRUCT          *item_info;
```

```
PRT_REGION_UPDATE_STRUCT *reg_info;
COR_STATUS                *retstat;
```

Data Structures

```
typedef struct
{
    char att_name[PRT_ATTRIBUTE_LEN + 1] - item attribute name
    char att_value[PRT_ATTRIBUTE_LEN + 1] - item attribute value
} PRT_ITEM_ATT
typedef struct
{
    long          reg_entry_time;
    COR_I2        event_code;
    COR_U2        modify_bitmask;
    char          region_1_id[PRT_REGION_ID_LEN + 1];
    COR_I2        region_1_loc;
    char          region_2_id[PRT_REGION_ID_LEN + 1];
    COR_I2        region_2_loc;
    COR_I1        item_tracking_type;
    char          item_id[PRT_ITEM_ID_LEN + 1];
    char          reference_id[PRT_ITEM_ID_LEN + 1];
    char          item_type_id[PRT_ITEM_TYPE_ID_LEN + 1];
    char          parent_item_id[PRT_ITEM_ID_LEN + 1];
    COR_U4        item_status;
    COR_BOOLEAN   int_hold_active;
    char          int_hold_reason[SC_DESCRIPTION_LEN + 1];
    COR_BOOLEAN   ext_hold_active;
    char          group_id[PRT_REGION_ID_LEN + 1];
    COR_I1        num_atts_valid;
    PRT_ITEM_ATT  item_att_list[PRT_NUM_ATTRIBUTES];
    char          comment[SC_DESCRIPTION_LEN + 1];
    char          user_or_svc_id[SERVICE_ID_LEN + 1];
    COR_I1        action;
} PRT_ITEM_UPDATE_STRUCT, *PRT_ITEM_UPDATE_STRUCT_PTR;
typedef struct
{
    COR_I1 event_code;
    char   region_id[PRT_REGION_ID_LEN + 1];
    COR_U4 region_status;
    char   comment[SC_DESCRIPTION_LEN + 1];
    char   user_or_svc_id[SERVICE_ID_LEN + 1];
    COR_I2 location;
    COR_I1 action;
} PRT_REGION_UPDATE_STRUCT, *PRT_REGION_UPDATE_STRUCT_PTR;
```

Notes: See the next chapter for the list of macros that can be used with the **modify_bitmask** parameter to determine which fields in the **PRT_ITEM_UPDATE_STRUCT** contain valid data.

The values supplied which identify region location are in the range FIRST (or EXIT), LAST (or ENTRY), and 0, 1, 2...n. '0' means that the region location is not appropriate. 1-n is the region location. FIRST and LAST are first and last to exit, respectively.

In the current implementation of production tracking, **region_2_id** and **region_2_loc** are not populated in the item update structure. These elements are reserved for GE Intelligent Platforms use.

Arguments

Argument	Description
Input	
item_info	Update information storage location on an item
reg_info	Update information storage location on a region
Output	
*update_type	Interested process update type
item_info	Updated information on an item
reg_info	Updated information on a region
*retstat	Pointer to COR_STATUS structure Note: See the chapter on event codes for a complete listing of the possible event codes for either item or region updated information.

Return Value

Either COR_SUCCESS, COR_FAILURE or COR_WARNING.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

If there are no requests pending, the function returns COR_WARNING with the COR_STATUS err_code set to PRTI_NOTHING_PENDING and **update_type** set to PRT_ERROR_RESP.

Error Codes

IPC_ERR_PARTDEAD	Partner Dead (Warning)
PRTI_UNKNOWN_INIT_SRC	MF_INIT segment received from unknown service (Warning)
PRTI_INIT_MSG_RCVD	Init segment received from Data Collector (Warning)
PRTI_UNEXPECTED_SEG	Unexpected segment type (Warning)
PRTI_NOTHING_PENDING	No requests pending (Warning)
PRTI_BAD_MSG	Error in segment (Failure)
PRTI_READ_PORT_ERR	Error in ipc_read_port (Failure)

PRTI_EF_CHECK_ERR	Error checking event flag (Failure)
-------------------	-------------------------------------

prt_api_get_item_tracking_type

Name

prt_api_get_item_tracking_type

Purpose

Return the tracking type for a specific item.

Description

This function returns the tracking type for a specific item id. A value of 1 is returned if the item id is a serialized item otherwise the value 2 is returned.

Syntax

```
int prt_api_get_item_tracking_type ( item_type_id,
                                   tracking_type, retstat )
char          *item_type_id;
COR_I1        *tracking_type;
COR_STATUS    *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	
item_type_id	Pointer to a null terminated string containing a valid item type id.
Output	
tracking_type	Pointer to a 1 byte integer specifying the tracking type. A value of 1 is a serialized item, a value of 2 is a non-serialized item.
retstat	Pointer to COR_STATUS structure.

Return Value

Either COR_SUCCESS or COR_WARNING.

COR_SUCCESS is returned if the item id has been found. COR_FAILURE is returned if the item id cannot be found in any tracking region.

prt_api_get_item_types

Name

prt_api_get_item_types

Purpose

Return a list of all configured item types.

Description

This routine returns a list of all the configured item types in the PRT system. It also returns a counter for the number of items found.

Syntax

```
int prt_api_get_item_types ( item_type_ids, total, retstat )
char          **item_type_ids;
COR_I4        *total;
COR_STATUS    *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	
None	
Output	
item_type_ids	Pointer to a pointer to list of item ids.
Total	Pointer to an integer with the total number of item ids in the item_type_ids list.

Return Value

Either COR_SUCCESS or COR_WARNING.

COR_SUCCESS is returned as long as the list exists. COR_WARNING is returned if no list exists.

`prt_api_get_req`

Name

`prt_api_get_req`

Purpose

Get request from external process.

Description

This routine provides a mechanism for an application to receive requests from the PRT Data Collector Resident Process (PRT_DC). For example, a PRT_DC can make a request to an external process for Item Tracking Data related to a specific Serialized Item. This API allows the application process to access the request issued by the PRT process.

There are currently two requests a process may receive from PRT:

- Request for Item Tracking Data - the manifest constant for this request type is **PRT_ITEM_REQ**.
- Request for instructions on placing a HOLD status on a list of serialized Items - the manifest constant for this request type is **PRT_ITEM_HOLD_REQ**.

If a Data Collector process which the application is configured to receive requests from terminates (or some other error occurs) the application will be notified by receiving a message with a request type of **PRT_ERROR_RESP**. In this case the service id of the associated Data Collector process will be placed in the `prt_svc_id` field of the **PRT_REQ_INFO** structure.

When `prt_api_get_req` is called, it empties the datagram receive buffer of all messages that have arrived since the last call to `prt_api_get_req` (or since the process registered as an external process) via successive reads, until a read is posted and the event flag remains low. The messages which are read are placed into one of two lists, one containing errors messages, and the other containing valid requests. If there are any messages in the error message list, these are always returned before the valid requests.

Syntax

```
int prt_api_get_req (req_type, req_info, retstat)
COR_U2             *req_type;
PRT_REQ_INFO_PTR  req_info;
COR_STATUS        *retstat;
```

Data Structures

```
typedef struct
{
  char      prt_svc_id[SERVICE_ID_LEN + 1] - PRT service id
  char      group_id[PRT_REGION_ID_LEN + 1] - group id
  char      item_id[PRT_ITEM_ID_LEN + 1] - item id
  char      reference_id[PRT_ITEM_ID_LEN + 1] - reference id
  COR_BOOLEAN ext_hold_active - TRUE if HOLD active
} PRT_REQ_INFO, *PRT_REQ_INFO_PTR;
```

Arguments

Argument	Description
Input	
None	
Output	
*req_type	Request type
req_info	Information contained in request
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_FAILURE or COR_WARNING.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

If there are no requests pending, the function returns COR_WARNING with the COR_STATUS err_code set to PRTI_NOTHING_PENDING.

Error Codes

PRTI_NOTHING_PENDING	No requests pending (Warning)
IPC_ERR_PARTDEAD	Partner Dead (Warning)
PRTI_UNKNOWN_INIT_SRC	MF_INIT segment received from unknown service (Warning)
PRTI_INIT_MSG_RCVD	Init segment received from Data Collector (Warning)
PRTI_UNEXPECTED_SEG	Unexpected segment type (Warning)
PRTI_EF_CHECK_ERR	Error checking event flag (Failure)
PRTI_BAD_MSG_MSG	Error in segment (Failure)

PRTI_READ_PORT_ERR	Error in <code>ipc_read_port</code> (Failure)
--------------------	---

prt_api_group_hold

Name

`prt_api_group_hold`

Purpose

Activate/deactivate hold specification for a tracking group.

Description

This routine provides a mechanism to activate or deactivate a HOLD specification for a Tracking Group. The calling routine receives back a status indicating the success or failure of the request.

Syntax

```
int prt_api_group_hold (group_id, ext_hold_reason,
                       act_hold_flag, comment, user_or_svc_id,
                       retstat)
char   group_id[PRT_REGION_ID_LEN + 1];
char   ext_hold_reason[SC_DESCRIPTION_LEN + 1];
COR_BOOLEAN act_hold_flag;
char   comment[SC_DESCRIPTION_LEN + 1];
char   user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	
group_id	Unique identifier of tracking group (required)
ext_hold_reason	Text describing the reason for placing holds on Tracking Regions in this Tracking Group (required for activate, does not apply to deactivate)
act_hold_flag	If set to TRUE, the Hold Specification is activated. If set to FALSE, the Hold Specification is Deactivated. (required)
Comment	Comment to be recorded in PRT history log file. (optional)
user_or_svc_id	User or service identifier (optional)

Argument	Description
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING, or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

Error Codes

PRTC_NO_SUCH_GROUP	No such group configured (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_EXT_HOLD_NO_REASON	Group hold specified without reason (Warning)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)

Plus all error codes returnable by `prt_api_send_cmd`.

`prt_api_init`

Name

`prt_api_init`

Purpose

Initialize the PRT_API interface.

Description

The **PRT_API** initialization routine is responsible for initializing the interface between an application process and the **PRT_API** utilities. `prt_api_init` must be called by an application before any other **PRT_API** functions. If an application process wishes to be an interested process to PRT it must instead call one of the PRT interested process initialization routines `prt_api_register_intproc` or `prt_api_register_dyn_intproc`. If an application process wants to access the standard **PRT_API** routines as well as sign up as an interested PRT process it must call both **PRT_API** initialization routines.

Syntax

```
int prt_api_init (event_flag, rcv_req_flag, prt_system, cimp_sys, retstat)
COR_U4 event_flag;
COR_BOOLEAN rcv_req_flag;
Char *prt_system;
Char *cimp_sys;
COR_STATUS *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	
event_flag	Message receipt event flag; not used if rcv_req_flag is FALSE
rcv_req_flag	Receive requests flag; if TRUE, this indicates the application will receive requests from PRT Data Collector processes for item tracking data and serialized item hold information.
prt_system	Name of the PRT data server to connect to. This parameter allows you to connect to different Production Tracking systems with the same API. You may also specify the prefix of all the data servers you wish to connect to. This allows the API to connect to multiple servers.
cimp_sys	Name of the project or node where the PRT data collector(s) reside.
Output	

* retstat	Pointer to COR_STATUS structure
------------------	---------------------------------

Return Value

Either **COR_SUCCESS** or **COR_FAILURE**.

If the function returns **COR_FAILURE** additional error information can be found in the **COR_STATUS** structure.

Error Codes

PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)
PRTI_BAD_MSG	Error in segment (Failure)
PRTI_FILE_OPEN_ERR	Error opening one of the configuration files (Failure)
PRTI_EMPTY_FILE	Configuration file has no records (Failure)

PRTI_FILE_READ_ERR	Error reading configuration file (Failure)
PRTI_REG_DEF_MISSING	Undefined Region in file (Failure)
PRTI_GRP_DEF_MISSING	Undefined Group in file (Failure)
PRTI_REGION_NOT_IN_GROUP	Region does not belong to any group (Failure)
PRTI_GROUP_HAS_NO_REGIONS	Group does not 'own' any regions (Failure)
PRTI_CFG_NO_ROUTE	Route has no source and no destination region (Failure)
PRTI_ITEM_TYPE_NDEF	Item type identifier not valid (Failure)
PRTI_ROUTE_NDEF	Non-exist. route in PRT_INV_TYPE (Failure)
PRTI_INV_EXPLICIT_ROUTE	Can't deny explicit item-route (Failure)
PRTI_NO_TYPE_FOR_ROUTE	Denied type not on route (Failure)
PRTI_SVC_DEF_MISSING	Undefined Service in file (Failure)
PRTI_BAD_SVC_NAME	Bad service name (no underscore) (Failure)
PRTI_XLATE_ERR	Error in ipc_xlate getting physical addr (Failure)
PRTI_AUX_NAM_ERR	Error in ipc_aux_nam (Failure)
PRTI_ADD_PORT_ERR	Error in ipc_add_port (Failure)
PRTI_WRITE_PORT_ERR	Error in ipc_write_port (Warn/Fail)

`prt_api_insert_item`

Name

`prt_api_insert_item`

Purpose

Insert item into tracking region queue.

Description

This routine provides a mechanism to insert an Item into the Tracking Region queue. If an Item is to be inserted into the queue at a particular location (other than at the head or tail of the queue), the region location of the item to be inserted ahead of is specified, i.e. specifying **region_loc** = 1, is equivalent to specifying **region_loc** = FIRST. Defined constants are provided as a way of indicating that the item is to be inserted either at the head or at the tail of the queue. The region location can be specified as FIRST (or also as EXIT) to insert the item as the first (head) item in the queue, i.e. this item will be the next item to transition out of the queue. LAST (and ENTRY) are provided to insert the item as the last item at the tail of the queue, BEHIND all other items.

Syntax

```

int prt_api_insert_item ( timestamp, ext_proc_flag, region_id,
                        region_loc, item_id, reference_id,
                        item_type_id, parent_item_id,
                        item_status, ext_hold_active,
                        group_id, int_hold_active,
                        int_hold_reason, num_atts_valid,
                        item_att_list, comment,
                        user_or_svc_id, retstat )

cor_time_t      timestamp;
COR_BOOLEAN    ext_proc_flag;
char           region_id[PRT_REGION_ID_LEN + 1];
COR_I2        region_loc;
char           item_id[PRT_ITEM_ID_LEN + 1];
char           reference_id[PRT_ITEM_ID_LEN + 1];
char           item_type_id[PRT_ITEM_TYPE_ID_LEN + 1];
char           parent_item_id[PRT_ITEM_ID_LEN + 1];
COR_U4        item_status;
COR_BOOLEAN    ext_hold_active;
char           group_id[PRT_REGION_ID_LEN + 1];
COR_BOOLEAN    int_hold_active;
char           *int_hold_reason;
COR_I1 num_atts_valid;
PRT_ITEM_ATT *item_att_list;
char           comment[SC_DESCRIPTION_LEN + 1];
char           user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS    *retstat;

```

Data Structures

See Static Information Requirements

Arguments

Argument	Description
Input	
Serialized Items: Only	
ext_proc_flag	External process flag; if TRUE, get all information from external process. (required)
item_id	Unique identifier of a Serialized Item (optional if reference_id is specified)
reference_id	Secondary identifier of a Serialized Item (optional if item_id is specified)
ext_hold_active	If TRUE, item held due to external hold. (required)
group_id	Group identifier providing reason for hold (required if ext_hold_active = TRUE).

Argument	Description
num_atts_valid	Number of item attributes valid; this specifies the number of array elements in the item attribute list that contain valid data. If item_att_list is a NULL pointer, the function sends a zero to the Data Collector for this argument, regardless of what was actually passed. (required)
item_att_list	Item attribute list. This argument is the pointer to an array of PRT_NUM_ATTRIBUTES elements. If the pointer is NULL, this clearly indicates that no item attributes are being provide for the item being added. If the pointer is non-NULL, however, it must point to an array of exactly PRT_NUM_ATTRIBUTES elements. The num_atts_valid argument indicates how many of these elements contain valid data. (optional)
Non-Serialized Items Only	
parent_item_id	Associated serialized item identifier (optional).
Both	
Timestamp	The time the item was last moved or modified. The timestamp is used to ensure that, when attempting to insert an item, the item(s) currently at the location where the new item is to be inserted in front of have not moved or been modified between the time when information was last obtained for that item(s) and the attempt to insert the new item in front of it them). (One of the fields in the structure which returns data on an item is the last_mod_time field, which indicates the time at which the item was last moved or modified.) If the timestamp argument provided to this function is older than the timestamp associated with the item(s) at the location where the new item is to be inserted in front of, then the 'insert' function is not performed and a warning returned. If the timestamp argument passed to this function is zero, the operation is performed without any timestamp checking.
region_id	Region where the item is to be added to the queue (required).
rregion_loc	Specific location in the region where the Item is to be added (required argument). Specified as FIRST (EXIT), LAST (ENTRY), or a number from 1 to n, where n is the number of items in the queue. (required)
item_type_id	Item type identifier (required).
item_status	Item status (required).
int_hold_active	If TRUE, item held due to internal hold. (required)
int_hold_reason	Comment specifying reason for internal hold (required if int_hold_active = TRUE).
Comment	Comment to be recorded in PRT history log file. (optional)
user_or_svc_id	User or service identifier. This is used for logging purposes and is optional. If this pointer is NULL, the host process id is used. (optional)
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS or COR_FAILURE.

If the function returns COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

PRTC_NON_SER_ID	Non-serialized item has item or ref. id (Failure)
PRTC_SER_NO_ID	Serialized item missing both item & ref. id (Failure)
PRTC_UNKNOWN_REGION	Unknown REGION specified (Failure)
PRTC_BAD_REG_LOC_SPEC	Bad region location specified (Failure)
PRTC_INS_IN_OCC_NON_NORMAL	Attempt to Insert in occupied COMBINE/DISPERSE region (Failure)
PRTC_ITEM_STAMP_TOO_NEW	An affected item has changed since the last display (Failure)
PRTC_ADD_LOC_TOO_BIG	Cannot add item to unoccupied location- inserting (Warning)
PRTC_ADD_MULTIPLE_SERIAL	Cannot have two serialized items at one location (Warning)
PRTC_NO_DETAINMENT_CFG	No detainment region configured for (Warning)
PRTI_ITEM_TYPE_NULL	Item type identifier NULL (Warning)
PRTI_ITEM_TYPE_NDEF	Item type identifier not valid (Warning)
PRTI_NS_EXT_PROC_INVALID	ext_proc_flag set for non-serialized item (Warning)
PRTI_NON_SER_ID	Non-serialized item has item id or reference id (Warning)
PRTI_SER_NO_ID	Serialized item missing both item id and reference id (Warning)
PRTI_SER_PARENT	Parent ID specified for serialized item (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_EXT_HOLD_NON_SER	External hold specified for non-serialized item (Warning)
PRTI_INT_HOLD_NO_REASON	Internal hold specified without reason (Warning)
PRTI_NON_SER_ATTS	Non-serialized item has attribute(s) (Warning)
PRTI_NUM_ATTS_INVALID	Invalid number of item attributes specified (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_INVALID_REG_LOC	Invalid region location (Warning)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)

Plus all error codes returnable by `prt_api_send_cmd`.

`prt_api_intproc_region`

Name

prt_api_intproc_region

Purpose

Receive updates only from specific regions.

Description

This function provides a mechanism for a process to request that only specific region data is required. A flag is passed to specify if the region should be added, replaced, or deleted. The calling routine receives back a status indicating the success or failure of the request.

Syntax

```
int prt_api_intproc_region (region_id, action_code, retstat)
char          *region_id;
COR_U4       action_code;
COR_STATUS   *retstat;
```

Data Structures

None

Arguments

Argument	Description	
Input		
region_id	Region that this call applies to	
action_code	Code to specify what action needs to occur. Valid values are:	
	PRT_FILTER_ADD	Add this region_id to filter list.
	PRT_FILTER_REPLACE	Replace existing filter region_id with this one, if one exists. Otherwise add region_id
	PRT_FILTER_DELETE	Delete this region_id from filter list.
Output		
*retstat	Pointer to COR_STATUS structure	

Return Value

Either COR_SUCCESS or COR_FAILURE.

If the function returns `COR_FAILURE` additional error information can be found in the `COR_STATUS` structure.

Error Codes

<code>IPC_ERR_PARTDEAD</code>	Partner Dead (Warning)
<code>PRTI_UNKNOWN_INIT_SRC</code>	MF_INIT segment received from unknown service (Warning).
<code>PRTI_UNEXPECTED_SEG</code>	Unexpected segment type (Warning)
<code>PRTI_BAD_MSG</code>	Error in segment (Failure)
<code>PRTI_READ_PORT_ERR</code>	Error in <code>ipc_read_port</code> (Failure)
<code>PRTI_EF_CHECK_ERR</code>	Error checking event flag (Failure)

`prt_api_intproc_retry_connect`

Name

`prt_api_intproc_retry_connect`

Purpose

Send INIT segment to all data servers.

Description

This function provides a mechanism for a process to send an INIT segment to all data servers we are interested in. This routine allows the user to easily reestablish the connection to a data server when the connection has been broken.

Syntax

```
COR_BOOLEAN prt_api_intproc_retry_connect ( )
```

Data Structures

None

Arguments

Input

None

Output

None

Return Value

Either TRUE or FALSE.

If the function returns TRUE, then this means that an INIT segment was sent to at least one data server.

Error Codes

None

`prt_api_load_item`

Name

`prt_api_load_item`

Purpose

Load serialized item tracking data.

Description

This PRT_API routine is used to load the answer for a PRT request for Serialized Item Tracking Data into a response buffer. The calling routine receives back a status indicating the success or failure of the request. This function can be called multiple times to load multiple responses into the response buffer. If the buffer becomes full the return status will be unsuccessful and additional error information will be found in the COR_STATUS structure. The PRT_API routine **`prt_api_send_resp`** must be called to send the response buffer to the appropriate PRT process.

Syntax

```
int prt_api_load_item (data_valid_flag, item_id,
                     reference_id, item_type_id, item_status,
                     ext_hold_active, group_id,
                     int_hold_active, int_hold_reason,
                     num_atts_valid, item_att_list, retstat)
COR_BOOLEAN  data_valid_flag;
char         item_id[PRT_ITEM_ID_LEN + 1];
char         reference_id[PRT_ITEM_ID_LEN + 1];
char         item_type_id[PRT_ITEM_TYPE_ID_LEN + 1];
COR_U4      item_status;
COR_BOOLEAN  ext_hold_active;
```

```

char          group_id[PRT_REGION_ID_LEN + 1];
COR_BOOLEAN  int_hold_active;
char          int_hold_reason[SC_DESCRIPTION_LEN + 1];
COR_I1       num_atts_valid;
PRT_ITEM_ATT item_att_list[PRT_NUM_ATTRIBUTES];
COR_STATUS   *retstat;

```

Data Structures

```

typedef struct
{
    char att_name[PRT_ATTRIBUTE_LEN + 1] - item attribute name
    char att_value[PRT_ATTRIBUTE_LEN + 1] - item attribute value
} PRT_ITEM_ATT

```

Arguments

Argument	Description
Input	
data_valid_flag	If TRUE, data requested is being returned (is valid). FALSE indicates the request could not be satisfied. (required)
item_id	Unique identifier of a Serialized Item. (optional if reference_id is specified)
reference_id	Secondary identifier of a Serialized Item. (optional if item_id is specified)
item_type_id	Item type identifier. (required)
item_status	Item status. (required)
ext_hold_active	If TRUE, Serialized item held due to external hold. (optional)
group_id	Identifier of the Tracking Group placing the external hold on the item. (required if ext_hold_active = TRUE)
int_hold_active	If TRUE, item held due to internal hold. (optional)
int_hold_reason	Comment specifying reason for internal hold. (optional)
num_atts_valid	Number of item attributes valid; this specifies the number of array elements in the item attribute list that contain valid data. If item_att_list is a NULL pointer, the function sends a zero to the Data Collector for this argument, regardless of what was actually passed.
item_att_list	Item Attributes pertain to Serialized Items. (required)
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_FAILURE, or COR_WARNING.

If the function returns `COR_WARNING` or `COR_FAILURE`, additional error information can be found in the `COR_STATUS` structure. If no more item information will fit in the datagram buffer, the function returns `COR_WARNING` with the `COR_STATUS` `err_code` set to `PRTI_DG_BUF_FULL`.

Error Codes

<code>PRTI_API_CANT_RCV</code>	PRT Application not initialized to receive asynch messages (Warning)
<code>PRTI_NON_SER_ID</code>	Non-serialized item has item id or reference id (Warning)
<code>PRTI_SER_NO_ID</code>	Serialized item missing both item id and reference id (Warning)
<code>PRTI_ITEM_TYPE_NULL</code>	Item type identifier NULL (Warning)
<code>PRTI_ITEM_TYPE_NDEF</code>	Item type identifier not valid (Warning)
<code>PRTI_GRP_ID_NULL</code>	Group identifier NULL (Warning)
<code>PRTI_GRP_ID_NDEF</code>	Group identifier not valid (Warning)
<code>PRTI_EXT_HOLD_NO_GRP</code>	External hold specified without Group ID (Warning)
<code>PRTI_INT_HOLD_NO_REASON</code>	Internal hold specified without reason (Warning)
<code>PRTI_NUM_ATTRS_INVALID</code>	Invalid number of item attributes specified (Warning)
<code>PRTI_DG_BUF_FULL</code>	Datagram buffer full (Warning)
<code>PRTI_CRESEG_ERR</code>	Error creating segment (Failure)

`prt_api_load_item_hold`

Name

`prt_api_load_item_hold`

Purpose

Load serialized item hold data.

Description

This `PRT_API` routine is used to load the answer for a PRT request for Serialized Item Hold Data into a response buffer. The calling routine receives back a status indicating the success or failure of the request. This function can be called multiple times to load multiple responses into the response buffer. If the buffer becomes full the return status will be unsuccessful and additional error information will be found in the `COR_STATUS` structure. The `PRT_API` routine `prt_api_send_resp` must be called to send the response buffer to the appropriate PRT process.

Syntax

```
int prt_api_load_item_hold (data_valid_flag, item_id, reference_id,
                           ext_hold_active, group_id, retstat)
COR_BOOLEAN
char      item_id[PRT_ITEM_ID_LEN + 1];
char      reference_id[PRT_ITEM_ID_LEN + 1];
COR_BOOLEAN ext_hold_active;
char      group_id[PRT_REGION_ID_LEN + 1];
COR_STATUS *retstat;
```

Data Structures

See Static Information Requirements

Arguments

Argument	Description
Input	
data_valid_flag	If TRUE, data requested is being returned (is valid). FALSE indicates the request could not be satisfied. (required)
item_id	Unique identifier of a Serialized Item. (optional if reference id specified)
reference_id	Secondary identifier of a Serialized Item. (optional if item id specified)
ext_hold_active	If TRUE, Serialized Item held due to external hold.
group_id	Tracking Group applying the hold. (required if ext_hold_active = TRUE)
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_FAILURE, or COR_WARNING.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure. If no more item information will fit in the datagram buffer, the function returns COR_WARNING with the COR_STATUS err_code set to PRTI_DG_BUF_FULL.

Error Codes

PRTI_DG_BUF_FULL	Datagram buffer full (Warning)
PRTI_API_CANT_RCV	PRT Application not initialized to receive asynch messages (Warning)
PRTI_NON_SER_ID	Non-serialized item has item id or reference id (Warning)

PRTI_SER_NO_ID	Serialized item missing both item id and reference id (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_EXT_HOLD_NO_GRP	External hold specified without Group ID (Warning)
PRTI_CRESEG_ERR	Error creating segment (Failure)

`prt_api_locate_item`

Name

`prt_api_locate_item`

Purpose

Locate serialized item.

Description

This API provides a mechanism for locating a Serialized Item within all configured PRT services, a specific PRT service, a specific Tracking Group or a specific Tracking Region. If the Item is located, the Item Tracking Data is returned to the calling routine with a return status of `COR_SUCCESS`. If the item was located in more than one tracking region, the first location is returned in the initial call. Successive calls to this routine return the item's additional locations. When the list of locations is complete, the routine returns `COR_WARNING` with the additional information in the `COR_STATUS` structure. If the Item was not located the calling routine receives back a status indicating the request failed.

Syntax

```
int prt_api_locate_item (prt_svc_id, group_id, region_id,
                        item_id, reference_id, item_data,
                        more_data, retstat)
char                    prt_svc_id[SERVICE_ID_LEN + 1];
char                    group_id[PRT_REGION_ID_LEN + 1];
char                    region_id[PRT_REGION_ID_LEN + 1];
char                    item_id[PRT_ITEM_ID_LEN + 1];
char                    reference_id[PRT_ITEM_ID_LEN + 1];
PRT_ITEM_STRUCT_PTR    item_data;
COR_BOOLEAN             *more_data;
COR_STATUS              *retstat;
```

Data Structures

```
typedef struct {
    cor_time_t - reg_entry_time
```

```

    cor_time_t - last_mod_time
    char region_id[PRT_REGION_ID_LEN + 1] - identifier of the region
    COR_I2 region_loc - specific location in the region
    char item_id[PRT_ITEM_ID_LEN + 1] - unique identifier of a
        Serialized Item.
    char reference_id[PRT_ITEM_ID_LEN + 1] - secondary identifier
        of a Serialized Item.
    char item_type_id[PRT_ITEM_TYPE_ID_LEN + 1] - item type
        identifier.
char parent_item_id[PRT_ITEM_ID_LEN + 1] - pertains only to
        Non-serialized Items.
    COR_U4 item_status - item status.
    COR_BOOLEAN int_hold_active - if TRUE, item held due to
        internal hold.
char int_hold_reason[SC_DESCRIPTION_LEN + 1] - Comment specifying
        reason for internal
        hold.
    COR_BOOLEAN ext_hold_active - if TRUE, item held due to
        external hold.
char group_id[PRT_REGION_ID_LEN + 1] - identifier of the group
COR_I1 num_atts_valid - Number of attributes in the list
PRT_ITEM_ATT item_att_list[PRT_NUM_ATTRIBUTES] - Item Attributes
        pertain to
        Serialized
        Items.
} PRT_ITEM_STRUCT, *PRT_ITEM_STRUCT_PTR

```

Arguments

Argument	Description
Input	
prt_svc_id	PRT service identifier (optional)
group_id	Unique identifier of tracking group (optional)
region_id	Unique identifier of tracking region (optional)
item_id	Unique identifier of serialized item (optional if reference_id specified)
reference_id	Secondary identifier of serialized item (optional if item_id specified)
more_data	If this flag is set to true by the caller, the function attempts to return an item found by a previous call to this function. If this flag is set to FALSE, any items found by a previous call to this function are deleted and new request(s) are made to the specified service(s).
Output	
PRT_ITEM_STRUCT_PTR item_data	Serialized Item Tracking Data
more_data	If only a single item was found, this flag is set to FALSE. If more than one item was found, this flag is set to TRUE.

Argument	Description
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING, or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

Error Codes

PRTS_NO_GLOBAL_XREF	No XREF global section available (Warning)
PRTS_REGION_NOT_DEFINED	Region not configured (Warning)
PRTS_NO_GLOBAL_REGION	No global section available for Region (Warning)
PRTI_BAD_POINTER	Pointer to data structure is NULL (Warning)
PRTI_NO_MORE_ITEMS	No more items available (Warning)
PRTI_SVC_ID_NULL	PRT service identifier NULL (Warning)
PRTI_SVC_ID_NDEF	PRT service identifier not valid (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_NON_SER_ID	Non-serialized item has item_id or reference_id (Warning)
PRTI_SER_NO_ID	Serialized item missing both item_id and reference_id (Warning)
PRTI_REG_NOT_IN_GRP	Region not in group specified (Warning)
PRTI_REG_NOT_IN_SVC	Region not managed by service specified (Warning)
PRTI_GRP_NOT_IN_SVC	Group ID inconsistent with service specified (Warning)
PRTI_UNEXPECTED_SEG	Unexpected segment type (Warning)
PRTI_ITEM_NOT_FOUND_INC_SRCH	Item not found: some service(s) down (Warning)
PRTI_ITEM_NOT_FOUND	Item not found (Warning)
PRTI_WRITE_PORT_ERR	Error in ipc_write_port (Warn/Fail)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)
PRTI_BAD_MSG	Error in segment (Failure)

PRTI_BAD_SERVER_RESPONSE	PRT Data Server returned bad/error response (Failure)
PRTI_UNKNOWN_RESPONDER	Response received from unknown service (Failure)

prt_api_modify_item

Name

`prt_api_modify_item`

Purpose

Modify item tracking data.

Description

This routine provides a mechanism to modify Item Tracking Data, associate an Item Type to an unknown Item, associate a Reference ID to an Item ID and associate two Items. The calling routine receives back a status indicating the success or failure of the request. This routine does not allow for the modification of Item attributes, Item attributes are modified using the **prt_api_modify_item_att** routine.

Syntax

```
int prt_api_modify_item (timestamp, modify_bitmask, status_bitmask,
                        region_id, region_loc, item_id,
                        reference_id, item_type_id,
                        parent_item_id, item_status,
                        ext_hold_active, group_id,
                        int_hold_active, int_hold_reason,
                        comment, user_or_svc_id, retstat)
COR_TIME_T      timestamp;
COR_U2         modify_bitmask;
COR_U4         status_bitmask;
char           region_id[PRT_REGION_ID_LEN + 1];
COR_I2         region_loc;
char           item_id[PRT_ITEM_ID_LEN + 1];
char           reference_id[PRT_ITEM_ID_LEN + 1];
char           item_type_id[PRT_ITEM_TYPE_ID_LEN + 1];
char           parent_item_id[PRT_ITEM_ID_LEN + 1];
COR_U4         item_status;
COR_BOOLEAN    ext_hold_active;
char           group_id[PRT_REGION_ID_LEN + 1];
COR_BOOLEAN    int_hold_active;
char           int_hold_reason[SC_DESCRIPTION_LEN + 1];
char           comment[SC_DESCRIPTION_LEN + 1];
char           user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS     *retstat;
```

Data Structures

None

Arguments

Argument	Description																
Input	(See SYNTAX for proper order)																
Serialized Items Only																	
item_id	Unique identifier of a Serialized Item (optional if reference id specified)																
reference_id	Secondary identifier of a Serialized Item (optional if item id specified)																
ext_hold_active	If TRUE, item held due to external hold (optional)																
group_id	Group identifier providing reason for hold (required if ext_hold_active = TRUE)																
Non-Serialized Items Only																	
parent_item_id	Item ID of associated serialized Item (optional)																
Both																	
Timestamp	The time the item was last moved or modified. The timestamp is used to ensure that, when attempting to modify an item, the item has not changed (or moved) between the time when information was last obtained for that item and the attempt to modify it. (One of the fields in the structure that returns data on an item is the last_mod_time field, which indicates the time at which the item was last moved or modified.) If the timestamp argument provided to this function is older than the timestamp associated with the item at the time of the attempted modification, then the modification is not performed and a warning returned. If the timestamp argument passed to this function is zero, the operation is performed without any timestamp checking.																
modify_bitmask	Indicates fields that have been modified (required). If a bit in the bitmask is set, the corresponding argument provided to this function is required. If a bit is clear, the corresponding argument is not accessed in any way.																
	The following fields are potentially modifiable and have a corresponding bit in the modify_bitmask .																
	<table border="1"> <thead> <tr> <th>Field Name</th> <th>Bit Name</th> </tr> </thead> <tbody> <tr> <td>item_status</td> <td>ITEM_STATUS_BIT</td> </tr> <tr> <td>status_bitmask</td> <td>ITEM_STATUS_BIT</td> </tr> <tr> <td>item_id</td> <td>ITEM_ID_MOD_BIT</td> </tr> <tr> <td>reference_id</td> <td>REFERENCE_ID_MOD_BIT</td> </tr> <tr> <td>item_type_id</td> <td>ITEM_TYPE_ID_BIT</td> </tr> <tr> <td>parent_item_id</td> <td>PARENT_ITEM_ID_BIT</td> </tr> <tr> <td>ext_hold_active</td> <td>EXT_HOLD_BIT</td> </tr> </tbody> </table>	Field Name	Bit Name	item_status	ITEM_STATUS_BIT	status_bitmask	ITEM_STATUS_BIT	item_id	ITEM_ID_MOD_BIT	reference_id	REFERENCE_ID_MOD_BIT	item_type_id	ITEM_TYPE_ID_BIT	parent_item_id	PARENT_ITEM_ID_BIT	ext_hold_active	EXT_HOLD_BIT
Field Name	Bit Name																
item_status	ITEM_STATUS_BIT																
status_bitmask	ITEM_STATUS_BIT																
item_id	ITEM_ID_MOD_BIT																
reference_id	REFERENCE_ID_MOD_BIT																
item_type_id	ITEM_TYPE_ID_BIT																
parent_item_id	PARENT_ITEM_ID_BIT																
ext_hold_active	EXT_HOLD_BIT																

Argument	Description
group_id	EXT_HOLD_BIT
int_hold_active	INT_HOLD_BIT
int_hold_reason	INT_HOLD_BIT
	Macros can be used to set the various bits in modify_bitmask to indicate which fields are being modified.
	Note: The item_id and reference_id, if both are specified, are used to find the item unless either the ITEM_ID_MOD_BIT or REFERENCE_ID_MOD_BIT is set (both cannot be set). If neither is set, the item you are looking to modify will only be found if both the item_id and the reference_id you supply match that of an item already in the production tracking system. The item_ids need not match, however, if the ITEM_ID_MOD_BIT is set, and similarly, the reference_ids need not match if the REFERENCE_ID_MOD_BIT is set.
status_bitmask	Indicates fields in item status that have been modified. (optional)
region_id	Region where item last resided (required)
region_loc	Location in the region where the item resides (required) For non-serialized items, a specific region location must be specified. For serialized items, the constant ALL_LOCATIONS may be used. This indicates that the item may reside at any location in the region.
item_type_id	Item type identifier (required for non-serialized item, optional for serialized items)
item_status	Item status (optional)
Comment	Comment to be recorded in prt history log file. (optional)
user_or_svc_id	User or service identifier (optional)
int_hold_active	If TRUE, item held due to internal hold (optional)
int_hold_reason	Comment specifying reason for internal hold. (required for activate, does not apply to deactivate)
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING, or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

All error codes returnable by **prt_api_modify_item_all** (except PRTI_NUM_ATTS_INVALID)

prt_api_modify_item_all

Name

prt_api_modify_item_all

Purpose

Modify item tracking data (all).

Description

This routine provides a mechanism to modify all information fields associated with an item, both those modifiable by the **prt_api_modify_item** function and (for a serialized item) the item's attributes, also. For non-serialized items, this function does nothing more than the `prt_api_modify_item` function. The calling routine receives back a status indicating the success or failure of the request.

Syntax

```

int prt_api_modify_item_all( timestamp, modify_bitmask,
                             status_bitmask, region_id,
                             region_loc, item_id, reference_id,
                             item_type_id, parent_item_id,
                             item_status, ext_hold_active,
                             group_id, int_hold_active,
                             int_hold_reason, num_atts_valid,
                             item_att_list, comment,
                             user_or_svc_id, retstat)
cor_time_t      timestamp;
COR_U2          modify_bitmask;
COR_U4          status_bitmask;
char            region_id[PRT_REGION_ID_LEN + 1];
COR_I2          region_loc;
char            item_id[PRT_ITEM_ID_LEN + 1];
char            reference_id[PRT_ITEM_ID_LEN + 1];
char            item_type_id[PRT_ITEM_TYPE_ID_LEN + 1];
char            parent_item_id[PRT_ITEM_ID_LEN + 1];
COR_U4          item_status;
COR_BOOLEAN    ext_hold_active;
char            group_id[PRT_REGION_ID_LEN + 1];
COR_BOOLEAN    int_hold_active;
char            int_hold_reason[SC_DESCRIPTION_LEN + 1];
COR_I1         num_atts_valid;
PRT_ITEM_ATT   *item_att_list;
char            comment[SC_DESCRIPTION_LEN + 1];
char            user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS     *retstat;

```

Data Structures

None

Input (See SYNTAX for proper order)**Arguments**

Argument	Description	
Serialized Items Only		
item_id	Unique identifier of a Serialized Item (optional if reference id specified)	
reference_id	Secondary identifier of a Serialized Item (optional if item id specified).	
ext_hold_active	If TRUE, item held due to external hold (optional)	
group_id	Group identifier providing reason for hold (required if ext_hold_active = TRUE)	
num_atts_valid	Number of item attributes valid; this specifies the number of array elements in the item attribute list that contain valid data.	
item_att_list	Item Attributes pertain to Serialized Items.	
Non-Serialized Items Only		
parent_item_id	Item ID of associated serialized Item (optional)	
Both		
timestamp	The time the item was last moved or modified. The timestamp is used to ensure that, when attempting to modify an item, the item has not changed (or moved) between the time when information was last obtained for that item and the attempt to modify it. (One of the fields in the structure which returns data on an item is the last_mod_time field, which indicates the time at which the item was last moved or modified.) If the timestamp argument provided to this function is older than the timestamp associated with the item at the time of the attempted modification, then the modification is not performed and a warning returned. If the timestamp argument passed to this function is zero, the operation is performed without any timestamp checking.	
modify_bitmask	Indicates fields that have been modified (required). If a bit in the bitmask is set, the corresponding argument provided to this function is required. If a bit is clear, the corresponding argument is not accessed in any way. The following fields are potentially modifiable and have a corresponding bit in the modify_bitmask .	
	Field Name	Bit Name
	item_status	ITEM_STATUS_BIT
	status_bitmask	ITEM_STATUS_BIT
	item_id	ITEM_ID_MOD_BIT
	reference_id	REFERENCE_ID_MOD_BIT
	item_type_id	ITEM_TYPE_ID_BIT
	parent_item_id	PARENT_ITEM_ID_BIT
	ext_hold_active	EXT_HOLD_BIT

Argument	Description
group_id	EXT_HOLD_BIT
int_hold_active	INT_HOLD_BIT
int_hold_reason	INT_HOLD_BIT
num_atts_valid	ATTS_VALID_BIT
item_att_list	ATTS_VALID_BIT
	Macros can be used to set the various bits in modify_bitmask to indicate which fields are being modified. Note that the item_id and reference_id , if both are specified, are used to find the item unless either the ITEM_ID_MOD_BIT or REFERENCE_ID_MOD_BIT is set (both cannot be set). If neither is set, the item you are looking to modify will only be found if both the item_id and the reference_id you supply match that of an item already in the production tracking system. The item_ids need not match, however, if the ITEM_ID_MOD_BIT is set, and similarly, the reference_ids need not match if the REFERENCE_ID_MOD_BIT is set.
status_bitmask	Indicates fields in item status that have been modified (optional)
region_id	Region where item last resided (required)
region_loc	Location in the region where the item resides (required) For non-serialized items, a specific region location must be specified. For serialized items, the constant ALL_LOCATIONS may be used. This indicates that the item may reside at any location in the region.
item_type_id	Item type identifier (required for non-serialized item, optional for serialized items)
item_status	Item status (optional)
comment	Comment to be recorded in PRT history log file. (optional)
user_or_svc_id	User or service identifier (optional)
int_hold_active	If TRUE, item held due to internal hold (optional)
int_hold_reason	Comment specifying reason for internal hold. (required for activate, does not apply to deactivate).
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING, or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

PRTC_INVALID_ITEM_TYPE	Invalid item type specified (Warning)
PRTC_CANT_MATCH_BOTH	Item/XREF entry mismatch (Warning)
PRTC_ITEM_NOT_FOUND	Item ID not found (Warning)

PRTC_REF_NOT_FOUND	Reference ID not found (Warning)
PRTC_BAD_REG_SPEC	Bad region specified (Warning)
PRTC_BAD_REG_LOC_SPEC	Bad region location specified (Warning)
PRTC_ITEM_STAMP_TOO_NEW	An affected item has changed since the last display (Warning)
PRTC_NO_ITEM_AT_LOC	Specified item not found specified location (Warning)
PRTI_CANT_MOD_ITEM_REF_IDS	Can't modify both item & ref. ids (Warning)
PRTI_CANT_MOD_ITEM_WOUT_REF	Must specify ref. id to modify item id (Warning)
PRTI_ITEM_TYPE_NULL	Item type identifier NULL (Warning)
PRTI_ITEM_TYPE_NDEF	Item type identifier not valid (Warning)
PRTI_NON_SER_ID	Non-serialized item has item id or reference id (Warning)
PRTI_SER_NO_ID	Serialized item missing both item id and reference id (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_EXT_HOLD_NON_SER	External hold specified for non-serialized item (Warning)
PRTI_INT_HOLD_NO_REASON	Internal hold specified without reason (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_INVALID_REG_LOC	Invalid region location (Warning)
PRTI_ITEM_TYPE_NULL	Item type identifier NULL (Warning)
PRTI_ITEM_TYPE_NDEF	Item type identifier not valid (Warning)
PRTI_SER_PARENT	Parent ID specified for serialized item (Warning)
PRTI_NUM_ATTRS_INVALID	Invalid number of item attributes specified (Warning)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)

Plus all error codes returnable by `prt_api_send_cmd`.

`prt_api_modify_item_att`

Name

`prt_api_modify_item_att`

Purpose

Modify item tracking data attributes.

Description

This routine provides a mechanism to modify Serialized Item attributes. The calling routine receives back a status indicating the success or failure of the request.

Syntax

```
int prt_api_modify_item_att ( timestamp, region_id, item_id,
                             reference_id, num_atts_valid,
                             item_att_list, comment,
                             user_or_svc_id, retstat)
cor_time_t      timestamp;
char            region_id[PRT_REGION_ID_LEN + 1];
char            item_id[PRT_ITEM_ID_LEN + 1];
char            reference_id[PRT_ITEM_ID_LEN + 1];
COR_I1         num_atts_valid;
PRT_ITEM_ATT   item_att_list[PRT_NUM_ATTRIBUTES];
char            comment[SC_DESCRIPTION_LEN + 1];
char            user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS     *retstat;
```

Data Structures

```
typedef struct
{
    char att_name[PRT_ATTRIBUTE_LEN + 1] - item attribute name
    char att_value[PRT_ATTRIBUTE_LEN + 1] - item attribute value
} PRT_ITEM_ATT
```

Arguments

Argument	Description
Input	
Serialized Items Only	
Timestamp	The time the item was last moved or modified. The timestamp is used to ensure that, when attempting to modify an item's attributes, the item has not changed (or moved) between the time when information was last obtained for that item and the attempt to modify it. (One of the fields in the structure which returns data on an item is the last_mod_time field, which indicates the time at which the item was last moved or modified.) If the timestamp argument provided to this function is older than the timestamp associated with the item at the time of the attempted modification, then the modification is not performed and a warning returned. If the timestamp argument passed to this function is zero, the operation is performed without any timestamp checking.

Argument	Description
region_id	Region where item last resided (required)
item_id	Unique identifier of a Serialized Item (optional if reference id specified)
reference_id	Secondary identifier of a Serialized Item (optional if item id specified)
num_atts_valid	Number of item attributes valid; this specifies the number of array elements in the item attribute list that contain valid data.
item_att_list	Item Attributes pertain to Serialized Items. (required)
comment	Comment to be recorded in PRT history log file. (optional)
user_or_svc_id	User or service identifier (optional)
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING, or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

Error Codes

PRTI_NUM_ATTS_INVALID	Invalid number of item attributes specified (Warning)
PRTC_INVALID_ITEM_TYPE	Invalid item type specified (Warning)
PRTC_CANT_MATCH_BOTH	Item/XREF entry mismatch (Warning)
PRTC_ITEM_NOT_FOUND	Item ID not found (Warning)
PRTC_REF_NOT_FOUND	Reference ID not found (Warning)
PRTC_BAD_REG_SPEC	Bad region specified (Warning)
PRTC_ITEM_STAMP_TOO_NEW	An affected item has changed since the last display (Warning)
PRTI_ITEM_TYPE_NDEF	Item type identifier not valid (Warning)
PRTI_SER_NO_ID	Serialized item missing both item id and reference id (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_ITEM_TYPE_NULL	Item type identifier NULL (Warning)
PRTI_ITEM_TYPE_NDEF	Item type identifier not valid (Warning)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)

PRTI_CRESEG_ERR	Error creating segment (Failure)
-----------------	----------------------------------

prt_api_modify_region

Name

prt_api_modify_region

Purpose

Modify tracking region data.

Description

This routine provides a mechanism to modify Tracking Region Data; specifically, the region status.

Syntax

```
int prt_api_modify_region ( status_bitmask, region_id,
                           region_status, comment,
                           user_or_svc_id, retstat )
COR_U4      status_bitmask;
char        region_id;[PRT_REGION_ID_LEN + 1]
COR_U4      region_status;
char        comment[SC_DESCRIPTION_LEN + 1];
char        user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS *retstat;
```

Data Structures

See Static Information Requirements

Arguments

Argument	Description
Input	
status_bitmask	Indicates the bit positions in the region_status that are to be modified. (required)
region_id	Region identifier (required)
region_status	Tracking Region status code (e.g. IN-LOCKED, OUT-LOCKED, FULL, OUT-OF-SEQ). (optional)
comment	Comment to be recorded in PRT history log file (optional).
user_or_svc_id	User or service identifier. This is used for logging purposes and is optional. If this pointer is NULL, the host process id is used. (optional)
Output	

Argument	Description
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS or COR_FAILURE.

If the function returns COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

PRTC_BAD_REG_SPEC	Bad region specified (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)

Plus all error codes returnable by `prt_api_send_cmd`.

`prt_api_move_item`

Name

`prt_api_move_item`

Purpose

Move item to location in tracking region

Description

This routine provides a mechanism for moving an Item from the Tracking Region where it currently resides to a new Tracking Region. It also permits the movement of an individual Item from one location to another within the same Tracking Region. The semantics of this call specify that if neither the **item_id**, **reference_id**, or **item_type_id** is specified, all items are moved from the specified source to the specified destination region. If neither the **item_id** or **reference_id** is specified, but the **item_type_id** is, a single item of the specified type is moved from the source to the destination region. If we're moving a serialized item, the source location need not be given. The specification of the destination region location argument is the same as that for the `prt_api_add_item` function if the **insert_flag** passed in is FALSE, and is the same as that for the `prt_api_insert_item` function if the **insert_flag** passed in is TRUE.

Syntax

```

int prt_api_move_item ( src_reg_id, dest_reg_id, src_reg_loc,
                       dest_reg_loc, insert_flag, item_id,
                       reference_id, item_type_id, comment,
                       user_or_svc_id, retstat)
char      src_reg_id[PRT_REGION_ID_LEN + 1];
char      dest_reg_id[PRT_REGION_ID_LEN + 1];
COR_I2    src_reg_loc;
COR_I2    dest_reg_loc;
COR_BOOLEAN insert_flag;
char      item_id[PRT_ITEM_ID_LEN + 1];
char      reference_id[PRT_ITEM_ID_LEN + 1];
char      item_type_id[PRT_ITEM_TYPE_ID_LEN + 1];
char      comment[SC_DESCRIPTION_LEN + 1];
char      user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS *retstat;

```

Data Structures

See Static Information Requirements

Arguments

Argument	Description
Input	
Serialized Items Only	
item_id	Unique identifier of a Serialized Item (optional if reference_id is specified).
reference_id	Secondary identifier of a Serialized Item (optional if item_id is specified).
Both Serialized and Non-Serialized Items	
src_reg_id	Source region id; region the item is moving out of (required).
dest_reg_id	Destination region id; region the item is moving into (required).
src_reg_loc	Source region location; sequence number of the item in the source tracking region. Specified as FIRST (EXIT), LAST (ENTRY), or a number from 1 to n, where n is the number of items in the queue. required for non-serialized items, optional for serialized items).
dest_reg_loc	Destination region location; sequence number of the item in the destination tracking region. Specified as FIRST (EXIT), LAST (ENTRY), or a number from 1 to n, where n is the number of items in the queue. (required).

Argument	Description
insert_flag	If TRUE, the moved item is to be inserted in the Destination Tracking Region queue before the item located at the specified destination region location, unless the manifest constant LAST (ENTRY) is specified, in which case the item is inserted after behind) the last item currently present in the region. (required).
item_type_id	Item type identifier (required for non-serialized items, optional for serialized items).
comment	Comment to be recorded in PRT history log file (optional).
user_or_svc_id	User or service identifier. This is used for logging purposes and is optional. If this pointer is NULL, the host process id is used. (optional)
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS or COR_FAILURE.

If the function returns COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

PRTC_NON_SER_ID	Non-serialized item has item or ref. id (Failure)
PRTC_SER_NO_ID	Serialized item missing both item & ref. id (Failure)
PRTC_UNKNOWN_REGION	Unknown REGION specified (Failure)
PRTC_BAD_REG_LOC_SPEC	Bad region location specified (Failure)
PRTC_ITEM_STAMP_TOO_NEW	An affected item has changed since the last display (Failure)
PRTC_ADD_LOC_TOO_BIG	Cannot add item to unoccupied location- inserting (Warning)
PRTC_ADD_MULTIPLE_SERIAL	Cannot have two serialized items at one location (Warning)
PRTC_REGION_NOT_LOCKED	Lockable region %s not locked; cannot perform operation (Warning)
PRTI_ITEM_TYPE_NULL	Item type identifier NULL (Warning)
PRTI_ITEM_TYPE_NDEF	Item type identifier not valid (Warning)
PRTI_NON_SER_ID	Non-serialized item has item id or reference id (Warning)
PRTI_SER_NO_ID	Serialized item missing both item id and reference id (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_INVALID_REG_LOC	Invalid region location (Warning)
PRTI_ITEM_TYPE_NULL	Item type identifier NULL (Warning)

PRTI_INVALID_DEST	Destination region is invalid for item of this type (Warning)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)

Plus all error codes returnable by `prt_api_send_cmd`.

`prt_api_move_item_to_detain`

Name

`prt_api_move_item_to_detain`

Purpose

Move item to detainment region

Description

This routine provides a mechanism for moving an Item from the Tracking Region where it currently resides to the detainment region of that region. This function checks that a detainment region exists for the source region and then calls `prt_api_move_item` to actually complete the work.

Syntax

```
int prt_api_move_item_to_detain ( src_reg_id, src_reg_loc,
                                item_id, reference_id, item_type_id,
                                comment, user_or_svc_id, retstat)
char      src_reg_id[PRT_REGION_ID_LEN + 1];
COR_I2    src_reg_loc;
char      item_id[PRT_ITEM_ID_LEN + 1];
char      reference_id[PRT_ITEM_ID_LEN + 1];
char      item_type_id[PRT_ITEM_TYPE_ID_LEN + 1];
char      comment[SC_DESCRIPTION_LEN + 1];
char      user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS *retstat;
```

Data Structures

See Static Information Requirements

Arguments

Argument	Description
Serialized Items Only	

Argument	Description
item_id	Unique identifier of a Serialized Item (optional if reference_id is specified).
reference_id	Secondary identifier of a Serialized Item (optional if item_id is specified).
Both Serialized and Non-Serialized Items	
src_reg_id	Source region id; region the item is moving out of (required).
src_reg_loc	Source region location; sequence number of the item in the source tracking region. Specified as FIRST (EXIT), LAST (ENTRY), or a number from 1 to n, where n is the number of items in the queue. required for non-serialized items, optional for serialized items).
item_type_id	Item type identifier (required for non-serialized items, optional for serialized items).
Comment	Comment to be recorded in PRT history log file (optional).
user_or_svc_id	User or service identifier. This is used for logging purposes and is optional. If this pointer is NULL, the host process id is used. (optional)
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS or COR_FAILURE.

If the function returns COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

PRTC_NON_SER_ID	Non-serialized item has item or ref. id (Failure)
PRTC_SER_NO_ID	Serialized item missing both item & ref. id (Failure)
PRTC_UNKNOWN_REGION	Unknown REGION specified (Failure)
PRTC_BAD_REG_LOC_SPEC	Bad region location specified (Failure)
PRTC_ITEM_STAMP_TOO_NEW	An affected item has changed since the last display (Failure)
PRTC_ADD_LOC_TOO_BIG	Cannot add item to unoccupied location- inserting (Warning)
PRTC_ADD_MULTIPLE_SERIAL	Cannot have two serialized items at one location (Warning)
PRTC_REGION_NOT_LOCKED	Lockable region %s not locked; cannot perform operation (Warning)
PRTI_ITEM_TYPE_NULL	Item type identifier NULL (Warning)
PRTI_ITEM_TYPE_NDEF	Item type identifier not valid (Warning)
PRTI_NON_SER_ID	Non-serialized item has item id or reference id (Warning)

PRTI_SER_NO_ID	Serialized item missing both item id and reference id (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_INVALID_REG_LOC	Invalid region location (Warning)
PRTI_ITEM_TYPE_NULL	Item type identifier NULL (Warning)
PRTI_INVALID_DEST	Destination region is invalid for item of this type (Warning)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)

Plus all error codes returnable by `prt_api_send_cmd`.

`prt_api_prod_start`

Name

`prt_api_prod_start`

Purpose

Initiate production start for an item.

Description

This routine provides a mechanism to initiate production start for an Item in a Tracking Region.

Syntax

```
int prt_api_prod_start ( ext_proc_flag, region_id, item_id,
                        reference_id, item_type_id,
                        parent_item_id, item_status,
                        ext_hold_active, group_id,
                        int_hold_active, int_hold_reason,
                        num_atts_valid, item_att_list,
                        comment, user_or_svc_id, retstat)
COR_BOOLEAN ext_proc_flag;
char        region_id[PRT_REGION_ID_LEN + 1]
char        item_id[PRT_ITEM_ID_LEN + 1];
char        reference_id[PRT_ITEM_ID_LEN + 1];
char        item_type_id[PRT_ITEM_TYPE_ID_LEN + 1];
char        parent_item_id[PRT_ITEM_ID_LEN + 1];
COR_U4      item_status;
COR_BOOLEAN ext_hold_active;
char        group_id[PRT_REGION_ID_LEN + 1];
COR_BOOLEAN int_hold_active;
char        *int_hold_reason;
```

```

COR_I1          num_atts_valid;
PRT_ITEM_ATT   *item_att_list;
char           comment[SC_DESCRIPTION_LEN + 1];
char           user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS     *retstat;

```

Data Structures

See Static Information Requirements

Arguments

Argument	Description
Input	
Serialized Items Only	
ext_proc_flag	External process flag; if TRUE, get all information from external process. (required)
item_id	Unique identifier of a Serialized Item (optional if reference_id specified)
reference_id	Secondary identifier of a Serialized Item (optional if item_id specified)
ext_hold_active	If TRUE, item held due to external hold (required).
group_id	Group identifier providing reason for hold (required if ext_hold_active = TRUE).
num_atts_valid	Number of item attributes valid. This specifies the number of array elements in the item attribute list that contain valid data. If item_att_list is a NULL pointer, the function sends a zero to the Data Collector for this argument, regardless of what was actually passed.
*item_att_list	Item attribute list. This argument is the pointer to an array of PRT_NUM_ATTRIBUTES elements. If the pointer is NULL, this clearly indicates that no item attributes are being provided along with the item being added. If the pointer is non-NULL, however, it must point to an array of exactly PRT_NUM_ATTRIBUTES elements. The num_atts_valid argument indicates how many of these elements contain valid data.
Non-Serialized Items Only	
parent_item_id	Associated serialized item identifier (optional).
Both	
region_id	Region where the item is to be added to the queue (required)
item_type_id	Item type identifier (required)
int_hold_active	If TRUE, item held due to internal hold. (required)
int_hold_reason	Comment specifying reason for internal hold (required if int_hold_active = TRUE).
comment	Comment to be recorded in PRT history log file (optional).
user_or_svc_id	User or service identifier. This is used for logging purposes and is optional. If this pointer is NULL, the host process id is used (optional).

Argument	Description
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS or COR_FAILURE.

If the function returns COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

PRTC_NON_SER_ID	Non-serialized item has item or ref. id (Failure)
PRTC_SER_NO_ID	Serialized item missing both item & ref. id (Failure)
PRTC_UNKNOWN_REGION	Unknown REGION specified (Failure)
PRTC_BAD_REG_LOC_SPEC	Bad region location specified (Failure)
PRTC_INS_IN_OCC_NON_NORMAL	Attempt to Insert in occupied COMBINE/DISPERSE region (Failure)
PRTC_ITEM_STAMP_TOO_NEW	An affected item has changed since the last display (Failure)
PRTC_ADD_LOC_TOO_BIG	Cannot add item to unoccupied location- inserting (Warning)
PRTC_ADD_MULTIPLE_SERIAL	Cannot have two serialized items at one location (Warning)
PRTC_NO_DETAINMENT_CFG	No detainment region configured for (Warning)
PRTI_ITEM_TYPE_NULL	Item type identifier NULL (Warning)
PRTI_ITEM_TYPE_NDEF	Item type identifier not valid (Warning)
PRTI_NS_EXT_PROC_INVALID	ext_proc_flag set for non- serialized item (Warning)
PRTI_NON_SER_ID	Non-serialized item has item id or reference id (Warning)
PRTI_SER_NO_ID	Serialized item missing both item id and reference id (Warning)
PRTI_SER_PARENT	Parent ID specified for serialized item (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_EXT_HOLD_NON_SER	External hold specified for non-serialized item (Warning)
PRTI_INT_HOLD_NO_REASON	Internal hold specified without reason (Warning)
PRTI_NON_SER_ATTS	Non-serialized item has attribute(s) (Warning)
PRTI_NUM_ATTS_INVALID	Invalid number of item attributes specified (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)

PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)

Plus all error codes returnable by:

- prt_api_check_route
- prt_api_send_cmd

prt_api_region_setpoints

Name

prt_api_region_setpoints

Purpose

Enable/disable tracking region setpoints.

Description

This routine provides a mechanism for enabling and disabling the Item transition setpoints configured in the PRT Tracking Route records. These points are set by PRT when an Item transitions into the configured destination region.

Syntax

```
int prt_api_region_setpoints ( region_id, enable_setpt_flag,
                             comment, user_or_svc_id, retstat )
char    region_id[PRT_REGION_ID_LEN + 1];
COR_BOOLEAN enable_setpt_flag;
char    comment[SC_DESCRIPTION_LEN + 1];
char    user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS *retstat;
```

Data Structures

See Static Information Requirements

Arguments

Argument	Description
Input	
region_id	Tracking Region identifier (required)

Argument	Description
enable_setpt_flag	If TRUE, enable the region item transition setpoint. If FALSE, disable it. (required)
Comment	Comment to be recorded in PRT history log file (optional).
user_or_svc_id	User or service identifier. This is used for logging purposes and is optional. If this pointer is NULL, the host process id is used. (optional)
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS or COR_FAILURE.

If the function returns COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

PRTC_BAD_REG_SPEC	Bad region specified (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)

Plus all error codes returnable by `prt_api_send_cmd`.

`prt_api_register_intproc`

Name

`prt_api_register_intproc`

Purpose

Register as an interested process.

Description

This function provides a mechanism for a process to request to register as an interested process to a PRT process. Either this initialization routine must be called by all PRT interested processes prior to calling any other PRT_API interested process utilities or `prt_api_dyn_register_intproc`. The calling routine receives back a status indicating the success or failure of the request. This routine is used

when you have configured your process as a service and is included in the `prt_service.dat` file. If this is not the case then you must use `prt_api_dyn_register_intproc`.

Syntax

```
int prt_api_register_intproc (intproc_event_flag, retstat)
COR_I4      intproc_event_flag;
COR_STATUS  *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	
<code>intproc_event_flag</code>	Interested process event flag (required)
Output	
<code>*retstat</code>	Pointer to <code>COR_STATUS</code> structure

Return Value

Either `COR_SUCCESS` or `COR_FAILURE`.

If the function returns `COR_FAILURE` additional error information can be found in the `COR_STATUS` structure.

Error Codes

<code>PRTI_PROC_NOT_CONFIGURED</code>	Process is not configured as network process (Failure)
<code>PRTI_FILE_OPEN_ERR</code>	Error opening one of the configuration files (Failure)
<code>PRTI_SVC_ID_NDEF</code>	PRT service identifier not valid (Failure)
<code>PRTI_FILE_READ_ERR</code>	Error reading configuration file (Failure)
<code>PRTI_EMPTY_FILE</code>	Configuration file has no records (Failure)
<code>PRTI_AUX_NAM_ERR</code>	Error in <code>ipc_aux_nam</code> (Failure)
<code>PRTI_ADD_PORT_ERR</code>	Error in <code>ipc_add_port</code> (Failure)
<code>PRTI_READ_PORT_ERR</code>	Error in <code>ipc_read_port</code> (Failure)
<code>PRTI_MF_INIT_ERR</code>	Error initializing segment (Failure)
<code>PRTI_CRESEG_ERR</code>	Error creating segment (Failure)

PRTI_BAD_MSG	Error in segment (Failure)
PRTI_BAD_SVC_NAME	Bad service name (no underscore) (Failure)
PRTI_XLATE_ERR	Error in ipc_xlate getting physical addr (Failure)
PRTI_WRITE_PORT_ERR	Error in ipc_write_port (Warn/Fail)

prt_api_reorder_region

Name

`prt_api_reorder_region`

Purpose

Reorder locations within tracking region.

Description

This routine provides a mechanism to completely reorder the locations within a tracking region. This has the effect of moving all the items at each location in the region to another (possibly the same) location in the region.

Syntax

```
int prt_api_reorder_region ( region_id, reorder_list,
                           num_locations, comment,
                           user_or_svc_id, retstat )
char   region_id[PRT_REGION_ID_LEN + 1];
COR_I2 reorder_list[num_locations];
COR_I2 num_locations;
char   comment[SC_DESCRIPTION_LEN + 1];
char   user_or_svc_id[SERVICE_ID_LEN + 1];
COR_STATUS *retstat;
```

Data Structures

See Static Information Requirements

Arguments

Argument	Description
Input	
region_id	Tracking Region identifier (required)

Argument	Description
reorder_list	Pointer to an array consisting of region locations. These location numbers (1 to num_locations), specify the new region location for items residing in their current location, e.g. the locations the items are to be moved to. The reorder_list array index implicitly identifies the current location number, i.e. the first element in the reorder_list array (reorder_list[0]) refers to the items that currently reside in region location 1. The value of reorder_list[0] specifies the region location where the items currently in region location 1 are to be moved to. This array must include elements for each location currently occupied in the region.
num_locations	The number of locations in the region which currently are occupied. This number must match what the Data Collector thinks are the number of locations currently occupied or else the Data Collector will return an error (err_code = PRTC_REORDER_MISMATCH) and the operation will not be performed. Failure to allocate the reorder_list array with num_locations elements is bad.
comment	Comment to be recorded in PRT history log file (optional).
user_or_svc_id	User or service identifier. This is used for logging purposes and is optional. If this pointer is NULL, the host process id is used. (optional)
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS or COR_FAILURE.

If the function returns COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

PRTC_BAD_REG_SPEC	Bad region specified (Warning)
PRTC_REORDER_MISMATCH	Num. of locs in request different from num. of locs in region (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_REORD_NUM_TOO_BIG	Element # bigger than list size (Warning)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)

Plus all error codes returnable by `prt_api_send_cmd`.

`prt_api_req_class_itemlist`

Name

`prt_api_req_class_itemlist`

Purpose

Request list of item tracking data.

Description

This routine provides a mechanism to request the Item Tracking Data for one or more Items of a particular Item Class from the PRT Data Server (PRT_DS) residing within all configured PRT services, a specific PRT service, a specific Tracking Group, a specific Tracking Region, or at a specific location within a specific Tracking Region. If the request is a success the calling routine will receive back a count of the number of items returned to the item list. If the request fails, the calling routine will receive back status information on the reason for the failure.

Syntax

```
int prt_api_req_class_itemlist (prt_svc_id, group_id,
                               region_id, region_loc,
                               item_class_id, item_count,
                               max_location, retstat)
char      prt_svc_id[SERVICE_ID_LEN + 1];
char      group_id[PRT_REGION_ID_LEN + 1];
char      region_id[PRT_REGION_ID_LEN + 1];
COR_I2    region_loc;
char      item_class_id[PRT_ITEM_CLASS_ID_LEN+1];
COR_I4    *item_count;
COR_I4    *max_location;
COR_STATUS *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	
prt_svc_id	PRT service identifier (optional, but required if no other arguments specified)
group_id	Identifier of the group. (optional, but required if no other arguments specified)
region_id	Identifier of the region. (optional, but required if no other arguments specified, required if region location specified) Note: At least one of the above three arguments must be specified. If not, all services will be queried. It is possible that a list of 'items' will be collected, but not all services will have responded (because one or more of them were down). Thus the user has a list of valid responses, but may not have ALL the 'items' that should have been returned. In this case, the function will return COR_WARNING with an err_code of PRTI_NOT_ALL_SVCS_RESPONDED. The function returns, through an argument passed, the number of 'items' actually received. This argument will show how many valid 'items' were received.

Argument	Description
region_loc	Specific location in the region (required). If all items within a region are being requested, the region_loc must be specified as ALL_LOCATIONS. Otherwise a specific location requests items at only that location.
Item_class_id	Filter to use to select a particular class. Enter a blank to select all items (that is for no filtering).
Output	
*item_count	Count of Items returned to schema
*max_location	Location number of the highest populated location in the region, regardless of the Item Class in the region.
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

Error Codes

PRTS_NO_GLOBAL_XREF	No XREF global section available (Warning)
PRTS_REGION_NOT_DEFINED	Region not configured (Warning)
PRTS_NO_GLOBAL_REGION	No global section available for Region (Warning)
PRTS_NO_ITEMS_AT_LOCATION	There are no items at location specified (Warning)
PRTI_SVC_ID_NULL	PRT service identifier NULL (Warning)
PRTI_SVC_ID_NDEF	PRT service identifier not valid (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_INVALID_REG_LOC	Invalid region location (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_REG_NOT_IN_GRP	Region not in group specified (Warning)
PRTI_REG_NOT_IN_SVC	Region not managed by service specified (Warning)
PRTI_GRP_NOT_IN_SVC	Group ID inconsistent with service specified (Warning)
PRTI_UNEXPECTED_SEG_TYPE	Unexpected segment type from PRT_DS (Warning)
PRTI_PARTNER_DEAD	Partner dead (Warning)
PRTI_EXTRA_SEGS	Ignored extra segments after STATUS_SEG or END_SEG (Warning)

PRTI_NOT_ALL_SVCS_RESPONDED	Not all services responded (Warning)
PRTI_WRITE_PORT_ERR	Error in ipc_write_port (Warn/Fail)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)
PRTI_BAD_MSG	Error in segment (Failure)
PRTI_READ_PORT_ERR	Error in ipc_read_port (Failure)

prt_api_req_class_itemlist_Ex

Name

prt_api_req_class_itemlist_Ex

Purpose

Request list of item tracking data.

Description:

This routine provides a mechanism to request the Item Tracking Data for one or more Items of a particular Item Class from the PRT Data Server (PRT_DS) residing within all configured PRT services, a specific PRT service, a specific Tracking Group, a specific Tracking Region, or at a specific location within a specific Tracking Region. If the request is a success the calling routine will receive back a count of the number of items returned to the item list. If the request fails, the calling routine will receive back status information on the reason for the failure.

Syntax

```
int prt_api_req_class_itemlist (prt_svc_id, group_id,
                               region_id, region_loc,
                               item_class_id, item_count,
                               max_location, search_flag, retstat)
char    prt_svc_id[SERVICE_ID_LEN + 1];
char    group_id[PRT_REGION_ID_LEN + 1];
char    region_id[PRT_REGION_ID_LEN + 1];
COR_I2  region_loc;
char    item_class_id[PRT_ITEM_CLASS_ID_LEN+1];
COR_I4  *item_count;
COR_I4  *max_location;
COR_BOOL search_flag;
COR_STATUS *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	
prt_svc_id	PRT service identifier (optional, but required if no other arguments specified)
group_id	Identifier of the group. (optional, but required if no other arguments specified)
region_id	Identifier of the region. (optional, but required if no other arguments specified, required if region location specified) Note: At least one of the above three arguments must be specified. If not, all services will be queried. It is possible that a list of 'items' will be collected, but not all services will have responded (because one or more of them were down). Thus the user has a list of valid responses, but may not have ALL the 'items' that should have been returned. In this case, the function will return COR_WARNING with an err_code of PRTI_NOT_ALL_SVCS_RESPONDED. The function returns, through an argument passed, the number of 'items' actually received. This argument will show how many valid 'items' were received.
region_loc	Specific location in the region (required). If all items within a region are being requested, the <code>region_loc</code> must be specified as ALL_LOCATIONS. Otherwise a specific location requests items at only that location.
Item_class_id	Filter to use to select a particular class. Enter a blank to select all items (that is for no filtering).
Search_flag	If search_flag is 0 then item are searched in the default project if any default is set already and if not found then other configured projects are searched. For setting a project as default project we can use <code>prt_api_cur_project</code> .
Output	
*item_count	Count of Items returned to schema
*max_location	Location number of the highest populated location in the region, regardless of the Item Class in the region.
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

Error Codes

PRTS_NO_GLOBAL_XREF	No XREF global section available (Warning)
PRTS_REGION_NOT_DEFINED	Region not configured (Warning)

PRTS_NO_GLOBAL_REGION	No global section available for Region (Warning)
PRTS_NO_ITEMS_AT_LOCATION	There are no items at location specified (Warning)
PRTI_SVC_ID_NULL	PRT service identifier NULL (Warning)
PRTI_SVC_ID_NDEF	PRT service identifier not valid (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_INVALID_REG_LOC	Invalid region location (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_REG_NOT_IN_GRP	Region not in group specified (Warning)
PRTI_REG_NOT_IN_SVC	Region not managed by service specified (Warning)
PRTI_GRP_NOT_IN_SVC	Group ID inconsistent with service specified (Warning)
PRTI_UNEXPECTED_SEG_TYPE	Unexpected segment type from PRT_DS (Warning)
PRTI_PARTNER_DEAD	Partner dead (Warning)
PRTI_EXTRA_SEGS	Ignored extra segments after STATUS_SEG or END_SEG (Warning)
PRTI_NOT_ALL_SVCS_RESPONDED	Not all services responded (Warning)
PRTI_WRITE_PORT_ERR	Error in ipc_write_port (Warn/Fail)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)
PRTI_BAD_MSG	Error in segment (Failure)
PRTI_READ_PORT_ERR	Error in ipc_read_port (Failure)

prt_api_req_groupitemlist

Name

`prt_api_req_groupitemlist`

Purpose

Request list of serialized item ID's in a tracking group.

Description

This routine provides a mechanism to request all Serialized Item IDs in a Tracking Group. If the request is a success the calling routine will receive back a count of the number of items in the list

of Serialized Items in the Group. If the request fails, the calling routine will receive back status information on the reason for the failure.

Syntax

```
prt_api_req_groupitemlist (group_id, item_count, retstat)
char          group_id[PRT_REGION_ID_LEN + 1];
COR_I4       *item_count;
COR_STATUS   *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	
group_id	Unique identifier of tracking group (required)
Output	
*item_count	Count of Items returned to schema
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

Error Codes

PRTS_NO_GLOBAL_XREF	No XREF global section available (Warning)
PRTS_REGION_NOT_DEFINED	Region not configured (Warning)
PRTS_NO_GLOBAL_REGION	No global section available for Region (Warning)
PRTS_NO_ITEMS_AT_LOCATION	There are no items at location specified (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_UNEXPECTED_SEG_TYPE	Unexpected segment type from PRT_DS (Warning)
PRTI_PARTNER_DEAD	Partner dead (Warning)
PRTI_EXTRA_SEGS	Ignored extra segments after STATUS_SEG or END_SEG (Warning)

PRTI_WRITE_PORT_ERR	Error in ipc_write_port (Warn/Fail)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)
PRTI_BAD_MSG	Error in segment (Failure)
PRTI_READ_PORT_ERR	Error in ipc_read_port (Failure)

prt_api_req_grouplist

Name

prt_api_req_grouplist

Purpose

Request list of tracking group data.

Description

This routine provides a mechanism to request Tracking Group Data for one or more Tracking Groups residing within all configured PRT services, a specific PRT service or a specific Tracking Group. If the request is a success the calling routine will receive back a count of the number of Tracking Groups returned to the group list. If the request fails, the calling routine will receive back status information on the reason for the failure.

Syntax

```
int prt_api_req_grouplist (prt_svc_id, group_id, group_count, retstat)
char      prt_svc_id[SERVICE_ID_LEN + 1];
char      group_id[PRT_REGION_ID_LEN + 1];
COR_I4    *group_count;
COR_STATUS *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	
prt_svc_id	Service identifier (optional, if group_id is specified)

Argument	Description
group_id	Tracking group identifier (optional, if prt_svc_id is specified) Note: At least one of the above arguments must be specified. If not, all services will be queried. It is possible that a list of 'items' will be collected, but not all services will have responded (because one or more of them were down). Thus the user has a list of valid responses, but may not have all the 'items' that should have been returned. In this case, the function will return COR_WARNING with an err_code of PRTI_NOT_ALL_SVCS_RESPONDED. The function returns, through an argument passed, the number of 'items' actually received. This argument will show how many valid 'items' were received.
Output	
*group_count	Count of Groups returned to schema
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

PRTS_GROUP_NOT_DEFINED	Group not configured (Warning)
PRTS_NO_INFO	No information available due to unknown reason (Warning)
PRTS_NO_GLOBAL_GROUP	No GROUPS global section available for (Warning)
PRTI_SVC_ID_NULL	PRT service identifier NULL (Warning)
PRTI_SVC_ID_NDEF	PRT service identifier not valid (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_GRP_NOT_IN_SVC	Group ID inconsistent with service specified (Warning)
PRTI_UNEXPECTED_SEG_TYPE	Unexpected segment type from PRT_DS (Warning)
PRTI_PARTNER_DEAD	Partner dead (Warning)
PRTI_EXTRA_SEGS	Ignored extra segments after STATUS_SEG or END_SEG (Warning)
PRTI_NOT_ALL_SVCS_RESPONDED	Not all services responded (Warning)
PRTI_WRITE_PORT_ERR	Error in ipc_write_port (Warn/Fail)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)
PRTI_BAD_MSG	Error in segment (Failure)
PRTI_READ_PORT_ERR	Error in ipc_read_port (Failure)

prt_api_req_itemlist

Name

prt_api_req_itemlist

Purpose

Request list of item tracking data.

Description

This routine provides a mechanism to request the Item Tracking Data for one or more Items from the PRT Data Server (PRT_DS) residing within all configured PRT services, a specific PRT service, a specific Tracking Group, a specific Tracking Region, or at a specific location within a specific Tracking Region. If the request is a success the calling routine will receive back a count of the number of items returned to the item list. If the request fails, the calling routine will receive back status information on the reason for the failure.

Syntax

```

int prt_api_req_itemlist (prt_svc_id, group_id, region_id,
                        region_loc, item_count, retstat)
char      prt_svc_id[SERVICE_ID_LEN + 1];
char      group_id[PRT_REGION_ID_LEN + 1];
char      region_id[PRT_REGION_ID_LEN + 1];
COR_I2    region_loc;
COR_I4    *item_count;
COR_STATUS *retstat;

```

Data Structures

None

Arguments

Argument	Description
Input	
prt_svc_id	PRT service identifier (optional, but required if no other arguments specified)
group_id	Identifier of the group. (optional, but required if no other arguments specified)
region_id	Identifier of the region. (optional, but required if no other arguments specified, required if region location specified)

Argument	Description
	Note: At least one of the above three arguments must be specified. If not, all services will be queried. It is possible that a list of 'items' will be collected, but not all services will have responded (because one or more of them were down). Thus the user has a list of valid responses, but may not have all the 'items' that should have been returned. In this case, the function will return COR_WARNING with an err_code of PRTI_NOT_ALL_SVCS_RESPONDED. The function returns, through an argument passed, the number of 'items' actually received. This argument will show how many valid 'items' were received.
region_loc	Specific location in the region (required). If all items within a region are being requested, the region_loc must be specified as ALL_LOCATIONS. Otherwise a specific location requests items at only that location.
Output	
*item_count	Count of Items returned to schema
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

Error Codes

PRTS_NO_GLOBAL_XREF	No XREF global section available (Warning)
PRTS_REGION_NOT_DEFINED	Region not configured (Warning)
PRTS_NO_GLOBAL_REGION	No global section available for Region (Warning)
PRTS_NO_ITEMS_AT_LOCATION	There are no items at location specified (Warning)
PRTI_SVC_ID_NULL	PRT service identifier NULL (Warning)
PRTI_SVC_ID_NDEF	PRT service identifier not valid (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_INVALID_REG_LOC	Invalid region location (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_REG_NOT_IN_GRP	Region not in group specified (Warning)
PRTI_REG_NOT_IN_SVC	Region not managed by service specified (Warning)
PRTI_GRP_NOT_IN_SVC	Group ID inconsistent with service specified (Warning)
PRTI_UNEXPECTED_SEG_TYPE	Unexpected segment type from PRT_DS (Warning)

PRTI_PARTNER_DEAD	Partner dead (Warning)
PRTI_EXTRA_SEGS	Ignored extra segments after STATUS_SEG or END_SEG (Warning)
PRTI_NOT_ALL_SVCS_RESPONDED	Not all services responded (Warning)
PRTI_WRITE_PORT_ERR	Error in ipc_write_port (Warn/Fail)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)
PRTI_BAD_MSG	Error in segment (Failure)
PRTI_READ_PORT_ERR	Error in ipc_read_port (Failure)

prt_api_req_regionlist

Name

prt_api_req_regionlist

Purpose

Request list of tracking region data.

Description

This routine provides a mechanism to request Tracking Region Data for one or more Tracking Regions in a Tracking Region Group residing within all configured PRT services, a specific PRT service, a specific Tracking Group or a specific Tracking Region. If the request is a success the calling routine will receive back a count of the number of Tracking Regions returned to the region list. If the request fails, the calling routine will receive back status information on the reason for the failure.

Syntax

```
prt_api_req_regionlist (prt_svc_id, group_id, region_id,
                      item_count, retstat)
char    prt_svc_id[SERVICE_ID_LEN + 1];
char    group_id[PRT_REGION_ID_LEN + 1];
char    region_id[PRT_REGION_ID_LEN + 1];
COR_I4  *item_count;
COR_STATUS *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	
prt_svc_id	PRT service identifier (optional, but required if no other arguments specified)
group_id	Tracking group identifier (optional, but required if no other arguments specified)
region_id	Tracking region identifier (optional, but required if no other arguments specified) Note: At least one of the above three arguments must be specified. If not, all services will be queried. It is possible that a list of 'items' will be collected, but not all services will have responded (because one or more of them were down). Thus the user has a list of valid responses, but may not have all the 'items' that should have been returned. In this case, the function will return COR_WARNING with an err_code of PRTI_NOT_ALL_SVCS_RESPONDED. The function returns, through an argument passed, the number of 'items' actually received. This argument will show how many valid 'items' were received.
Output	
*item_count	Count of regions returned to schema
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

Error Codes

PRTS_REGION_NOT_DEFINED	Region not configured (Warning)
PRTS_GROUP_NOT_DEFINED	Group not configured (Warning)
PRTS_NO_REGIONS_MAPPED	No regions attempted were mapped (Warning)
PRTS_NO_INFO	No information available due to unknown reason (Warning)
PRTS_NO_GLOBAL_REGION	No global section available for Region (Warning)
PRTI_SVC_ID_NULL	PRT service identifier NULL (Warning)
PRTI_SVC_ID_NDEF	PRT service identifier not valid (Warning)
PRTI_REG_ID_NULL	Region identifier NULL (Warning)
PRTI_REG_ID_NDEF	Region identifier not valid (Warning)
PRTI_GRP_ID_NULL	Group identifier NULL (Warning)
PRTI_GRP_ID_NDEF	Group identifier not valid (Warning)
PRTI_REG_NOT_IN_GRP	Region not in group specified (Warning)

PRTI_REG_NOT_IN_SVC	Region not managed by service specified (Warning)
PRTI_GRP_NOT_IN_SVC	Group ID inconsistent with service specified (Warning)
PRTI_UNEXPECTED_SEG_TYPE	Unexpected segment type from PRT_DS (Warning)
PRTI_PARTNER_DEAD	Partner dead (Warning)
PRTI_EXTRA_SEGS	Ignored extra segments after STATUS_SEG or END_SEG (Warning)
PRTI_NOT_ALL_SVCS_RESPONDED	Not all services responded (Warning)
PRTI_WRITE_PORT_ERR	Error in ipc_write_port (Warn/Fail)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)
PRTI_CRESEG_ERR	Error creating segment (Failure)
PRTI_BAD_MSG	Error in segment (Failure)
PRTI_READ_PORT_ERR	Error in ipc_read_port (Failure)

prt_api_send_resp

Name

`prt_api_send_resp`

Purpose

Send response to production tracking.

Description

This API provides a mechanism for external processes to send asynchronous responses to PRT_API requests. This routine packages the applications response and sends it to the appropriate PRT resident process.

Syntax

```
int prt_api_send_resp (prt_svc_id, retstat)
char          prt_svc_id[SERVICE_ID_LEN +1];
COR_STATUS   *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	
prt_svc_id	PRT service identifier
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS or COR_FAILURE.

If the function returns COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

PRTI_SVC_ID_NULL	PRT service identifier NULL (Warning)
PRTI_SVC_ID_NDEF	PRT service identifier not valid (Warning)
PRTI_WRITE_PORT_ERR	Error in ipc_write_port (Warn/Fail)
PRTI_BAD_MSG	Error in segment (Failure)
PRTI_MF_INIT_ERR	Error initializing segment (Failure)

prt_api_static_region_data

Name

prt_api_static_region_data

Purpose

Copy static region information into region structure.

Description

This function copies the static part of the region information into the region structure supplied by the application. Note that only region-configured capacity, status and quantity points are returned. If an invalid region is supplied, the **region_id** field is not populated.

Syntax

```
int prt_api_static_region_data (region_struct, region_id)
PRT_REGION_STRUCT region_list[list_size];
```



```
char          *region_id;
```

Data Structures

```
typedef struct {
    char    region_id[PRT_REGION_ID_LEN + 1] - unique identifier
                                                of tracking region.
    COR_U4  region_status - region status.
    COR_I4  total_items - total number of items in region
    COR_I2  region_cap - number of item carriers in region
    char    stat_ptid[POINT_ID_LEN + 1] - status point identifier
    char    item_qty_ptid[POINT_ID_LEN + 1] - item quantity point id
    COR_I2  total_item_counts - tells how many PRT_ITEM_COUNTS
                                elements are in the array pointed
                                to by type_totals.
    PRT_ITEM_COUNTS *type_totals - pointer to array of
                                PRT_ITEM_COUNTS structures
                                (array must be deallocated by the user)
} PRT_REGION_STRUCT, *PRT_REGION_STRUCT_PTR;
```

Arguments

Argument	Description
Input	
region_struct	Pointer to a buffer to receive tracking region data (required)
region_id	Character string identifying the region of interest.
Output	
region_struct	Populated tracking region data

Return Value

If the **region_id** field in the **region_struct** buffer contains the supplied **region_id**, then the configured region quantity in the **region_cap** field, the status point in the **stat_ptid** field, and the item quantity point in the **item_qty_ptid** field are all populated with valid data; otherwise, the region could not be found, and the fields contain invalid data.

prt_api_term

Name

prt_api_term

Purpose

Terminate PRT_API interface.

Description

This routine is responsible for de-allocating the communication buffers allocated by the `prt_api_init` routine. If the application registered as an interested process to a PRT process it must call the `prt_api_term_intproc` routine.

Syntax

```
int prt_api_term (retstat)
COR_STATUS *retstat;
```

Data Structures

None

Argument	Description
Input	
None	
Output	
*retstat	Pointer to COR_STATUS structure.

Return Value

`COR_SUCCESS`

Error Codes

None

`prt_api_term_intproc`

Name

`prt_api_term_intproc`

Purpose

Terminate PRT_API Interested Process interface.

Description

This routine is responsible for de-allocating all memory allocated by the PRT_API interested process routines and for terminating any services to other application programmer interfaces.

Syntax

```
int prt_api_term_intproc (retstat)
COR_STATUS *retstat;
```

Data Structures

See Static Information Requirements

Arguments

Argument	Description
Input	
None	
Output	
*retstat	Pointer to COR_STATUS structure

Return Value

None

prt_api_unload_groupitemlist

Name

prt_api_unload_groupitemlist

Purpose

Unload list of serialized item ID's in a tracking group.

Description

This routine provides a mechanism to unload the list of Serialized Items obtained through the prt_api_req_groupitemlist routine. If the request is a success the calling routine will receive back an array populated with the Serialized Item information. If the request fails, the calling routine will receive back status information on the reason for the failure.

Syntax

```
int prt_api_unload_groupitemlist (list_size, item_hold_list,
                                retstat)
COR_I4                          list_size;
PRT_ITEM_HOLD_STRUCT            item_hold_list[list_size];
COR_STATUS                      *retstat;
```

Data Structures

```
typedef struct {
char item_id[PRT_ITEM_ID_LEN + 1] - unique identifier of
                                the Item.
char group_id[PRT_REGION_ID_LEN + 1] - identifier of group
                                responsible for
                                placing a hold on this item
COR_BOOLEAN ext_hold_active - if TRUE, HOLD is activated
                                for Item.
} PRT_ITEM_HOLD_STRUCT, *PRT_ITEM_HOLD_STRUCT_PTR;
```

Arguments

Argument	Description
Input	
list_size	Size allocated for item_hold_list array (required)
item_hold_list	Array of hold items (required)
Output	
item_hold_list	Populated array of hold items
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_FAILURE or COR_WARNING.

If the function returns COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

PRTI_TOO_MANY_REQ	Total number in list less than number requested (Warning)
-------------------	---

prt_api_unload_grouplist

Name

`prt_api_unload_grouplist`

Purpose

Unload list of tracking groups.

Description

This routine provides a mechanism to unload the list of Tracking Groups obtained through the `prt_api_req_grouplist` routine. If the request is a success the calling routine will receive back an array populated with the Tracking Group Data. If the request fails, the calling routine will receive back status information on the reason for the failure.

Syntax

```
int prt_api_unload_grouplist (list_size, group_list, retstat)
COR_I4          list_size;
PRT_GROUP_STRUCT group_list[list_size];
COR_STATUS      *retstat;
```

Data Structures

```
typedef struct {
    char group_id[PRT_REGION_ID_LEN + 1] - unique identifier of
                                         tracking group.
    COR_BOOLEAN ext_hold_active - TRUE indicates HOLD is active
                                for group.
    char ext_hold_reason[SC_DESCRIPTION_LEN + 1] - reason for
                                                Item HOLDS.
} PRT_GROUP_STRUCT, * PRT_GROUP_STRUCT_PTR;
```

Arguments

Argument	Description
Input	
list_size	Number of array elements allocated for region_list array (required)
region_list	Pointer to a buffer to receive tracking region data (required)
Output	
group_list[list_size]	Populated list of groups
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

Error Codes

PRTI_TOO_MANY_REQ	Total number in list less than number requested (Warning).
-------------------	--

prt_api_unload_itemlist

Name

prt_api_unload_itemlist

Purpose

Unload list of items.

Description

This routine provides a mechanism to unload the list of Items obtained through the prt_api_req_itemlist routine. If the request is a success the calling routine will receive back an array populated with the Item Tracking Data information. If the request fails, the calling routine will receive back status information on the reason for the failure.

Syntax

```
int prt_api_unload_itemlist (list_size, item_list, retstat)
COR_I4          list_size;
PRT_ITEM_STRUCT item_list[list_size];
COR_STATUS      *retstat;
```

Data Structures

```
typedef struct {
  cor_time_t - reg_entry_time
  cor_time_t - last_mod_time
  char region_id[PRT_REGION_ID_LEN + 1] - identifier of the region
  COR_I2 region_loc - specific location in the region
  char item_id[PRT_ITEM_ID_LEN + 1] - unique identifier of a
  Serialized Item.
  char reference_id[PRT_ITEM_ID_LEN + 1] - secondary identifier
  of a Serialized Item.
  char item_type_id[PRT_ITEM_TYPE_ID_LEN + 1] - item type identifier.
```

```

char parent_item_id[PRT_ITEM_ID_LEN + 1] - pertains only to
                                         Non-serialized Items.
COR_U4 item_status - item status.
COR_BOOLEAN int_hold_active - if TRUE, item held due to
                              internal hold.
char int_hold_reason[SC_DESCRIPTION_LEN + 1] - Comment
                                         specifying reason for internal hold.
COR_BOOLEAN ext_hold_active - if TRUE, item held due to
                              external hold.
char group_id[PRT_REGION_ID_LEN + 1] - identifier of the group
COR_I1 num_atts_valid - Number of valid attributes in the list
PRT_ITEM_ATT item_att_list[PRT_NUM_ATTRIBUTES] - Item Attributes
                                         pertain to Serialized Items.
} PRT_ITEM_STRUCT, *PRT_ITEM_STRUCT_PTR

```

Arguments

Argument	Description
Input	
list_size	Number of array elements allocated for item_list array (required)
item_list	Pointer to a buffer to receive returned items (required)
Output	
item_list	Populated list of Items
*retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_FAILURE or COR_WARNING.

If the function returns COR_FAILURE additional error information can be found in the COR_STATUS structure.

Error Codes

PRTI_TOO_MANY_REQ	Total number in list less than number requested (Warning).
-------------------	--

prt_api_unload_regionlist

Name

prt_api_unload_regionlist

Purpose

Unload list of tracking region data.

Description

This routine provides a mechanism to unload the list of Tracking Regions obtained through the `prt_api_req_regionlist` routine. If the request is a success the calling routine will receive back an array populated with the Tracking Region Data. This array contains a pointer to another array (allocated by the application interface routine) which contains an element for each type of item in the region and the total number of items of that type in the region. The caller is responsible for de-allocating the `PRT_ITEM_COUNTS` array when it is no longer needed. If the request fails, the calling routine will receive back status information on the reason for the failure.

Syntax

```
int prt_api_unload_regionlist (list_size, region_list, retstat)
COR_I4          list_size;
PRT_REGION_STRUCT  region_list[list_size];
COR_STATUS      *retstat;
Data Structures
typedef struct {
    COR_I4 item_type_count - number of items of that type in region
    char   item_type_id[PRT_ITEM_TYPE_ID_LEN] - item type identifier
} PRT_ITEM_COUNTS;
typedef struct {
    char   region_id[PRT_REGION_ID_LEN + 1] - unique identifier
                                                of tracking region.
    COR_U4 region_status - region status.
    COR_I4 total_items - total number of items in region
    COR_I2 region_cap - number of item carriers in region
    char   stat_ptid[POINT_ID_LEN + 1] - status point identifier
    char   item_qty_ptid[POINT_ID_LEN + 1] - item quantity point id
    COR_I2 total_item_counts - tells how many PRT_ITEM_COUNTS
                                elements are in the array pointed
                                to by type_totals.
    PRT_ITEM_COUNTS *type_totals - pointer to array of PRT_ITEM_COUNTS
                                structures (array must be
                                deallocated by the user)
} PRT_REGION_STRUCT, *PRT_REGION_STRUCT_PTR;
```

Arguments

Argument	Description
Input	
list_size	Number of array elements allocated for region_list array (required)

Argument	Description
region_list	Pointer to a buffer to receive tracking region data (required)
Output	
region_list	Populated array of Tracking Region Data
*retstat	pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS, COR_WARNING or COR_FAILURE.

If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in the COR_STATUS structure.

Error Codes

PRTI_TOO_MANY_REQ	Total number in list less than number requested (Warning)
-------------------	---

The following code example shows how to de-allocate the user memory:

```
PRT_REGION_STRUCT region_struct, *region_struct_ptr;
COR_STATUS retstat;
if (prt_api_unload_regionlist (1, &region_struct, &retstat) == COR_SUCCESS)
{
    region_struct_ptr = &region_struct;
    /* Deallocate PRT_ITEM_COUNTS structure */
    if ( region_struct_ptr->total_item_counts > 0 )
    {
        cor_mfree ( region_struct_ptr->type_totals, 0 );
    }
}
```

prti_get_RCMconnection

Name

prti_get_RCMconnection

Purpose

Connect to remote connection manager.

Description

In order to connect to the data server on a remote node, this function must first make sure the remote node is active. Once we know it is active and we have initialized with RCM, RCM will handle the connections later.

Syntax

```
int prt_i_get_RCMconnection ( cimp_sys, retstat )
char *cimp_sys;
COR_STATUS *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	
cimp_sys	Pointer to a null terminated string containing the project or node name of the remote system to connect to.
Output	
retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS or COR_FAILURE

COR_SUCCESS is returned if the remote system is active. COR_FAILURE is returned if we cannot connect to the remote system.

prt_i_term_RCMconnection

Name

prt_i_term_RCMconnection

Purpose

Disconnect from remote connection manager.

Description

This function breaks the connection to the remote connection manager that was made by **prti_get_RCMconnection** .

Syntax

```
int prti_term_RCMconnection ( retstat )
COR_STATUS *retstat;
```

Data Structures

None

Arguments

Argument	Description
Input	
None	
Output	
retstat	Pointer to COR_STATUS structure

Return Value

Either COR_SUCCESS or COR_FAILURE

COR_SUCCESS is returned if the remote system is disconnected. COR_FAILURE is returned if we cannot disconnect from the remote system.

Process Resynchronization

The use of the **cor_dialect_check** function provides a convenient means for the user application to synchronize/resynchronize with the appropriate PRT processes. When an INIT_SEG is received by the application, and invoking the **cor_dialect_check** function returns the PRT subsystem string, the user has the following options:

- If the application is an Interested Process, the **prt_api_register_intproc** function should be called;
- If the application is an API client, the **prt_api_init** function should be called.
- If the application is both Interested Process and API client, both functions should be called.

These routines provide the user with a consistent means for synchronizing with PRT processes when the application program is initializing, and re-synchronizing with PRT processes after they have been stopped and restarted.

In order to use the **cor_dialect_check** function, the **cor_dialect.h** header file must be included in your source code file. You must manage your primary port, and call **cor_dialect_check** with the buffer returned from the **ipc_read** on the primary port. The application must also include **%BSM_ROOT%\api\lib\corutil.lib** in the link command file. The calling sequence is as follows:

Syntax

```
void cor_dialect_check ( input_buffer, port_index, result_str )
IPCDBG *input_buffer;
int port_index;
char result_str[COR_DIALECT_RETURN_STRING+1];
```

Data Structures

None.

Arguments

Argument	Description
Input	
input_buffer	Buffer returned from ipc_read on primary port
port_index	Output port index
Output	
result_str	Returned identifying string. See cor_dialect.h for valid strings returned.

Return Value

None.

PRT Macros

Following are macros for clearing, setting and testing the various bits of the **modify_bitmask** argument used by the **prt_api_modify_item** and **prt_api_get_intproc_updates** functions.

Macro	Description
SET	Sets the bit that corresponds to the actual data field in the item information structure.

Macro	Description
CLR	Clears the bit.
IS	Returns a non-zero value if the bit is set and zero if the bit is clear.

Example

```


SET_ITEM_ID_MOD_BIT(x)
CLR_ITEM_ID_MOD_BIT(x)          item_id
IS_ITEM_ID_MOD(x)
SET_REFERENCE_ID_MOD_BIT(x)
CLR_REFERENCE_ID_MOD_BIT(x)     reference_id
IS_REFERENCE_ID_MOD(x)
SET_ITEM_TYPE_BIT(x)
CLR_ITEM_TYPE_BIT(x)           item_type_id
IS_ITEM_TYPE(x)
SET_PARENT_ITEM_BIT(x)
CLR_PARENT_ITEM_BIT(x)         parent_item_id
IS_PARENT_ITEM(x)
SET_ITEM_STATUS_BIT(x)
CLR_ITEM_STATUS_BIT(x)         item_status
IS_ITEM_STATUS(x)
SET_INT_HOLD_BIT(x)
CLR_INT_HOLD_BIT(x)            int_hold_active
IS_INT_HOLD(x)                  If int_hold_active = TRUE,
                                int_hold_reason also contains valid data
SET_EXT_HOLD_BIT(x)
CLR_EXT_HOLD_BIT(x)            ext_hold_active
IS_EXT_HOLD(x)                  If ext_hold_active = TRUE,
                                group_id also contains valid data
SET_ATTS_VALID_BIT(x)
CLR_ATTS_VALID_BIT(x)          not needed when using the
IS_ATTS_VALID(x)               'prt_api_modify_item_att' function

```

Event Codes

Event Codes

When **prt_api_get_intproc_updates** returns updated information, one of the fields is the event code. This event code, which is a field in either the `PRT_ITEM_UPDATE_STRUCT` or `PRT_REGION_UPDATE_STRUCT`, indicates the exact nature of the update.

 **Note:** Note: **Modify_bitmask** is set to **0** for all codes except **IMOD**.

Item-Related Event Codes

- IADD

- IADX
- IAUN
- IAUX

- IDEL
- IDRF
- IINS (Destination)
- IINS (Via API)
- IMOD

- IMRA
- IMRM
- IMVN
- IMVX
- ISCR
- ISTP
- ISTT

IADD

When item is added via API (or user interface). If item is not production start and is added (not inserted).

region_1_id and **region_1_loc** contain the location of the event

IADX

When item is being deleted in source region due to being advanced via API (or user interface).

region_1_id and **region_1_loc** contain the location of item being deleted.

IAUN

When item is being added to destination region due to a point transition.

region_1_id and **region_1_loc** contain the destination of the added item.

IAUX

When serialized item is added to region and XREF table indicates that the item was already somewhere within the view of that service, the item already there is deleted.

region_1_id and **region_1_loc** contain the location of the item being deleted

When item is being deleted in source region due to a point transition.

region_1_id and **region_1_loc** contain the location of the item being deleted.

IDEL

When item deleted via API (or user interface) and item is being deleted (not scrapped and not production-stopped).

region_1_id and **region_1_loc** contain the location of the item being deleted.

IDRF

This item has been deleted because room was needed in the region.

region_1_id and **region_1_loc** contain the location of the item being deleted.

IINS (Destination Due)

When item is being added to destination region due to being advanced via API (or user interface).

region_1_id and **region_1_loc** contain the destination of added item.

IINS (Via API)

When item is added via API (or user interface). If item is not production start and is inserted (not added).

region_1_id and **region_1_loc** contain the location of the event.

IMOD

Automatic item association when adding an item:

PARENT_ITEM_BIT set in **modify_bitmask**

region_1_id contains the region id where the modify occurred

When modifying item:

modify_bitmask passed from API call

region_1_id contains the region id where the modify occurred

When item moves and **transition_type_indicator** is ASSOCIATE_REF_TO_ITEM or ASSOCIATE_TYPE_TO_ITEM:

modify_bitmask set to REFERENCE_ID_MOD_BIT or ITEM_TYPE_BIT respectively

region_1_id and **region_1_loc** contain the destination location.

IMRA

When a combine or disperse region has a serialized item in it and another serialized item is moved into the region, the item is automatically moved to a detainment region and two messages generated, one for the item being deleted and one for the item being added.

region_1_id and **region_1_loc** contain the region from which the item is being deleted

region_1_id and **region_1_loc** contain the destination of detainment region to which the item is being added

When an item becomes delayed and the region is configured to detain (rather than scrap) delayed items, two messages are generated for the item being detained, one for the item being deleted from the combine/disperse region and one for the item being added to the detainment region.

region_1_id and **region_1_loc** contain the location of item being deleted

region_1_id and **region_1_loc** contain the destination of detainment region to which the item is being added.

IMRM

When an item is moved via an API call (or user interface) to its detainment region, two messages are generated, one for the item being deleted from the source region and one for the item being added to the detainment region.

region_1_id and **region_1_loc** contain the location of the item being deleted

region_1_id and **region_1_loc** contain the destination of detainment region to which the item is being added.

IMVN

When item is being added to destination region due to being moved via API (or user interface).

region_1_id and **region_1_loc** contain the destination of the added item.

IMVX

When item is being deleted in source region due to being moved via API (or user interface).

region_1_id and **region_1_loc** contain the location of the item being deleted.

ISCR

When item deleted via API (or user interface) and item is being scrapped (not deleted and not production-stopped).

region_1_id and **region_1_loc** contain the location of the scrapped item

When an item becomes delayed and the region is configured to scrap (rather than detain) delayed items, a message is generated for the item being scrapped.

region_1_id and **region_1_loc** contain the location of the scrapped item.

ISTP

Production stop generated via API call.

region_1_id and **region_1_loc** contain the location of the item being stopped.

ISTT

When adding item via API and **prod_start_flag** is set:

region_1_id and **region_1_loc** contain the location of the event

When moving and SRC region is NULL:

region_1_id and **region_1_loc** contain the destination location.

Region-related Interested Process Updates

RORD

Generated when a region is reordered via API call or user interface.

region_1_id contains the region where the reorder occurred

modify_bitmask set to zero.

PRT Interface to External Software

The Production Tracking process provides Tracking Region data and Item tracking data to external application programs through the PRT Application Interface. PRT also accepts requests through the application interface to dynamically provide interested external application programs with

Item tracking data modifications. The external process must send PRT a message to indicate it is requesting to become an interested process in order to receive dynamic serialized Item updates.

PRT Objects for use in the Basic Control Engine

PRT Objects for use in the Basic Control Engine

The CIMPLICITY API for Production Tracking (PRT) has been implemented using an object model approach to allow the user to manipulate PRT information in a logical and consistent manner.

The Model below demonstrates the relationships between the 6 most important objects available for use with the Basic Control Engine in PRT calls. In code, each arrow below would be replaced with a "." to allow access to the child objects.

Example

"PRT.Group(0).Region(3).Item(0).Attr(6).Id"

PRT Object Model Member List

Prt Object Model Member List

The object attributes for the Basic Control Engine extensions for Production Tracking are:

- Prt
- PrtService
- PrtGroup
- PrtRegion
- PrtItem

Prt		
<p>Can be considered the root object of Production Tracking. From the root you can obtain a list of:</p> <ul style="list-style-type: none"> • PRT Services • PRT Groups • PRT Regions • PRT Items. <p>Obtaining a list of items would return every item currently in the PRT system. You can also locate an item, which will return all instances of the item within PRT.</p>		
Properties	Methods	Functions

Prt		
GroupCount (Read)	GetGroupList	Group
ItemClassId (Read/Write)	GetItemList	Item
ItemCount (Read)	GetRegionList	Region
ProjectId (Read/Write)	GetServiceList	Service
RegionCount(Read)	LocateItem	
ServiceCount (Read)		

PrtService		
<p>Contains information about a particular PRT Service. The object can be obtained from either:</p> <ul style="list-style-type: none"> • A list of PRT Services obtained from the Prt object • By initializing the object with the appropriate Service ID. <p>From a Service Object, you can obtain a list of all groups, regions and items contained within the service. You can also locate an item, which will return all instances of the item within PRT Service.</p>		
Properties	Methods	Functions
GroupCount (Read)	GetGroupList	Group
Id (Read/Write)	GetItemList	Item
ItemClassId (Read/Write)	GetRegionList	Region
ItemCount (Read)	LocateItem	
RegionCount (Read)		

PrtGroup		
<p>Contains information about the groups configured in your PRT System. Groups may be places on or off hold via this object. A group object may be obtained from and of the following:</p> <ul style="list-style-type: none"> • A Prt • A PrtService List • By initializing the ID (GroupId) property of the object <p>From a group object, you can obtain a list of all region and items contained within the group. You can also locate an item, which will return all instances of the item within PRT Group.</p>		
Properties	Methods	Functions
ExtHold (Read)	ClearExtHold	Item
ExtHoldReason (Read)	GetItemList	Region
Id (Read/Write)	GetRegionList	
ItemClassId (Read/Write)	LocateItem	
ItemCount (Read)	Modify	

PrtGroup		
ProjectId (Read/Write)	SetExtHold	
RegionCount (Read)		

PrtRegion		
<p>Contains a configured PRT region. Region status, capacity, etc, may be found in this object. A region object may be obtained from a list created by nand of the following:</p> <ul style="list-style-type: none"> • A Prt, PrtService • A PrtGroup object • By setting the ID to a known group ID. <p>From a region object, you can obtain a list of all items within the region. You can also locate an item, which will return all instances of the item within PRT Region.</p>		
Properties	Methods	Functions
Capacity (Read)	AdvanceModel	Item
Id (Read/Write)	ClearStatusBit	ItemType
ItemClassId (Read/Write)	Deleteltem	
ItemCount (Read)	GetData	
ItemTypeCount (Read)	GetItemList	
Loc (Read/Write)	Locateltem	
MaxLocation (Read)	Modify	
ProjectId (Read/Write)	Reorder	
Status (Read/Write)	SetPoints	
StatusPointId (Read)	SetStatusBit	
TotalItems (Read)	StatusBit	
TotalItemsPointId (Read/Write)		

PrtItem		
Contains a configured PRT Item. Item status and attributes may be accessed via this object.		
Properties	Methods	Functions
AttrCount (Read)	Add	Attr
Comment (Read/Write)	ClearModifyTime	
Delete (Read/Write)	ClearStatusBit	
EntryTime (Read)	Insert	
ExtHold(Read/Write)	Modify	

PrtItem		
ExtHoldGroupId (Read/Write)	MoveTo	
IntHold (Read/Write)	ProdStart	
IntHoldReason (Read/Write)	ProdStop	
ItemClassId (Read/Write)	RemoveAttr	
ItemId (Read/Write)	SetAttr	
ItemTypeId (Read/Write)	SetStatusBit	
ModifyTime (Read)		
ParentId (Read/Write)		
ProjectId (Read/Write)		
RefId (Read/Write)		
RegionId (Read/Write)		
RegionLoc (Read/Write)		
ServiceId (Read/Write)		
Status (Read/Write)		
UserId (Read/Write)		

PrtAttribute			
Contains an attribute for a PRT Item.			
Properties	Methods	Functions	
Id (Read/Write)	None	None	
Value (Read)			

Methods

PRTCheckRoute

Prt.GetGroupList (Method)

Syntax

Prt.GetGroupList

Description

Gets a list of all PRT groups. After calling the method, the GroupCount and Group routines may be used to access the returned information.)

Example

```
Dim prt as new Prt
' Get a list of groups from PRT
' and display them one by one in a message box.
prt.GetGroupList
for j = 0 to prt.GroupCount - 1
    MsgBox prt.Group(j).Id
next j
```

Prt.GetItemList (Method)

Syntax

Prt.GetItemList

Description

Gets a list of all items in PRT. After calling the method, the total item count can be obtained from **ItemCount** and the individual items may be accessed via the **Item** function.

To get the item list for a particular item class, set Prt.ItemClassId to the desired item class id before calling **Prt.GetItemList**. See Prt.ItemClassId for more details.

Example

```
Dim prt as new Prt
' Get a list of item from PRT
' and display them one by one in a message box.
prt.GetItemList
for j = 0 to prt.ItemCount - 1
    MsgBox prt.Item(j).ItemId
next j
```

Prt.GetRegionList (Method)

Syntax

Prt.GetRegionList

Description

Gets a list of all Tracking regions. After calling the method, the total region count can be obtained from **RegionCount** and the individual regions may be accessed via the **Region** function.

Example

```
Dim prt as new Prt
' Get a list of groups from PRT
' and display them one by one in a message box.
prt.GetRegionList
for j = 0 to prt.RegionCount - 1
    MsgBox prt.Region(j).Id
next j
```

Prt.GetServiceList (Method)

Syntax

Prt.GetServiceList

Description

Gets a list of the configured PRT services. After calling this method, the total number of services can be obtained from **ServiceCount** and the individual services may be accessed via the **Service** function.

Example

```
Dim prt as new Prt
' Get a list of services from PRT
' and display them one by one in a message box.
prt.GetServiceList
for j = 0 to prt.ServiceCount - 1
    MsgBox prt.Service(j).Id
next j
```

Prt.Group (Function)

Syntax

Prt.Group(index)

Description

Returns a group at a specified index from the list was found by a previous call to **GetGroupList**.

Example

```

Dim prt as new Prt
' Get a list of groups from PRT
' _____ and display them one by one in a message box.
prt.GetGroupList
for j = 0 to prt.GroupCount - 1
    _____ MsgBox prt.Group(j).Id
next j

```

Prt.GroupCount (Property Read)

Syntax

Prt.GroupCount

Description

Contains the number of Groups found by a prior call to **GetGroupList**.

Example

```

Dim prt as new Prt
' Get a list of groups from PRT
' _____ and display them one by one in a message box.
prt.GetGroupList
for j = 0 to prt.GroupCount - 1
    _____ MsgBox prt.Group(j).Id
next j

```

Prt.Item (Function)

Syntax

Prt.Item (index)

Description

Returns an item at the specified index from the list which was found by a previous call to **GetItemList**.

Example

```

Dim prt as new Prt
' Get a list of item from PRT
' _____ and display them one by one in a message box.
prt.GetItemList
for j = 0 to prt.ItemCount - 1
    _____ MsgBox prt.Item(j).ItemId
next j

```


Prt.ItemClassId (Property Read/Write)

Syntax

Prt.ItemClassId

Description

String property to set or get the Item Class ID in the Prt object. Used in conjunction with the Prt.GetItemList method to retrieve a list of all items of a particular item class in PRT.

Example

```
Dim prt as new Prt
' Get a list of item with a item class id of "CARRIER" from PRT
' and display them one by one in a message box.
prt.ItemClassId = "CARRIER"
prt.GetItemList
for j = 0 to prt.ItemCount - 1
    MsgBox prt.Item(j).ItemId
next j
```

Prt.ItemCount (Property Read)

Syntax

Prt.ItemCount

Description

Contains the number of items find by a previous **GetItemList** or **LocateItem** call.

Example

```
Dim prt as new Prt
' Get a list of item from PRT
' and display them one by one in a message box.
prt.GetItemList
for j = 0 to prt.ItemCount - 1
    MsgBox prt.Item(j).ItemId
next j
```

Prt.LocateItem (Method)

Syntax

Prt.LocateItem itemId\$, [refId\$ [, numRetries]]

Description

Locates a serialized Item within PRT. On return, the total Item count can be obtained from **ItemCount**, and the individual Items may be accessed via the **Item** function.

Parameter	Description
itemId\$	String. The Item ID to locate, may be an empty string if refId\$ is specified.
refId\$	String. The Reference ID of the item to locate, may omitted or an empty string if itemId\$ is specified.
NumRetries	Integer (optional) - The number of times to retry to find the item before failing. If an item is in transition from one tracking region to the next, there is a small period of time when it doesn't appear in either regions tracking queue. Defaults to 3 if not specified.

Example

```
Dim prt as new Prt
' Get a single item from PRT with its RefId
'      and display its ItemId in a message box.
PRT.LocateItem "",REFNUM$
MsgBox prt.Item(0).ItemId
```

Prt.ProjectId (Property Read/Write)

Syntax

Prt.ProjectId

Description

String property to set the Project which subsequent Production Tracking extensions will communicate with.

Example

```
Dim prt as new PRT
' Get a list of services from PRT '
'      and display them one by one in a message box
prt.ProjectId = "\\project1"
prt.GetServiceList
for j = 0 to prt.ServiceCount - 1
'      MsgBox prt.Service(j).Id
next j
```

Prt.Region (Function)

Syntax

Prt.Region(index)

Description

Returns a region at the specified index from the list which was found by a previous call to **GetRegionList**.

Example

```
Dim prt as new Prt
' Get a list of groups from PRT
' and display them one by one in a message box.
prt.GetRegionList
for j = 0 to prt.RegionCount - 1
    MsgBox prt.Region(j).Id
next j
```

Prt.RegionCount (Property Read)

Syntax

Prt.RegionCount

Description

Contains the count of the number of Tracking Regions found by a prior **GetRegionList** call.

Example

```
Dim prt as new Prt
' Get a list of groups from PRT
' and display them one by one in a message box.
prt.GetRegionList
for j = 0 to prt.RegionCount - 1
    MsgBox prt.Region(j).Id
next j
```

Prt.Service (Function)

Syntax

Prt.Service

Description

Returns a Service at the specified index from the list which was found by a previous call to **GetServiceList**.

Example

```
Dim prt as new Prt
' Get a list of services from PRT
' and display them one by one in a message box.
prt.GetServiceList
for j = 0 to prt.ServiceCount - 1
    MsgBox prt.Service(j).Id
next j
```

Prt.ServiceCount (Property Read)

Syntax

Prt.ServiceCount

Description

Contains a count of the number of Service found with a prior **GetServiceList** call.

Example

```
Dim prt as new Prt
' Get a list of services from PRT
' and display them one by one in a message box.
prt.GetServiceList
for j = 0 to prt.ServiceCount - 1
    MsgBox prt.Service(j).Id
next j
```

PrtAttribute.Id (Property Read/Write)

Syntax

PrtAttribute.Id

Description

String property contains the attribute name of an Item ID .

Example

```
Dim prt as new Prt
' Get a list of items from PRT and display the first item's
' attribute names and values one by one in a message box.
prt.GetItemList
for j = 0 to prt.Item(0).AttrCount - 1
    MsgBox prt.Item(0).Attr(j).Id &"="& prt.Item(0).Attr(j).Value
```

```
next j
```

PrtAttribute.Value (Property Read)

Syntax

PrtAttribute.Value

Description

String property contains the attribute value of an Item ID.

Example

```
Dim prt as new Prt
' Get a list of items from PRT and display the first item's
' attribute names and values one by one in a message box.
prt.GetItemList
for j = 0 to prt.Item(0).AttrCount - 1
  MsgBox prt.Item(0).Attr(j).Id &"=" & prt.Item(0).Attr(j).Value
next j
```

PrtCheckRoute (Method)

Syntax

PrtCheckRoute sourceRegionId\$, destRegionId\$, itemtypeId\$

Description

Validates that an Item of a specific type is configured to move from a Source Tracking region to a Destination Tracking Region. Route validation is performed locally by the API from configuration information that was read into the API's internal schema during initialization.

If success the routine returns.

If failure an error is generated and should be caught using the **on error goto Syntax**.

Parameter	Description
SourceRegionId\$	String. The source region of the item.
DestRegionId\$	String. The destination region of the item.
ItemtypeId\$	String. The item type ID .

Example

```
RtFail = 0
On Error goto RouteFail
```

```

PrtCheckRoute "Schedule", "Production", "MARKV"
If RtFail = 0 then msgbox "Item Type Valid for this Route."
If RtFail = 1 then msgbox "Item Type Invalid for this Route!"
Exit Sub
RouteFail:
    RtFail = 1
    Resume Next

```

PrtGroup

This object contains information on the groups configured in your PRT System.

Groups may be places on or off hold via this object. A group object may obtained from a Prt or PrtService List or by initializing the ID (GroupId) property of the object.

From a group object, you can obtain a list of all region and items contained within the group. You can also locate an item, which will return all instances of the item within PRT Group.

PrtGroup.ClearExtHold (Method)

Syntax

PrtGroup.ClearExtHold

Description

Clears an external hold for this tracking group. This setting will be updated to PRT after a call to **Modify**.)

Example

```

Dim prt as new Prt
' Get a list of groups from PRT
' and clear each one's External Hold status bit.
prt.GetGroupList
for j = 0 to prt.GroupCount - 1
    prt.Group(j).ClearExtHold
    prt.Group(j).Modify
next j

```

PrtGroup.ExtHold (Property Read)

Syntax

PrtGroup.ExtHold

Description

Boolean property contains the setting of External Hold for the Tracking Group.

Example

```
Dim prt as new Prt
' Get a list of groups from PRT and display their External Hold
' status bits one by one in a message box.
prt.GetGroupList
for j = 0 to prt.GroupCount - 1
    MsgBox prt.Group(j).ExtHold
next j
```

PrtGroup.ExtHoldReason (Property Read)

Syntax

PrtGroup.ExtHoldReason

Description

String property contains the reason for setting External Hold for this Tracking Group.

Example

```
Dim prt as new Prt
' Get a list of groups from PRT and display their External Hold
' status bit Reasons one by one in a message box.
prt.GetGroupList
for j = 0 to prt.GroupCount - 1
    MsgBox prt.Group(j).ExtHoldReason
next j
```

PrtGroup.GetItemList (Method)

Syntax

PrtGroup.GetItemList

Description

Gets a list of all items in the Tracking Group. On return, the total item count can be obtained from **PrtGroup.ItemCount** and the individual items may be accessed via the **PrtGroup.Item** function.

To get the item list for a particular item class, set **PrtGroup.ItemClassId** to the desired item class id before calling **PrtGroup.GetItemList**.

Example

```

Dim main as new Prtgroup
main.Id = "MAIN"
' Get a list of items from group "MAIN"
' and display them one by one in a message box.
main.GetItemList
for j = 0 to main.ItemCount - 1
    MsgBox main.Item(j).ItemId
next j

```

PrtGroup.GetRegionList (Method)

Syntax

PrtGroup.GetRegionList

Description

Gets a list of Tracking regions contained in the Tracking Group. On return, the total number of regions can be obtained from **PrtGroup.RegionCount** and the individual items may be accessed via the **PrtGroup.Region** function.

Example

```

Dim main as new PrtGroup
main.Id = "MAIN"
' Get a list of regions from a PRT group
' and display them one by one in a message box.
main.GetRegionList
for j = 0 to main.RegionCount - 1
    MsgBox main.Region(j).Id
next j

```

PrtGroup.Id (Property Read/Write)

Syntax

PrtGroup.Id

Description

String property contains the Group Id of the object.

Example

```

Dim main as new PrtGroup
main.Id = "MAIN"
Get a list of regions from a PRT group
' and display them one by one in a message box.

```



```

main.GetRegionList
for j = 0 to main.RegionCount - 1
    MsgBox main.Region(j).Id
next j

```

PrtGroup.Item (Function)

Syntax

PrtGroup.Item(index)

Description

Returns an item at the specified index from the list which was found by a previous call to **GetItemList**.

Example

```

Dim main as new Prtgroup
main.Id = "MAIN"
' Get a list of items from group "MAIN"
' and display them one by one in a message box.
main.GetItemList
for j = 0 to main.ItemCount - 1
    MsgBox main.Item(j).ItemId
next j

```

PrtGroup.ItemClassId (Property Read/Write)

Syntax

PrtGroup.ItemClassId

Description

String property to set or get the Item Class ID in the PrtGroup object. Used in conjunction with the PrtGroup.GetItemList method to retrieve a list of all items of a particular item class in a Tracking Group.

Example

```

Dim main as new Prtgroup
main.Id = "MAIN"
' Get a list of items with an item class id "CARRIER" from the
' group "MAIN" and display them one by one in a message box.
main.ItemClassId = "CARRIER"
main.GetItemList
for j = 0 to main.ItemCount - 1
    MsgBox main.Item(j).ItemId

```

```
next j
```

PrtGroup.ItemCount (Property Read)

Syntax

PrtGroup.ItemCount

Description

Integer property contains the number of items find by a previous **GetItemList** or **LocateItem** call.

Example

```
Dim main as new Prtgroup
main.Id = "MAIN"
' Get a list of items from group "MAIN"
' and display them one by one in a message box.
main.GetItemList
for j = 0 to main.ItemCount - 1
    MsgBox main.Item(j).ItemId
next j
```

PrtGroup.LocateItem (Method)

Syntax

PrtGroup.LocateItem itemId\$, refId\$ [, numRetries]

Description

Locates a serialized item within the tracking group. Upon return, the total item count may be obtained from **ItemCount**, and the individual items may be accessed via the **Item** function.

Parameter	Description
itemId\$	String. The Item ID to locate, may be an empty string if refId\$ is specified.
refId\$	String. The Reference ID of the item to locate, may be omitted or an empty string if itemId\$ is specified.
numRetries	Integer (optional) - The number of times to retry to find the item before failing. If an item is in transition from one tracking region to the next, there is a small period of time when it doesn't appear in either region's tracking queue. Defaults to 3 if not specified.

Example

```
Dim main as new PrtGroup
main.Id = "MAIN"
' Get a single item from PRT with its RefId
```

```
'      and display its ItemId in a message box.
PRT.LocateItem "",REFNUM$
MsgBox prt.Item(0).ItemId
```

PrtGroup.Modify (Method)

Syntax

PrtGroup.Modify

Description

Updates the PRT Tracking Group with the current contents of this object. Used to set or clear External Hold.

Example

```
Dim prt as new Prt
' Get a list of groups from PRT
'      and clear each one's External Hold status bit.
prt.GetGroupList
for j = 0 to prt.GroupCount - 1
    prt.Group(j).ClearExtHold
    prt.Group(j).Modify
next j
```

PrtGroup.ProjectId (Property Read/Write)

Syntax

PrtGroup.ProjectId

Description

String property to set the project that subsequent Production Tracking extensions will communicate with.

Example

```
Dim prt as new PrtGroup
' Get a list of services from PRT
'      and display them one by one in a message box.
prt.ProjectId = "\\project1"
prt.GetRegionList
for j = 0 to prt.RegionCount - 1
    MsgBox prt.Region(j).Id
next j
```

PrtGroup.Region (Function)

Syntax

PrtGroup.Region(index)

Description

Returns a region at the specified index from the list that was found by a previous call to **GetRegionList**.

Example

```
Dim main as new PrtGroup
main.Id = "MAIN"
' Get a list of regions from a PRT group
' and display them one by one in a message box.
main.GetRegionList
for j = 0 to main.RegionCount - 1
    MsgBox main.Region(j).Id
next j
```

PrtGroup.RegionCount (Property Read)

Syntax

PrtGroup.RegionCount

Description

Integer property contains a count of the number of Tracking Regions found by a prior **GetRegionList** call.

Example

```
Dim main as new PrtGroup
main.Id = "MAIN"
' Get a list of regions from a PRT group
' and display them one by one in a message box.
main.GetRegionList
for j = 0 to main.RegionCount - 1
    MsgBox main.Region(j).Id
next j
```

PrtGroup.SetExtHold (Method)

Syntax

PrtGroup.SetExtHold reason\$

Description

Sets an External Hold for this tracking group. The setting will be updated to PRT after a call to **Modify**.

Parameter	Description
reason\$	String. The reasons for placing the group on hold.

Example

```
Dim prt as new Prt
Reason$ = "-----"
' Get a list of groups from PRT
' and initialize their External Hold status bits and Reasons.
prt.GetGroupList
for j = 0 to prt.GroupCount - 1
    prt.Group(j).SetExtHold Reason$ & " " & j
    prt.Group(j).Modify
next j
```

PrtItem.Add (Method)

Syntax

PrtItem.Add [extProcFlag]

Description

Adds an individual item at the same location where another item currently resides. The **PrtItem.RegionLoc** indicates the location at which the item will be added. The **RegionLoc** can be specified as **PRT_FIRST**, **PRT_LAST**, or a valid region location within the region.

Parameter	Description
ext_proc_flag	Boolean. External process flag, if set to TRUE; get all information from an external process. If not provided defaults to FALSE.

If the new item is not being copied from a previously created item, the following **PRTItem** object methods must be set for the add to be successful:

. **RefId** or **.ItemId** (if this is a serialized item)

.Itemtype

.RegionId

.Status

.IntHold (usually false)

.ExtHold (usually false)

.RegionLoc

Example

```
dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.IntHold false
NEWITEM.ExtHold false
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
'Upload to Tracking Database
NEWITEM.Add
```

PrtItem.Attr (Function)

Syntax

PrtItem.Attr(index)

PrtItem.Attr(attrib\$)

Description

Prototype 1.

Returns an Item Attribute at the specified index from the list which was found by a previous call to **GetItemList**. The index must be in range from **0** to **AttrCount**. If the index is out of range, an error is generated.

Prototype 2.

This function returns the Item Attribute associated with the specified PRT Attribute or an empty string if the Item does not have the Attribute.

Example

```
Dim prt as new Prt
```

```
'Get a list of items from PRT and display each of their
'attributes one by one in a message box.
prt.GetItemList
For i = 0 To prt.ItemCount - 1
    For j = 0 To prt.Item(i).AttrCount - 1
        MsgBox prt.Item(i).Attr(j).Id & " = " &
        prt.Item(i).Attr(j).Value
    Next j
Next i
'Get a list of items from PRT again but display their COLOR only
For i = 0 To prt.ItemCount - 1
    For j = 0 To prt.Item(i).AttrCount - 1
        MsgBox prt.Item(i).Attr("COLOR")
    Next j
Next i
```

PrtItem.AttrCount (Property Read)

Syntax

PrtItem.AttrCount

Description

Integer property contains a count of the number of Attributes for the Item.

Example

```
Dim prt as new Prt
'Get a list of items from PRT and display each of their
'attributes one by one in a message box.
prt.GetItemList
For i = 0 To prt.ItemCount - 1
    For j = 0 To prt.Item(i).AttrCount - 1
        MsgBox prt.Item(i).Attr(j).Id & " = " & prt.Item(i).Attr(j).Value
    Next j
Next i
'Get a list of items from PRT again but display their COLOR only
For i = 0 To prt.ItemCount - 1
    For j = 0 To prt.Item(i).AttrCount - 1
        MsgBox prt.Item(i).Attr("COLOR")
    Next j
Next i
```

PrtItem.ClearModifyTime (Method)

Syntax

PrtItem.ClearModifyTime

Description

Clears the modify time on the object. When calling **Modify**, PRT will reject the request if the item's last modify time is less than the current modify time of the item in PRT. Calling **ClearModifyTime** overrides this behavior. This should be used with care, since you may overwrite properties of the object that the user may have manipulated through prt_ui, etc.

Example

```
Dim prt as new Prt
' Get a list of items from PRT
' and initialize the modify time for each.
prt.GetItemList
for j = 0 to prt.ItemCount - 1
    prt.Item(j).ClearModifyTime
    prt.Item(j).Modify
next j
```

PrtItem.ClearStatusBit (Property Read/Write)

Syntax

PrtItem.ClearStatusBit Bit%

Description

Clears the specified bit (0 through 31) of the status mask.)

Parameter	Description
Bit	Integer. The bit position 0 - 31 to clear.

Example

```
Dim prt as new Prt
n% = 17
' Get a list of items from PRT and initialize a particular
' status bit for each one.
prt.GetItemList
for j = 0 to prt.ItemCount - 1
    prt.Item(j).ClearStatusBit n%
    prt.Item(j).Modify
next j
```

PrtItem.Comment (Property Read/Write)

Syntax

PrtItem.Comment

Description

String property to set or get the comment associated with the item for an impending **Modify** or **Add**. The comment is written to the PRT log file as the modifications are sent to the object.

Example

```
dim NEWITEM as new prtitem
' Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.IntHold false
NEWITEM.ExtHold false
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
NEWITEM.Comment = "New item created."
' Upload to Tracking Database
NEWITEM.Add
```

PrtItem.Delete (Property Read/Write)

Syntax

PrtItem.Delete disposition [, comment\$]

Description

Deletes the item from the tracking model. A disposition must be supplied, and a comment is optional.

Parameter	Description
disposition	Integer. Reason for deleting. See table below for manifest constants.
comment\$	String (optional) - The optional comment to send to the log file.
Disposition	Description
PRT_SCRAP	Scrap the item.
PRT_DELETE	Delete the item.
PRT_NO_CHECK_SCRAP	Scrap the item overriding region locking status
PRT_NO_CHECK_DELETE	Delete the item overriding region locking status

Example

```
Dim prt as new Prt
' Get a single item from PRT with its RefId and delete it.
PRT.LocateItem "",REFNUM$
prt.Item(0).Delete PRT_NO_CHECK_DELETE, "Item Removed"
```

PrtItem.EntryTime (Property Read)

Syntax

PrtItem.EntryTime

Description

Date property contains the date and time at which the item entered the current region.

Example

```
Dim Region as new prt.Region
Region.Id = "Detainment"
'Get a list of items in a region and display their entry times
' one by one in a message box.
Region.GetItemList
For j = 0 to Region.ItemCount - 1
    msgbox Region.Item(j).EntryTime
Next j
```

PrtItem.ExtHold (Property Read/Write)

Syntax

PRTItem.ExtHold [True/False]

Description

Boolean property to set or get the state of External Hold for the Item.

Example

```
dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.Inthold false
NEWITEM.Exthold false
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
```

```
'Upload to Tracking Database
NEWITEM.Add
```

PrtItem.ExtHoldGroupId(Property Read/Write)

Syntax

PrtItem.ExtHoldGroupId

Description

String property to set or get the Group ID which has placed an External Hold on the item.

Example

```
dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.IntHold false
NEWITEM.ExtHold true
NEWITEM.ExtHoldGroupId = "SHIPPING"
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
'Upload to Tracking Database
NEWITEM.Add
```

PrtItem.Insert (Method)

Syntax

PrtItem.Insert [extProcFlag]

Description

Inserts an Item into the Tracking Region queue. If an Item is to be inserted into the queue at a particular location (other than at the head or tail of the queue), the Region location of the Item to be inserted ahead of is specified, i.e. specifying **regionLoc = 1**, is equivalent to specifying **regionLoc = PRT_FIRST**.

Parameter	Description
ext_proc_flag	Boolean. External process flag, if set to TRUE; get all information from an external process. If not provided defaults to FALSE.

Example

```

dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = 3
NEWITEM.IntHold false
NEWITEM.ExtHold false
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
'Upload to Tracking Database
NEWITEM.Insert

```

PrtItem.IntHold (Property Read/Write)

Syntax

PrtItem.IntHold

Description

Boolean property to set or get the state of Internal Hold on the Item.

Example

```

dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.IntHold false
NEWITEM.ExtHold false
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
'Upload to Tracking Database
NEWITEM.Add

```

guide: Guidelines for internal holds

- If you add an item with the internal hold bit set to TRUE, you must give it a reason ([PrtItem.IntHoldReason \(page 723\)](#)).
- Do not set an Internal Hold System Attribute using the .SetAttr method, PrtItem.SetAttr.

PrtItem.SetAttr is a method to create/set Standard Attributes, not System attributes.

- Use PrtItem.IntHold and PrtItem.IntHoldReason to get and set internal holds for an item.

Example

```
MyPRTItem.IntHold = True
```

```
MyPRTItem.IntHoldReason = "Held by Script"
```

```
MyPRTItem.Modify
```

```
myHold = MyPRTItem.IntHold
```

```
myHoldReason = MyPRTItem.IntHoldReason
```

The following statement will create a standard attribute with the name INTERNALHOLD, instead of modifying [PRT Item's Internal Hold \(page 217\)](#) system attribute.

```
MyPRTItem.SetAttr "INTERNALHOLD", "1"
```

- All the function blocks that deal with extended attributes can be provided INTERNALHOLD text to read the value of internal hold.

Values are:

0	Internal Hold is not set
1	Internal Hold is set

PrtItem.IntHoldReason (Property Read/Write)

Syntax

```
PrtItem.IntHoldReason
```

Description

String property to set or get the Internal Hold Reason for the Item.

Example

```
dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.IntHold true
NEWITEM.IntHoldReason = "Insufficient Parts"
NEWITEM.ExtHold false
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
```

```
'Upload to Tracking Database
NEWITEM.Add
```

PrtItem.ItemClassId (Property Read/Write)

Syntax

PrtItem.ItemClassId

Description

String property to set or get the Item Class ID of the Item.

Example

```
dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.IntHold false
NEWITEM.ExtHold false
NEWITEM.ItemClassId = CLASS$
NEWITEM.Status = 0
'Upload to Tracking Database
NEWITEM.Add
```

PrtItem.ItemId (Property Read/Write)

Syntax

PrtItem.ItemId

Description

String property to set or get the Item ID of the Item in the region. Setting has does not update PRT until the **Modify** method is invoked.

Example

```
dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.IntHold false
NEWITEM.ExtHold false
```

```
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
'Upload to Tracking Database
NEWITEM.Add
```

PrtItem.ItemTypeId (Property Read/Write)

Syntax

PrtItem.ItemTypeId

Description

String property to set or get the Item Type ID of the Item.

Example

```
dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.IntHold false
NEWITEM.ExtHold false
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
'Upload to Tracking Database
NEWITEM.Add
```

PrtItem.Modify (Method)

Syntax

PrtItem.Modify

Description

Modifies the Item. All modified attributes and properties of the item are sent to the PRT Data Server. An error will be generated if the PRT Data Server rejects the request.

Example

```
Dim prt as new Prt
n% = 17
' Get a list of items from PRT and initialize a particular
' status bit for each one.
prt.GetItemList
for j = 0 to prt.ItemCount - 1
```

```

prt.Item(j).ClearStatusBit n%
prt.Item(j).Modify
next j

```

PrtItem.ModifyTime (Property Read)

Syntax

PrtItem.ModifyTime

Description

Date property contains the time at which the Item was modified. When the user modifies the Item, PRT checks this modify time against the last modify time of the Item. If the modify time is older than the current modify time of the Item in PRT, PRT will not allow the user to change the Item. This is to prevent two processes from modifying an Item without each being aware of the other's modification. To override this behavior, call **ClearModifyTime** which will disable this checking.

Example

```

Dim Region as new prt.Region
Region.Id = "Detainment"
'Get a list of items in a region and display their modify times
' one by one in a message box.
Region.GetItemList
For j = 0 to Region.ItemCount - 1
    msgbox Region.Item(j).ModifyTime
Next j

```

PrtItem.MoveTo (Method)

Syntax

PrtItem.MoveTo destRegionId\$, destLoc, insertFlag [, comment\$]

Description

Moves an item from its current location to a new location in a region (could be the same region.)

Parameter	Description
destRegionId\$	String. Destination Region ID. The source Region ID is contained in the item object already.
destLoc	Integer. Location to move to in destination region. Sequence number of the item in the destination tracking region. Specified as PRT_FIRST (EXIT) or PRT_LAST (Entry), or a number from 1 to n, where n is the number of items in the queue.

Parameter	Description
insert_flag	Boolean. If true, the moved item is to be inserted in the destination region queue before the item located at the specified destination region location, unless the manifest constant last is specified, in which case the item is inserted after (behind) the last item currently present in the region.
comment\$	String (optional). An optional comment to write to the PRT log file.

Example

```
Dim Region as new PRTRREGION
Region.Id = SRC$
Region.GetItemList
' Set timestamp attributes for the head item in a region, and
' then move the item to another region.
DSUF$ = "_DATE"
TSUF$ = "_TIME"
Region.Item(0).Setattr SRC$ & DSUF$, DATE$
Region.Item(0).Setattr SRC$ & TSUF$, TIME$
Region.Item(0).Modify
Region.Item(0).Moveto DESTREGION$, RegionLOC%, 0
```

PrtItem.ParentId (Property Read/Write)

Syntax

PrtItem.ParentId

Description

String property to set or get the Parent ID of the item. Parent IDs are only valid for non-serialized items.

Example

```
Dim Region as new PRTRRegion
Region.Id = "DETAINMENT"
' Get a list of all the items which exist in 1 location of a
' region and display the ParentIds one by one in a message box.
Region.Loc = 0
Region.GetItemList
for j = 0 to Region.ItemCount - 1
    msgbox Region.Item(j).ParentId
next j
```

PrtItem.ProdStart (Method)

Syntax

PrtItem.ProdStart [extProcFlag]

Description

Initiates production start for an item in a Tracking Region. It is equivalent to the **Add** method, but logs differently.

Parameter	Description
ext_proc_flag	Boolean. External process flag, if set to TRUE, get all information from an external process. If not provided defaults to FALSE.

Example

```
dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.IntHold false
NEWITEM.ExtHold false
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
'Upload to Tracking Database
NEWITEM.ProdStart
```

PrtItem.ProdStop (Method)

Syntax

PrtItem.ProdStop

Description

Removes an Item from the Production Tracking system as the result of its exiting out of a region along a route that has no destination region. The specified Item must currently reside at a source region associated with such a route. If the Item cannot legitimately transition out of the system along such a route, an error is returned to the calling routine.

Example

```
Dim Region as new prt.Region
Region.Id = "Detainment"
'Get a list of items in a region and remove each of them from
' tracking one by one.
Region.GetItemList
For j = 0 to Region.ItemCount - 1
    Region.Item(j).ProdStop
```

```
Next j
```

PrtItem.ProjectId (Property Read/Write)

Syntax

PrtItem.ProjectId

Description

String property to set the project with which subsequent Production Tracking extensions will communicate.

Example

```
Dim prt as new PrtItem
' Get a list of services from PRT and display them one by one
'   in a message box
prt.ProjectId = "\\project1"
for j = 0 to 19
    index$ = CSTR(j)
    prt.SetAttr "NAME" & index$, "Val" & index$
next j
```

PrtItem.RefId (Property Read/Write)

Syntax

PrtItem.RefId

Description

String property to set or get the Reference ID of the Item.

Example

```
dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.RefId = RefNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.IntHold false
NEWITEM.ExtHold false
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
'Upload to Tracking Database
```

```
NEWITEM.ProdStart
```

PrtItem.RegionId (Property Read/Write)

Syntax

```
PrtItem.RegionId
```

Description

String property to set or get the Region ID of the Item. Setting the Region ID is not updated into PRT until the Item is modified.

Example

```
dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.RefId = RefNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.IntHold false
NEWITEM.ExtHold false
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
'Upload to Tracking Database
NEWITEM.Add
```

PrtItem.RegionLoc (Property Read/Write)

Syntax

```
Point.RegionLoc
```

Description

Integer property to set or get the location of the Item in the region. Setting does not update PRT until the **Modify** method is invoked.

Example

```
dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.RefId = RefNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
```

```

NEWITEM.IntHold false
NEWITEM.ExtHold false
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
'Upload to Tracking Database
NEWITEM.Add

```

PrtItem.RemoveAttr(Method)

Syntax

PrtItem.RemoveAttr attrib\$

Description

Removes an Attribute from the Item. If the Attribute is not found an error is generated.

Example

```

Dim Region as new prt.Region
Region.Id = "Detainment"
'Get a list of items in a region and remove an attribute from
' each of them one by one.
Region.GetItemList
For j = 0 to Region.ItemCount - 1
    Region.Item(j).RemoveAttr "BODYCOLOR"
Next j

```

PrtItem.ServiceId (Property Read/Write)

Syntax

PrtItem.ServiceId

Description

String property to get or set the Service ID associated with the Item.

Example

```

dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.ServiceId = "PRT_DC"
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.IntHold false
NEWITEM.ExtHold false

```

```

NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
'Upload to Tracking Database
NEWITEM.Add

```

PrtItem.SetAttr (Method)

Syntax

PrtItem.SetAttr attrib\$, value\$

Description

Sets an Attribute to a value for the Item. If the Attribute exists for the Item, it is overwritten. If the Attribute does not exist for the Item a new Attribute is created. PRT will be updated with the modified Attribute on the next **Modify** method.

Example

```

Dim Region as new PRTREGION
Region.Id = SRC$
Region.GetItemList
'Set timestamp attributes for the head item in a region, and
' then move the item to another region.
DSUF$ = "_DATE"
TSUF$ = "_TIME"
Region.Item(0).Setattr SRC$ & DSUF$, DATE$
Region.Item(0).Setattr SRC$ & TSUF$, TIME$
Region.Item(0).Modify
Region.Item(0).Moveto DESTREGION$, RegionLOC%, 0

```

PrtItem.SetStatusBit (Method)

Syntax

PrtItem.SetStatusBit Bit%

Description

Sets a specific bit (0 through 31) of the Status Mask for an Item.

Example

```

Dim CurPrt as new prtregion
CurPrt.Id = "EXITREGION"
CurPrt.GetItemList
'Get a list of items in a region, check each one's status bit,
' then set it if it's not already set.
If CurPrt.Item(0).StatusBit(J%) = FALSE Then

```

```

    CurPrt.Item(I).SetStatusBit J%
    CurPrt.Item(I).Modify
End If

```

PrtItem.Status (Property Read/Write)

Syntax

PrtItem.Status

Description

Long property to get or set the Status Mask of the Item.

Example

```

dim NEWITEM as new prtitem
'Initialize the new item structure
NEWITEM.ItemId = IDNUM$
NEWITEM.GroupId = Groupname$
NEWITEM.RegionId = Regname$
NEWITEM.RegionLoc = PRT_LAST
NEWITEM.IntHold false
NEWITEM.ExtHold false
NEWITEM.ItemTypeId = TYPE$
NEWITEM.Status = 0
'Upload to Tracking Database
NEWITEM.Add

```

PrtItem.UserId (Property Read/Write)

Syntax

PrtItem.UserId

Description

String property to set or get the User ID associated with the Item. The User ID is recorded into the PRT log when an Item is modified.

Example

```

Dim prt as new Prt
n% = 17
' Get a list of items from PRT and initialize a particular
' status bit for each one, setting the object to log the
' user id performing the action.
prt.GetItemList
for j = 0 to prt.ItemCount - 1

```

```

prt.Item(j).ClearStatusBit n%
prt.Item(j).UserId = "ADMINISTRATOR"
prt.Item(j).Modify
next j

```


PrtRegion.AdvanceModel (Method)

Syntax

PrtRegion.AdvanceModel destRegionId\$ [, comment\$ [, userId\$]

Description

Advances the next Item Carrier to leave the region to a destination region.

 **Note:** This call moves all items at the FIRST (EXIT) location in the source region to the LAST (ENTRY) location in the destination region.

Parameter	Description
destRegionId\$	String. Destination Region ID; region the item is moving into.
comment\$	String. (Optional) Comment to place in PRT log file.
userId\$	String. (Optional) User ID to place into PRT log file.

Example

```

Dim Region as new PrtRegion
Region.Id = "Schedule"
'Determine if there are any items in the region,
' and advance the first one if there are.
Region.GetItemList
If Region.ItemCount > 0 then
    Region.AdvanceModel "Production"
End If

```

PrtRegion.Capacity (Property Read)

Syntax

PrtRegion.Capacity

Description

Integer property to return the region capacity.

Example


```

Dim Prt as new Prt
Prt.GetRegionList
For i = 0 to Prt.RegionCount - 1
    MsgBox Prt.Region(i).Capacity
Next i

```

PrtRegion.ClearStatusBit (Method)

Syntax

PrtRegion.ClearStatusBit b%

Description

Clears a particular status bit in the regions status mask.

Comments

To obtain the region's status mask, use the **PrtRegion.GetData** method. To update the Region's status mask to PRT, use the **PrtRegion.Modify**.

Bits 0-15 are for User Use. Bits 16-31 are for GE Intelligent Platforms use. The following constants may be used to access GE Intelligent Platforms specific bits.

Constants	Description
PRT_REGION_FULL	Region Full Bit
PRT_REGION_EXT_HOLD	Region External Hold Bit
PRT_REGION_IN_LOCK	Region in Lock Bit
PRT_REGION_OUT_LOCK	Region Out Lock Bit
PRT_REGION_OUT_OF_SEQ	Region Out of Sequence Bit
PRT_REGION_CAP_EXCEEDED	Region Capacity Exceeded Bit
PRT_REGION_NORMAL	Region Normal Bit

Example

```

Dim Prt as new Prt
n% = 17
' Get a list of items from PRT and initialize a particular
' status bit for each one.
Prt.GetRegionList
for j = 0 to prt.ItemCount - 1
    Prt.Region(j).ClearStatusBit n%
    Prt.Region(j).Modify

```

next j

PrtRegion.DeleteItem (Method)

Syntax

PrtRegion.DeleteItem regionLoc, disposition [, comment\$]

Description

Deletes all items in a location whose number was defined as a parameter.

Parameter	Description
regionLoc	Integer. The location of the item in the region.
disposition	Integer. The disposition of the deleted item. See table below.
comment\$	String. (Optional) - an optional comment to place in the PRT log.
Disposition	Description
PRT_SCRAP	Scrap the item.
PRT_DELETE	Delete the item.
PRT_NO_CHECK_SCRAP	Scrap the item overriding region locking status.
PRT_NO_CHECK_DELETE	Delete the item overriding region locking status.

Example

```
Dim Region as new PrtRegion
```

```
Region.Id = "PRODUCTION"
```

```
Region.GetData
```

```
For i = Region.MaxLocation to 1 Step -1
```

```
    Region.DeleteItem i, PRT_DELETE
```

```
Next i
```

PrtRegion.GetData (Method)

Syntax

PrtRegion.GetData

Description

Returns region information such **Status**, **TotalItems**, **Capacity**, **StatusPointId**, and **TotalItemsPointId**, which then become available through the object.

Example

```
'Example of PrtRegion.StatusBit Function using a constant to
' get the Region Full status bit and display it's status in
' a message box
Dim DestRegion as new PrtRegion
DestRegion.Id = "Region_Name" 'Replace Region Name
DestRegion.GetData
If DestRegion.StatusBit(PRT_REGION_FULL) Then
    MsgBox "Region Full"
Else
    MsgBox "Region Not Full"
End If
```

PrtRegion.GetItemList (Method)

Syntax

PrtRegion.GetItemList

Description

Retrieves a list of items for the current region.

Comments

If you call **PrtRegion.GetItemList** before setting **PrtRegion.Loc** you get the item list for all locations.

To get the item list for a current region location, set **PrtRegion.Loc** to that location number before calling **PrtRegion.GetItemList**.

If you want to get the item list for all locations and **PrtRegion.Loc** currently contains a location number, set **PrtRegion.Loc** to **PRT_ALL_LOCATIONS** before calling **PrtRegion.GetItemList**.

To get the item list for a particular item class, set **PrtRegion.ItemClassId** to the desired item class id before calling **PrtRegion.GetItemList**.

Example

```
Dim Region As New PRtRegion
Region.Id = "PRODUCTION"
Region.GetItemList
'Get all items in a region and delete those items with a
' specific ItemTypeId.
```

```

For i = Region.ItemCount - 1 To 0 Step -1
    If Region.Item(i).ItemTypeId = "MARKVII" then
        Region.Item(i).Delete PRT_DELETE, "Obsolete Type" ' Correct -
    Region.Item(i).Delete will work.
    End If
Next

```

See Also

PrtRegion.Loc

PrtRegion.Id (Property Read/Write)

Syntax

PrtRegion.Id

Description

String property to get or set the Region ID of the object.

Example

```

Dim Region as new PrtRegion
Region.Id = "PRODUCTION"
Region.GetItemList
'Get all items in a region and delete those items with a
' specific ItemTypeId.
For i = 0 to Region.ItemCount - 1
    If Region.Item(i).ItemTypeId = "MARKVII" then
        Region.DeleteItem i, PRT_DELETE, "Obsolete Type"
    End If
Next i

```

PrtRegion.Item (Function)

Syntax

PrtRegion.Item(index)

Description

Returns **PrtItem**. To return a specific item by index from a previous call to **GetItemList**.

Example

```

Dim Region as new PrtRegion
Region.Id = "PRODUCTION"
Region.GetItemList

```

```
'Get all items in a region and delete those items with a
' specific ItemTypeId.
For i = 0 to Region.ItemCount - 1
  If Region.Item(i).ItemTypeId = "MARKVII" then
    Region.DeleteItem i, PRT_DELETE, "Obsolete Type"
  End If
Next I
```

PrtRegion.ItemClassId (Property Read/Write)

Syntax

PrtRegion.ItemClassId

Description

String property to set or get the Item Class ID in the PrtRegion object. Used in conjunction with the PrtRegion.GetItemList method to retrieve a list of all items of a particular item class in a Tracking Region.

Example

```
Dim Region as new Prtregion
Region.Id = "PRODUCTION"
Region.ItemClassId = "CARRIER"
Region.GetItemList
' Get a list of items with an item class id "CARRIER" from the
' region "PRODUCTION" and display them one by one in a
' message box.
Region.ItemClassId = "CARRIER"
Region.GetItemList
for j = 0 to Region.ItemCount - 1
  MsgBox Region.Item(j).ItemId
next j
```

PrtRegion.ItemCount (Property Read)

Syntax

PrtRegion.ItemCount

Description

Returns the total number of items in a region, regardless of the number of locations or number of items per location.

This integer property gets the number of items found by a previous **GetItemList** or **LocateItem** call.

Example

```
Dim Region as new PrtRegion
Region.Id = "PRODUCTION"
Region.GetItemList
'Get all items in a region and delete those items with a
' specific ItemTypeId.
For i = 0 to Region.ItemCount - 1
    If Region.Item(i).ItemTypeId = "MARKVII" then
        Region.DeleteItem i, PRT_DELETE, "Obsolete Type"
    End If
Next i
```

PrtRegion.ItemType (Function)

Syntax

PrtRegion.ItemType(index)

PrtRegion.ItemType(itemTypeId\$)

Description

Returns an **PrtItemType** object accessed by a particular index or itemItemId\$.

The **Region.ItemType**(index) object supports the properties **.Id** and **.Count**.

Example

```
Dim Region as new PrtRegion
Region.id = "Schedule"
' Display Id and Number for all Item Types in a region
for i = 0 to Region.ItemTypeCount - 1
    x = Region.ItemType(i).Id & " = " & Region.ItemType(i).Count
    msgbox x
next i
```

PrtRegion.ItemTypeCount (Property Read)

Syntax

PrtRegion.ItemTypeCount

Description

Integer property to get the number of item types available for the region.

Example

```

Dim Region as new PrtRegion
Region.id = "Schedule"
' Display Id and Number for all Item Types in a region
for i = 0 to Region.ItemTypeCount - 1
    x = Region.ItemType(i).Id & " = " & Region.ItemType(i).Count
    msgbox x
next i

```

PrtRegion.Loc (Property Read/Write)

Syntax

PrtRegion.Loc

Description

Integer property to set a specific region location for a subsequent **GetItemList**. The search will then only return a list of items which exist in that single location.

Example

```

Dim Region as new PrtRegion
' Get a single item from a region with its RefId and display
' its ItemId in a message box.
Region.Loc = 0
Region.GetItemList
'Get the Item list from the first location in the region and
' display the results in a message box
If Region.ItemCount = 0 then
    MsgBox "Region " & Region.ID & " has no Items."
Else
    MsgBox "Found " & Region.ItemCount & items."
End If

```

PrtRegion.LocateItem (Method)

Syntax

PrtRegion.LocateItem itemId\$, refId\$ [, numRetries]

Description

Locates a serialized item within the tracking region. Upon return, **PrtRegion.ItemCount** contains the number of items found, and may access each found item through the **PrtRegion.Item** function.

Parameter	Description
itemId\$	String. The Item ID to locate, may be an empty string if refId\$ is specified.

Parameter	Description
refId\$	String. The Reference ID of the item to locate, may omitted or an empty string if itemId\$ is specified.
numRetries	Integer (optional) - The number of times to retry to find the item before failing. If an item is in transition from one tracking region to the next, there is a small period of time when it doesn't appear in either regions tracking queue. Defaults to 3 if not specified.

Example

```
Dim Region as new PrtRegion
Region.Id = "SCHEDULE"
' Get a single item from a region with its RefId and display
' its ItemId in a message box.
Region.LocateItem "", REFNUM$
MsgBox Region.Item(0).ItemId
```

PrtRegion.MaxLocation (Property Read)

Syntax

PrtRegion.MaxLocation

Description

Integer property to get the last occupied region location.

Example

```
Dim Region as new PrtRegion
Region.Id = "PRODUCTION"
Region.GetData 'Get current max. location information
Region.Loc = Region.MaxLocation
'Get all items in the last occupied region location and
' display them one by one in a message box.
Region.GetItemList
for j = 0 to Region.ItemCount - 1
    MsgBox Region.Item(j).ItemId
next j
```

PrtRegion.Modify (Method)

Syntax

PrtRegion.Modify [comment\$] [, userId\$]

Description

Notifies PRT that the modifications are complete and the PRT can update the data.

Parameter	Description
comment\$	String. Optional. A comment to be written to the PRT log file.
userId\$	String. Optional. The User ID to be written to the PRT log file.

Example

```
Dim Prt as new Prt
n% = 17
' Get a list of regions from PRT and initialize a particular
' status bit for each one.
Prt.GetRegionList
for j = 0 to prt.RegionCount - 1
    Prt.Region(j).ClearStatusBit n%
    Prt.Region(j).Modify
next j
```

PrtRegion.ProjectId (Property Read/Write)

Syntax

PrtRegion.ProjectId

Description

String property to set the project that subsequent Production Tracking extensions will communicate with.

Example

```
dim prt as new PrtRegion
' Get a list of services from PRT and display them one by one
' in a message box
prt.ProjectId = "\\project1"
prt.GetItemList
for j = 0 to prt.ItemCount - 1
    MsgBox prt.Item(j).ItemId
next j
```

PrtRegion.Reorder (Method)

Syntax

PrtRegion.Reorder reorderArray [, comment\$ [, userId\$]]

Description

Reorders the locations within a tracking region. This has the effect of moving all the items at each location in the region to another (possibly the same) location in the region.

Parameter	Description
reorderArray	Array, integer. An integer array consisting of region locations. These location numbers (1 to num_locations), specify the new region location for items residing in their current location, that is the locations the items are to be moved to. The reorderArray array index implicitly define the current location number, For example, the first element in the array (reorderArray[0]) refers to the items that currently reside in region location 1. The value of reorderArray[0] specifies the region location where the items currently in region location 1 are to be moved to. This array must include elements for each location currently occupied in the region.
comment\$	String. (Optional) A comment to be recorded into the PRT history log file.
Userld\$	String. (Optional) This is used for logging purposes and is optional. If not supplied the host process name is used.

Example

```
Dim Region as new PrtRegion
Region.id = "OUTPUT"
'Invert the items in a region from first to last
Dim Array(99) as integer
For x% = 0 to 99
    Array(x%) = 100 - x%
Next x%
Region.Reorder Array
```

PrtRegion.SetPoints (Method)

Syntax

PrtRegion.SetPoints enable

Description

Enables or disables the item transition setpoints configured in the PRT tracking route records. These points are set by PRT when an Item transitions into the configured destination region.

enable	Boolean (TRUE or FALSE)
--------	-------------------------

Example

```
Dim Region as new PrtRegion
Region.Id = "PRODUCTION"
If TIME > 18:00:00 or TIME < 6:00:00 then
    Region.SetPoints False
Else
    Region.SetPoints True
```

```
End If
```

PrtRegion.SetStatusBit (Method)

Syntax

```
PrtRegion.SetStatusBit b%
```

Description

Clears a particular status bit in the regions status mask.

Comments

To obtain the region's status mask, use the **PrtRegion.GetData** method. To update the Region's status mask to PRT use the **PrtRegion.Modify**.

Bits 0-15 are for User Use. Bits 16-31 are for GE Intelligent Platforms use. The following constants may be used to access GE Intelligent Platforms specific bits.

Constants	Description
PRT_REGION_FULL	Region Full Bit
PRT_REGION_EXT_HOLD	Region External Hold Bit
PRT_REGION_IN_LOCK	Region in Lock Bit
PRT_REGION_OUT_LOCK	Region Out Lock Bit
PRT_REGION_OUT_OF_SEQ	Region Out of Seq Bit
PRT_REGION_CAP_EXCEEDED	Region Capacity Exceeded Bit
PRT_REGION_NORMAL	Region Normal Bit

Example

```
Dim Prt as new prt
Prt.GetRegionList
'Get a list of regions in PRT, check each one's first status
' bit, then set it if it's not already set.
For J% = 0 to Prt.RegionCount -1
    If Prt.Region(J%).StatusBit(0) = FALSE Then
        CurPrt.Item(I).SetStatusBit 0
        CurPrt.Item(I).Modify
    End If
next J%
```

PrtRegion.Status (Property Read/Write)

Syntax

PrtRegion.Status

Description

Integer property to get or set the region's status mask.

Example

```
Dim Region as new PrtRegion
Region.Id = "PRODUCTION"
Region.GetData
'Perform a quick check for any status bits for a region being
' set and display the results in a message box.
If Region.Status > 0 then
    msgbox "One or more status bits are set."
Else
    msgbox "No status bits set."
End If
```

PrtRegion.StatusBit (Method)

Syntax

PrtRegion.StatusBit(b%)

Description

Accesses a particular status bit in the region's status mask.

Comments

To obtain the region's status mask from PRT, use the **PrtRegion.GetData** method.

Bits 0-15 are for User Use. Bits 16-31 are for GE Intelligent Platforms use. The following constants may be used to access GE Intelligent Platforms specific bits.

Constants	Description
PRT_REGION_FULL	Region Full Bit
PRT_REGION_EXT_HOLD	Region External Hold Bit
PRT_REGION_IN_LOCK	Region in Lock Bit
PRT_REGION_OUT_LOCK	Region Out Lock Bit
PRT_REGION_OUT_OF_SEQ	Region Out of Sequence Bit

Constants	Description
PRT_REGION_CAP_EXCEEDED	Region Capacity Exceeded Bit
PRT_REGION_NORMAL	Region Normal Bit

Example

```
'Example of PrtRegion.StatusBit Function using a constant to
' get the Region Full status bit and display it's status in'
' a message box
Dim DestRegion as new PrtRegion
DestRegion.Id = "Region_Name" 'Replace Region Name
DestRegion.GetData
If DestRegion.StatusBit(PRT_REGION_FULL) Then
    MsgBox "Region Full"
Else
    MsgBox "Region Not Full"
End If
```

PrtRegion.StatusPointId (Property Read)

Syntax

PrtRegion.StatusPointId

Description

String property to get the Point ID configured to contain the status point for the region or an empty string if the status point is not configured for the region.

Example

```
Dim Region as new PrtRegion
Region.Id = "PRODUCTION"
'Get the name of a region's status point and display its
' contents in a message box.
StatPtId$ = Region.StatusPointId
If StatPtId$ <> "" then
    msgbox "Status: " & PointGet(StatPtId$)
Else
    msgbox "No point configured!"
End If
```

PrtRegion.TotalItems (Property Read)

Syntax

PrtRegion.TotalItems

Description

Integer property to get the total number of items in the region.

Example

```
Dim Region as new PrtRegion
Region.Id = "PRODUCTION"
'Get a region's data and display the total number of items it
' contains in a message box.
Region.GetData
msgbox "Region "& Region.id &" contains "& Region.TotalItems &" items."
```

PrtRegion.TotalItemsPointId (Property Read/Write)

Syntax

PrtRegion.TotalItemsPointId

Description

String property to get the Point ID configured to contain the total number of items in the region or an empty string if the ID is not configured.

Example

```
Dim Region as new PrtRegion
Region.Id = "PRODUCTION"
'Get the name of a region's Total Item point and display its
' contents in a message box.
TotPtId$ = Region.TotalItemsPointId
If TotPtId$ <> "" then
    msgbox "Total Items: " & PointGet(TotPtId$)
Else
    msgbox "No point configured!"
End If
```

PrtService.GetGroupList (Method)

Syntax

PrtService.GetGroupList

Description

Gets a list of groups contained in the PRT Service. Upon return, the total group count may be obtained from **GroupCount**, and the individual groups may be accessed via the **Group** function.

Example

```
Dim ServiceA as new PrtService
ServiceA.Id = "SERVICE_A"
' Get a list of groups in the service and display them one by
'   one in a message box.
ServiceA.GetGroupList
for j = 0 to ServiceA.GroupCount - 1
    MsgBox ServiceA.Group(j).Id
next j
```

PrtService.GetItemList (Method)

Syntax

PrtService.GetItemList

Description

Gets a list of all items for the tracking service. Upon rerun, the total item count may be obtained from **ItemCount** and the individual items may be accessed via the **Item** function.

To get the item list for a particular item class, set **PrtService.ItemClassId** to the desired item class id before calling **PrtService.GetItemList**.

Example

```
Dim ServiceA as new PrtService
ServiceA.Id = "SERVICE_A"
' Get a list of items in the service and display them one by
'   one in a message box.
ServiceA.GetItemList
for j = 0 to ServiceA.ItemCount - 1
    MsgBox ServiceA.Item(j).Itemid
next j
```

PrtService.GetRegionList (Method)

Syntax

PrtService.GetRegionList

Description

Gets a list of Tracking regions contained in the PRT Service. Upon return, the total region count may be obtained from **RegionCount**, and the individual regions may be accessed via the **Region** function.

Example

```
Dim ServiceA as new PrtService
ServiceA.Id = "SERVICE_A"
' Get a list of regions in the service and display them one by
' one in a message box.
ServiceA.GetRegionList
for j = 0 to ServiceA.RegionCount - 1
    MsgBox ServiceA.Region(j).Id
next j
```

PrtService.Group (Function)

Syntax

PrtService.Group(index)

Description

Returns a group at a specified index from the list which was found by a previous call to **GetGroupList**.

Example

```
Dim ServiceA as new PrtService
ServiceA.Id = "SERVICE_A"
' Get a list of groups in the service and display them one by
' one in a message box.
ServiceA.GetGroupList
for j = 0 to ServiceA.GroupCount - 1
    MsgBox ServiceA.Group(j).Id
next j
```

PrtService.GroupCount (Property Read)

Syntax

PrtService.GroupCount

Description

Integer property to get the number of Groups found by a prior call to **GetGroupList**.

Example

```
Dim ServiceA as new PrtService
ServiceA.Id = "SERVICE_A"
' Get a list of groups in the service and display them one by
```



```
' one in a message box.
ServiceA.GetGroupList
for j = 0 to ServiceA.GroupCount - 1
    MsgBox ServiceA.Group(j).Id
next j
```

PrtService.Id (Property Read/Write)

Syntax

PrtService.Id

Description

String property to get or set the Service ID associated with the object.

Example

```
Dim ServiceA as new PrtService
ServiceA.Id = "SERVICEA_DC"
' Get a list of groups in the service and display them one by
' one in a message box.
ServiceA.GetGroupList
for j = 0 to ServiceA.GroupCount - 1
    MsgBox ServiceA.Group(j).Id
next j
```

PrtService.Item (Function)

Syntax

PrtService.Item(index)

Description

Returns an Item at the specified index from the list which was found by a previous call to **GetItemList**.

Example

```
Dim ServiceA as new PrtService
ServiceA.Id = "SERVICEA_DC"
' Get a list of items in the service and display them one by
' one in a message box.
ServiceA.GetItemList
for j = 0 to ServiceA.ItemCount - 1
    MsgBox ServiceA.Item(j).ItemId
next j
```

PrtService.ItemClassId (Property Read/Write)

Syntax

PrtService.ItemClassId

Description

String property to set or get the Item Class ID in the PrtService object. Used in conjunction with the PrtService.GetItemList method to retrieve a list of all items of a particular item class in a Tracking Service.

Example

```
Dim ServiceA as new PrtService
ServiceA.Id = "SERVICE_A"
ServiceA.ItemClassId = "CARRIER"
' Get a list of items with an item class id "CARRIER" from the
' service "SERVICE_A" and display them one by one in a
' message box.
ServiceA.GetItemList
for j = 0 to ServiceA.ItemCount - 1
    MsgBox ServiceA.Item(j).Id
next j
```

PrtService.ItemCount (Property Read)

Syntax

PrtService.ItemCount

Description

Integer property to get the number of items find by a previous **GetItemList** or **LocateItem** call.

Example

```
Dim ServiceA as new PrtService
ServiceA.Id = "SERVICEA_DC"
' Get a list of items in the service and display them one by
' one in a message box.
ServiceA.GetItemList
for j = 0 to ServiceA.ItemCount - 1
    MsgBox ServiceA.Item(j).Id
next j
```

PrtService.LocateItem (Method)

Syntax

PrtService.LocateItem itemId, refId [, numRetries]

Description

Locates a serialized item within the tracking service. Upon return, the total item count may be obtained from **ItemCount**, and the individual items may be accessed via the **Item** function.

Parameter	Description
itemId\$	String. The Item ID to locate, may be an empty string if refId\$ is specified.
refId\$	String. The Reference ID of the item to locate, may omitted or an empty string if itemId\$ is specified.
NumRetries	Integer (optional) - The number of times to retry to find the item before failing. If an item is in transition from one tracking region to the next, there is a small period of time when it doesn't appear in either regions tracking queue. Defaults to 3 if not specified.

Example

```
Dim ServiceA as new PrtService
ServiceA.Id = "SERVICEA_DC"
' Get a single item from a region with its RefId and display
' its ItemId in a message box.
ServiceA.LocateItem " ",REFNUM$
MsgBox ServiceA.Item(0).ItemId
```

PrtService.Region (Function)

Syntax

PrtService.Region(index)

Description

Returns a region at the specified index from the list which was found by a previous call to **GetRegionList**.

Example

```
Dim ServiceA as new PrtService
ServiceA.Id = "SERVICEA_DC"
' Get a list of regions in the service and display them one by
' one in a message box.
ServiceA.GetRegionList
```

```
for j = 0 to ServiceA.RegionCount - 1
    MsgBox ServiceA.Region(j).Id
next j
```

PrtService.RegionCount (Property Read)

Syntax

PrtService.RegionCount

Description

Integer property to get a count of the number of Tracking Regions found by a prior **GetRegionList** call.

Example

```
Dim ServiceA as new PrtService
ServiceA.Id = "SERVICEA_DC"
' Get a list of regions in the service and display them one by
' one in a message box.
ServiceA.GetRegionList
for j = 0 to ServiceA.RegionCount - 1
    MsgBox ServiceA.Region(j).Id
next j
```

RCO External Decision

RCO External Decision API Functions

The RCOExtDec Application Programmer Interface (API) is used to access decisions from an external process. There are five API functions associated, each of which is explained in detail in this section.

The RCOExt Dec API:

- Creates the auxiliary port for asynchronous decision request handling.
- Reads the messages on the auxiliary port.
- Stores messages in queue.
- Provides messages to the application using the API.
- Sends the response to the requestor.

 **Note:** The response packed is provided by the application using this API.

The RCOExtDec API functions are:

- ExtDec_api_init
- ExtDec_api_get_msg
- ExtDec_api_init_comm
- ExtDec_api_send_decision
- ExtDec_api_term
- External decision simulator
- Simulator configuration requirements
- RcoExtDecision

ExtDec_api_init

The **ExtDec_api_init** function initializes the RCO External Decision API. It allocates memory for a read / write buffer, creates an auxiliary asynchronous port, and initializes internal data structures. The event flag passed in will be set when asynchronous messages are received from the RCO. This routine must be called after the application calls `ipc_register`, and before calling any other RCOExtDec API routine.

Syntax

COR_14 ExtDec_api_init (COR_U4 Event flag, COR_STATUS * Restat)

Input	
Event flag	Event flag used to indicate when asynchronous messages have arrived from the RCO process.
Output	
Restat	Pointer to status structure.

Return Value:

Either `COR_SUCCESS`, `COR_WARNING` or `COR_FAILURE`. If the function returns `COR_WARNING` or `COR_FAILURE`, additional error information can be found in **restat.err_msg** and **restat.err_code** . If `COR_FAILURE` is returned, the error was severe enough that the application should perform an orderly termination.

ExtDec_api_get_msg

The **ExtDec_api_get_msg** function is responsible for retrieving information from all asynchronous messages coming from an RCO. It is used if the application process is acting as an External Decision-making process; that is, this API function should be invoked by those applications that are expected to supply decisions to an RCO.

This function must be invoked when the event flag reserved for the ExtDec_api_init routine is set High. When this event is set High, the application program calls this function to unload the data contained in the message received from RCO process. This routine should be called repeatedly until it returns a status indicating that there are no more messages to retrieve.

The msg_type argument indicates which type of message has been received. The application determines which type of message was received and responds accordingly. If msg_type field indicates that a decision request was received, the application program should return a decision buffer to RCO using the **RCOExt_Send_decision** routine. If the msg_type indicates that the message is an exception indicator, the application should take the appropriate action. No response is needed to return to RCO. The exception condition is stored in the **exception_flag** field in the ctrl_loc_msg argument.

Syntax:

COR_14 ExtDec__api_get_msg(DADDR*addr,Cor_14*msg_type,PDC_DEC_REQUEST*dec_req, PDC_CTRL_LOC*ctrl_loc_msg, COR_STATUS*restat)

Input	
None	
Output	
Addr	Address of RCO process that sent the message.
msg_type	Pointer to type of message received (Either PDC_DEC_REQ_MSG or PDC_CTRL_LOC_MSG).
dec_req	Pointer to PDC_DEC_CTRL_REQUEST structure.
ctrl_loc_meg	Pointer to PDC_CTRL_LOC_MSG structure.
Restat	Pointer to status structure.

Return Value:

Either COR_SUCCESS, COR_WARNING or COR_FAILURE. If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in **restat.err_msg** and **restat.err_code** . If COR_FAILURE is returned, the error was severe enough that the application should perform an orderly termination. COR_WARNING will be returned when there are no more messages that need to be retrieved by the application program. PDCI_LAST_MESSAGE will be reported in **restat.err_code** .

ExtDec_api_init_comm

The **ExtDec_api_init_comm** function sends an INIT_SEG message to calling process (whose DADDR * is passed in), thereby allowing the calling process to recognize the async port address. This port will then handle the subsequent decision requests from the calling process. This routine must be called after the application calls and INIT_SEG is received from RCO.

Syntax:

COR_14 ExtDec_api_init_comm (DADDR*addr,Cor_STATUS*restat)

Input	
Addr	Address of RCO process that send INIT_SEG.
Output	
Restat	Pointer to status structure.

Return Value:

Either COR_SUCCESS, COR_WARNING or COR_FAILURE. If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in restat.err_msg and restat.err_code.

ExtDec_api_send_decision

The ExtDec_api_send_decision function sends a decision buffer to the RCO. If the external process is unable to return a decision, this function should be called after setting the **dec_quantity** field to zero in the PDC_CTRL_DECISIONS structure.

Syntax:

COR_14 ExtDec_api_send_decision

(DADDR*addr,PDC_CTRL_DECISIONS*ctrl_loc_dec,COR_STATUS*)

Input	
addr	Address of RCO process that will receive decisions.
Ctrl_loc_dec	Pointer to structure containing a list of control decisions.
Output	
Restat	Pointer to status structure.

Return Value:

Either COR_SUCCESS, COR_WARNING or COR_FAILURE. If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in restat.err_msg and restat.err_code.

ExtDec_api_term

The ExtDec_api_term function is responsible for reallocating message buffers and configuration data. All applications that call ExtDec_api_init must also call this routine prior to process termination.

Syntax

COR_14 ExtDec_api_term (COR_STATUS*)

Input	
None	
Output	
Restat	Pointer to status structure.

Return Value

Either COR_SUCCESS, COR_WARNING or COR_FAILURE. If the function returns COR_WARNING or COR_FAILURE, additional error information can be found in restat.err_msg and restat.err_code. If COR_FAILURE is returned, the error was severe enough that the application should perform an orderly termination.

RCOExtDec API Calls in relation to PDC_API Calls

RCOEXTDEC API Calls	Corresponding PDC_API Calls
COR_14 ExtDec_api_init (COR_U4, COR_STATUS)	COR_14 pdc_api_init (COR_U4, COR_STATUS)
COR_14 ExtDec_api_get_msg (DADDR*, COR_14, PDC_DEC_REQUEST*, PDC_CTRL_LOC*, COR_STATUS) Additional argument DADDR* - address of requesting process (read from message)	COR_14 pdc_api_get_msg (COR_14*, PDC_DEC_REQUEST*, PDC_CTRL_LOC*, COR_STATUS)
COR_14 ExtDec_api_init_comm (DADDR*, COR_STATUS*)	Used by pdc_api_init as init_comm (internal function —not exported), to initialize communication with Controller processes configured to talk to external process using this API.
COR_14 ExtDec_send_decision (DADDR*, COR_14, PDC_CTRL_DECISIONS*, COR_STATUS) Additional argument DADDR* - address of requesting process.	COR_14 pdc_api_send_decision (PDC_CTRL_DECISIONS*, COR_STATUS*)
COR_14 ExtDec_api_term (COR_STATUS)	COR_14 pdc_api_term (COR_STATUS)

External Decision Simulator

External Decision Simulator

The External Decision Simulator is used to test RCOExtDec API process. External decisions are simulated and provided to the appropriate structures. The Simulator is structured like PDC_SIM (Test driver) and uses three basic sets of functions as follows:

Set 1 (page 759)	Framework Functions (ExtDec_Sim.C).
Set 2 (page 759)	Data Management and Event Handler Functions (ExtDec_SimData.cpp).
Set 3 (page 761)	RCOExtDec API Calls (RCOExtDec.Dll/.Lib)

Set 1. Framework Functions (ExtDec_Sim

1. RCO ExtDec_api_init initializes the API's data structures.
2. Async ports are created to handle decision request messages.
3. Once initialized, the main program calls the event handler loop.
4. Event handler loop continuously checks for any message (read timeout = one second) on both main port and async port (through event flag).
5. The appropriate handlers are called (set two functions) through function pointers for these messages.
6. Once the main port receives TERM_SEG or Router dies, the main program calls **CleanUp** function to free the allocated memory.
7. Process terminates.

Set 2. Data Management and Event Handler Functions (ExtDec_SimData.cpp)

Set 2. Data Management and Event Handler Functions (ExtDec_SimData.cpp)

The data management and event handler functions being with a prefix, DecEml_, and provide support for simulating the external decisions. Simulated decisions are read from file **Dec_sim.rco**, which resides in sc_path directory, and stored in the appropriate structures. Whenever a request is made for one or n decisions, they are provided to caller from these structures. The event handler functions read the message from async port (through API calls), retrieves the data from decision structures, and sends the response to call (through API calls.)

- Data management functions.
- Event handler functions.
- ExtDec_Sim Calls in relation to PDC_Sim/API_Caller Calls

Data Management Functions

1. int **DecEml_init_data** (COR_STATUS*) Reads the decision and its control location information from Dec_Sim.Rco file, and stores it in the data structure (decision table).

Note: The **Dec_Sim.Rco** file has a .cfg file format, and all file operations are performed using the IDTPOP utility. A sample format is provided in ExtDec_SimData.cpp file.

2. int **DecEml_get_next_decision** (PDC_DEC_REQUEST*, PDC_CTRL_DECISIONS*)

Retrieves the number of decisions from the decision table based on request information, and provides them in PDC_CTRL_DECISIONS structure.

3. int **DecEml_handle_info_msg** (PDC_CTRL_LOC*) Retrieves the controller information from internal structures.

4. void **DecEml_cleanup** ()

Removes decision table and other structures from memory.

5. void **DecEml_reset_ctrl_dec** ()

Resets the current pointer in the decision table, which renders it the first decision to be provided.

Event Handler Functions

Event Handler Functions

1. int **DecEml_api_init** (COR_STATUS*,int) Initializes data and RCOExtDec API. Calls DecEml_init_data for application data initialization and RCOExtDec_api_init API function for API initialization.
2. int **DecEml_api_main_init** (DADDR*,COR_STATUS*) Initiates communication with RCO. When RCO sends an INIT_SEG to main port, this function calls RCOExtDec_api_init_comm API to send an INIT_SEG to RCO through async port; thereby enabling RCO to recognize async port and to send all subsequent decisions to async port.
3. int **DecEml_api_process** (COR_STATUS*) When the event flag indicates a message's arrival at async port, this function is called; which in turn, calls DecEml_handle_decision_requests function.
4. int **DecEml_handle_decision_requests** (COR_STATUS*) Forces ExtDec_api_get_msg to retrieve messages from async port. If it is a decision request, it calls DecEml_get_next_decision to obtain the number of decisions from the decision table, and sends these decisions to the requestor through ExtDec_api_send_decision API call.
5. int **DecEml_api_cleanup** (COR_STATUS*) Calls DecEml_cleanup and ExtDec_api_term API call.

ExtDec Sim Calls in relation to PDC Sim/API Caller Calls

Calls table:

EXTDEC_SIM Calls	PDC_SIM Calls
<code>int DecEml_init_data (COR_STATUS*)</code>	<code>int g2_init (COR_STATUS*)</code>
<code>int DecEml_get_next_decision (PDC_DEC_REQUEST*, PDC_CTRL_DECISIONS*)</code>	<code>int g2_get_next_decision (PDC_DEC_REQUEST*, PDC_CTRL_DECISIONS*)</code>
<code>int DecEml_handle_info_msg (PDC_CTRL_LOC*)</code>	<code>int g2_handle_info_msg (PDC_CTRL_LOC*)</code>
<code>void DecEml_cleanup()</code>	<code>void g2_cleanup()</code>
<code>void DecEml_reset_ctrl_dec()</code>	<code>void g2_reset_ctrl_dec (ctrl_loc_id_ptr)</code>
<code>int DecEml_api_init (COR_STATUS*,int)</code>	<code>int g2_api_init (COR_STATUS*, int)</code>
<code>int DecEml_api_main_init (DADDR*, COR_STATUS*,int)</code>	<code>int g2_api_main_init (COR_STATUS*, DADDR*, int event_flag)</code>
<code>int DecEml_api_process (COR_STATUS*,int)</code>	<code>int g2_api_process (COR_STATUS*)</code>
<code>int DecEml_api_handle_decision_requests (COR_STATUS*)</code>	<code>int handle_decision_requests (COR_STATUS*)</code>
<code>int DecEml_api_cleanup (COR_STATUS*)</code>	<code>int g2_api_cleanup (COR_STATUS*)</code>

Set 3. RCO External Decision API Calls

The RCO External Decision API calls are described in detail in the ["RCO External Decision API Functions" \(page 754\)](#) section.

Process Flowchart

The following flowchart depicts the process between the Decision Simulator interface and RCOExtDec API:

Simulator Configuration Requirements

Simulator Configuration Requirements

Four of the five files needed to configure the External Decision Simulator can be found in the Master directory of CIMPLICITY. The files are in binary format and must be converted to ASCII text using the IDTPOP Utility. The **master.mcp** can be edited directly. Also, files that have overlapping fields (for example, **process_id**) must have identical values in order to run the simulator.

The required configuration files are:

- **physproc.dat**
- **logproc.dat**
- **node_logproc.dat**
- **service.dat**
- **master.mcp**

The following steps provide instruction on configuring External Decision Simulator:

Step 1 (page 762)	Open an RCO project in Workbench
Step 2 (page 763)	Configure physproc
Step 3 (page 763)	Configure logproc
Step 4 (page 764)	Configure node_logproc
Step 5 (page 764)	Configure service
Step 6 (page 765)	Configure master.mcp
Step 7 (page 765)	Copy Decision configuration file


Step 1. Open an RCO Project in Workbench

1. From the Windows desktop, press Start.
2. Select (All) Programs>Proficy HMI SCADA - CIMPLICITY version>.

The Workbench window opens.

3. In the Workbench window, click File on the menu bar.
4. Select **Open** option to display the Open dialog box.
5. Navigate to your RCO project and click on it.

- Click **Open** to open your project in the Workbench window.

 **Tip:** To open a project quickly, create a shortcut for the project on your Windows desktop and double-click the icon.

Step 2. Configure physproc

- Use the IDTPOP Utility to convert the file to ASCII-text using the text editor of your choice, e.g. Notepad.
- Press **Ctrl+End** to bring you to the next available line to enter your record.
- Type your record using the following field descriptions:

Field No.	Field Name	Description	Entry Example
0	node_id	Node name, always MASTER.	MASTER
1	process_id	The name of the logical process that the program will look to for data.	EXTSIM1
2	object_name	Object name, same as Process ID.	EXTSIM1
3	order	Use default value = 0.	0

- Save the file.
- Use the SCPOP Utility to convert file back to binary format.

Step 3. Configure logproc

- Use the IDTPOP Utility to convert the file to ASCII-text using the text editor of your choice, e.g. Notepad.
- Press **Ctrl+End** to bring you to the next available line to enter your record—see figure.
- Type your record using the following field descriptions:

Field No.	Field Name	Description	Entry Example
0	process_id	The name of the process that the program will look to for data.	EXTSIM1
1	process_type_id	Process type ID; usually RESIDENT process.	RESIDENT
2	pm_flags	Use value = 0.	0

Field No.	Field Name	Description	Entry Example
3	description	Description of the process	Ext Dec Test

4. Save the file.
5. Use the SCPOP Utility to convert file back to binary format.

Step 4. Configure node_logproc

1. Use the IDTPOP Utility to convert the file to ASCII-text using the text editor of your choice, e.g. Notepad.
2. Press **Ctrl+End** to bring you to the next available line to enter your record.
3. For this record you are only concerned with the entries in the first three fields. Use the default values for the remaining fields, which can be found on the sample configuration.

Field No.	Field Name	Description	Entry Example
0	node_id	Node name, always MASTER.	MASTER
1	process_id	The name of the process that the program will look to for data.	EXTSIM1
2	image_name	The name of the executable image. Configure as follows: Directory_name:[exe]prt_inttest.exe	BSM_ROOT:[EXE] ExtDec_Sim.exe

4. Save the file.
5. Use the SCPOP Utility to convert file back to binary format.

Step 5. Configure service

1. Use the IDTPOP Utility to convert the file to ASCII-text using the text editor of your choice, e.g. Notepad.
2. Press **Ctrl+End** to bring you to the next available line to enter your record.
3. Type your record using the following field descriptions:

Field No.	Field Name	Description	Entry Example
0	service_id	The name given to this service.	EXTSIM1

Field No.	Field Name	Description	Entry Example
1	subsys_id	The ID of the subsystem, in this case RCO.	RCO
2	process_id	The name of the logical process that the program will look to for data.	EXTSIM1

4. Use the SCPOP Utility to convert file back to binary format.

Step 6. Configure master.mcp

1. Open the Windows Explorer.
2. Navigate to the **master.mcp** file in the directory. Typical file location = **\Program Files \Proficy\Proficy CIMPLICITY\bsm_root .**
3. Double-click on the file and open with the text editor of your choice, e.g. Notepad.
4. Press **Ctrl+End** to go to the last line in the file.
5. Enter the name of the logical process, e.g. **EXTSIM1**.
6. Save the file.

Step 7. Copy Decision Configuration File

1. Open the Windows Explorer.
2. Navigate to the ExtDec_sim folder.

The default file location is:

`\Program Files\Proficy\Proficy CIMPLICITY\Api\ExtDec_sim.`

3. Copy the **Dec_sim.Rco** file.
4. Select the Data folder in your project directory.
5. Paste Dec_sim.Rco into the Data folder.

RcoExtDecision

RcoExtDecision

RcoDecisionBuffer contains the array of RcoExtDecision objects

Structure

```

{
Dec_Name //Text
Dec_Attr //Text
Req_flag //Integer
}

```

Methods

This object has following methods to get and set the properties:

Get Methods

RcoExtDecision. Dec_Name = Text

RcoExtDecision. Dec_Attr = Text

RcoExtDecision. Req_flag = Integer

Set Methods

Text = RcoExtDecision. Dec_Name

Text = RcoExtDecision. Dec_Attr

Integer = RcoExtDecision. Req_flag

RcoDecisionBuffer

Structure of RcoDecisionBuffer basic object

```

{
    Id //Control location id of external
    decisions.
    TrigTime //Trigger time
    Dec_Qty //Number of decisions in buffer
    Error //Error message returned, if error
    occurred during any

```

//operation on RcoDecisionBuffer fails.

```

}

```


Methods

Following methods are available to assign values in object members:

`RcoDecisionBuffer.Id= Text //Specifies Control location id`

Following methods provide functionality to retrieve values from `RcoDecisionBuffer` Object:

Text = `RcoDecisionBuffer. Id` //Specifies Control location id

Date = `RcoDecisionBuffer .TrigTime` //Specifies trigger time

Integer = `RcoDecisionBuffer. Dec_Qty` //Specifies number of decisions in buffer

RcoExtDecision = `RcoDecisionBuffer .DecArray(I)` //Returns RcoExtDecision at index I

Text = `RcoDecisionBuffer. Error` //Error message returned by Rco if any

//operation on RcoDecisionBuffer fails.

Functions

- Append
- Delete
- DeleteAt
- Flush

- GetDecisions
- Insert
- RequestDecisions
- SelectNextDecision

Append

Appends a `RcoExtDecision` basic object in Decision buffer.


Syntax

```
RcoDecisionBuffer.id = text //id of control location
Integer = RcoDecisionBuffer .Append(RcoExtDecision)
```

Return

```
RCO_SUCCESS // = 1
RCO_WARNING // = 0
```

```
RCO_FAILURE // = 2
```


 **Note:** If return value is RCO_FAILURE or RCO_WARNING, the Error member of RcoDecisionBuffer will show cause of the error.

Delete

Deletes a decision from the buffer by decision name. If two or more decisions with the same name exist, then the first duplicate decision is deleted.

Syntax

```
RcoDecisionBuffer.id = text //id of control location
Integer = RcoDecisionBuffer .Delete( Decision name /Text/)
Return
RCO_SUCCESS // = 1
RCO_WARNING // = 0
RCO_FAILURE // = 2
```

 **Note:** If return value is RCO_FAILURE or RCO_WARNING, the Error member of RcoDecisionBuffer will show cause of the error.

DeleteAt


Deletes a decision from buffer at a particular index.

Syntax

```
RcoDecisionBuffer.id = text //id of control location
Integer = RcoDecisionBuffer .DeleteAt( Index /Integer/)
```

Return

```
RCO_SUCCESS // = 1
RCO_WARNING // = 0
RCO_FAILURE // = 2
```

 **Note:** If return value is RCO_FAILURE or RCO_WARNING, the Error member of RcoDecisionBuffer will show cause of the error.

Flush


Deletes all the decisions from decision buffer.

Syntax

```
RcoDecisionBuffer.id = text //id of control location
Integer = RcoDecisionBuffer . Flush ( )
```

Return

```
RCO_SUCCESS      // = 1
RCO_WARNING      // = 0
RCO_FAILURE      // = 2
```

 **Note:** If return value is RCO_FAILURE or RCO_WARNING, the Error member of RcoDecisionBuffer will show cause of the error.

GetDecisions


Gets the list of external decisions from RCO and stores them in order to populate itself with the data.

Syntax

```
RcoDecisionBuffer.id = text //id of control location
Integer = RcoDecisionBuffer . GetDecisions ()
```

Return

```
RCO_SUCCESS      // = 1
RCO_WARNING      // = 0
RCO_FAILURE      // = 2
```

 **Note:** If return value is RCO_FAILURE or RCO_WARNING, the Error member of RcoDecisionBuffer will show cause of the error.

Insert


Inserts an RcoExtDecision basic object at a particular index; if the index is more than size of buffer object, it is inserted at the tail. If the index is less than one, it is inserted at the first location.

Syntax

```
RcoDecisionBuffer.id = text //id of control location
Integer = RcoDecisionBuffer .Insert(RcoExtDecision, Index /Integer/)
```

Return

```
RCO_SUCCESS      // = 1
RCO_WARNING      // = 0
RCO_FAILURE      // = 2
```


 **Note:** If return value is RCO_FAILURE or RCO_WARNING, the Error member of RcoDecisionBuffer will show cause of the error.

RequestDecisions

Request a particular number of decisions from external process.


Syntax

```
RcoDecisionBuffer.id = text //id of control location
Integer = RcoDecisionBuffer . RequestDecisions (project name/external
process name /Text/, Decision
attribute /Text/, Number of decisions /Integer/)
```

 **Note:** Project name and process name should be passed as single string separated by "/"; however, if the process belongs to the local project, the process name is sufficient.

Return

```
RCO_SUCCESS // = 1
RCO_WARNING // = 0
RCO_FAILURE // = 2
```

 **Note:** If return value is RCO_FAILURE or RCO_WARNING, the Error member of RcoDecisionBuffer will show cause of the error.

SelectNextDecision

Selects a (first) decision from the decision buffer and executes when configured in RCOSite). If the decision is not configured in RCOSite, then it is not get executed and an error returns. Moreover, calling this function will also remove the first decision from buffer, so next call will execute next decision. If there are no decisions left in buffer then it returns an error stating there are no decisions left. A RequestDecisions call is required to be called once decision buffer empties, to execute more decisions.


Syntax

RcoDecisionBuffer.id = text //id of control location

Integer = RcoDecisionBuffer . SelectNextDecision ()

Return

```
RCO_SUCCESS // = 1
RCO_WARNING // = 0
RCO_FAILURE // = 2
```

 **Note:** If return value is RCO_FAILURE or RCO_WARNING, the Error member of RcoDecisionBuffer will show cause of the error.

External Decision Buffer

RCO has the ability to handle external decisions provided by external processes through function blocks. This includes requesting and receiving external decisions from external processes, storing these decisions in a buffer, allowing manipulation like append, insert, flush, delete, and execution of these decisions in the decision buffer.

Two new basic objects handle decisions through function blocks as follows:

Object	Holds
RcoDecisionBuffer	A list of decisions received for a particular controller location.
RcoExtDecision	The decision data of a decision received from external process.

For additional information about external decision function blocks, click:

- [Append a Decision in ExtDecBuffer](#)
- [Delete a Decision in ExtDecBuffer](#)
- [Delete at a Decision in ExtDecBuffer](#)
- [Flush the ExtDecBuffer](#)
- [Get Ext Decision List](#)
- [Insert a Decision in ExtDecBuffer](#)
- [Request External Decisions](#)
- [Select Next External Decision](#)

Chapter 5. Order Execution Management

Order Execution Management

About Order Execution Management

Order Execution Management provides a comprehensive addition to Tracker that enables you to track, store, categorize and sequence your customers' orders based on your configured criteria.

The following diagram shows an overview Order Execution Management schema

rect -5, 9, 349, 152 [About External Scheduling \(page 773\)](#)

rect -5, 172, 184, 324 [About XMLT \(page 773\)](#)

rect -5, 322, 185, 388 [About DIR_WATCHER.bcl \(page 796\)](#)

rect -5, 386, 185, 498 [About CimView Order Entry \(page 810\)](#)

rect 286, 156, 413, 258 [About the Tracker Attribute Database \(page 817\)](#)

rect 479, 146, 579, 236 [About the Tracker Query Engine \(page 893\)](#)


rect 206, 275, 581, 486 [About the Product Order Management System \(POMS\) \(page 799\)](#)

rect 224, 490, 356, 551 [About Order Execution Management Broadcast \(page 934\)](#)

rect 580, 395, 638, 464 [About Range Source Architecture \(page 859\)](#)

Item	Description
1 (page 773)	Raw order data is entered into your system's computer.
2 (page 773)	XMLT translation files are triggered by your custom application to convert the text files into an XML output format.
3 (page 796)	Dir_Watcher.bcl starts up at each 10 minutes on the clock, e.g. 1:10PM, 1:20PM, and gathers XML output files.
4 (page 799)	Gathered XML files are the source to be evaluated and go through the Product Order Management System.
5 (page 810)	Pre-configured CimView screens provide maintenance and monitoring capability.
6 (page 817)	The Tracker Attribute Database (TADB) stores both order and production the data.

Item	Description
7 <i>(page 859)</i>	Range Source Architecture (RSA) dynamically sets the range source for routing logic (RLM) or output logic module
8 <i>(page 893)</i>	The Query Engine retrieves production and/or order data in response to a query expression.
9 <i>(page 934)</i>	Orders can be Broadcast at any configured time during the order management process.
10 <i>(page 126)</i>	Accepted orders go to production in an organized sequence.

 **Important:** POMS only processes files that have an .xml extension.

1. External Scheduling

About External Scheduling

External scheduling is performed at your site.

Raw data that is based on your organization and business decisions must be in .xml format for Order Execution Management to process.

You can convert the data using your own conversion methods or take advantage of the XMLT utility supplied by Order Execution Management.

rect -5, 123, 184, 304 [About XMLT \(page 773\)](#)

rect -2, 318, 188, 384 [About DIR_WATCHER.bcl \(page 796\)](#)

rect 206, 275, 581, 486 [About the Product Order Management System \(POMS\) \(page 799\)](#)

rect 7, 393, 197, 505 [About CimView Order Entry \(page 810\)](#)

rect 286, 156, 413, 258 [About the Tracker Attribute Database \(page 817\)](#)

rect 580, 395, 638, 464 [About Range Source Architecture \(page 859\)](#)

rect 421, 144, 521, 234 [About the Tracker Query Engine \(page 893\)](#)

rect 224, 490, 356, 551 [About Order Execution Management Broadcast \(page 934\)](#)

2. XMLT

About XMLT

Order Execution Management provides tools to take raw data orders, translate them into an .xml format and enter valid data into PRT and TADB

rect 83, 7, 345, 138 [About External Scheduling \(page 773\)](#)
 rect 1, 316, 201, 407 [About DIR_WATCHER.bcl \(page 796\)](#)
 rect 212, 271, 567, 482 [About the Product Order Management System \(POMS\) \(page 799\)](#)
 rect 4, 410, 219, 485 [About CimView Order Entry \(page 810\)](#)
 rect 281, 154, 413, 257 [About the Tracker Attribute Database \(page 817\)](#)
 rect 574, 395, 638, 464 [About Range Source Architecture \(page 859\)](#)
 rect 420, 145, 520, 235 [About the Tracker Query Engine \(page 893\)](#)
 rect 224, 490, 356, 551 [About Order Execution Management Broadcast \(page 934\)](#)
 rect 224, 490, 356, 551 [About Order Execution Management Broadcast \(page 934\)](#)
 rect -6, 140, 190, 295 [XMLT Overview \(page 774\)](#)

[XMLT Translation \(page 780\)](#) begins this process.

XMLT Overview

The XMLT system includes the following.

rect 49, 9, 335, 132 [1. Sales Orders in External Scheduling \(page 774\)](#)
 rect 3, 129, 232, 307 [2. XMLT Translation Files \(page 775\)](#)
 rect -5, 311, 196, 399 [3. Dir_Watcher.bcl \(page 776\)](#)
 rect 213, 331, 365, 395 [4. POMS.xsd Schema \(page 776\)](#)

1 (page 774)	Sales orders (External scheduling)
2 (page 775)	XMLT Translation Files
3 (page 776)	Dir_Watcher.bcl
4 (page 776)	POMS.xsd Schema

1. Sales Orders in External Scheduling

Sales orders are taken and entered into the retail (dealer) computer.

Data in the sales order must be in text format in order to be processed by the XMLT translation utility.

Your ERP system or other external (to CIMPLICITY) scheduling system is responsible for:


- Creating the sales orders text files

Sales order entries in a text file format are FTP'd to the XMLT server.

- Moving files to a configured RAW_DUMP folder on the XMLT server
- Calling the XMLT translation utility.


Note: More information on calling XMLT is available in the [CIMPLICITY XML Translator help \(page 775\)](#). Also, an [XMLT User Interface \(page 795\)](#) is provided for testing your translation files.

- Creating custom XML translation files configured specifically for your text-based files (sales orders, parts lists, etc.)

 **Note:** XMLT supports a wide variety of text file formats by virtue of its flexible XML schema definition, specified by the file XMLT.xsd in the POMS\XMLT folder under **BSM_Root**. This schema defines custom XML tags (Entity Types) that XMLT uses to restructure data parsed from your text-based order files into a format suitable as input to POMS.

2. XMLT Translation Files

XMLT uses .xml translation files to translate the text files into an XML output specified by an XML schema file, POMS.xsd. These files are moved to the configured XMLT_DUMP folder.

 **Note:** Examples of XML translation files are provided in the ...\\Program Files\\Proficy\\Proficy CIMPLICITY\\POMS\\XMLT\\TranslationFiles folder.

An .xml translation file contains custom Element Type tags that are defined by the XML schema file, XMLT.xsd. These custom elements (tags) tell the XMLT translation utility how to parse information from a particular input file format.

The actual structure that a POMS XML order must adhere to is defined by a different schema file, POMS.xsd.

For example, you can create different XML translation files that:

- Build XML **order** files.

You can configure XMLT through your translation file to generate individual output files for each order or multiple-order files with a configurable target size.

- Build XML **group only** files.

For example, part data can be used to create a parts lookup table in TADB.

3. Dir_Watcher.bcl

The [Dir_Watcher.bcl \(page 797\)](#) script is automatically provided with any CIMPLICITY project you create using the POMS Wizard.

Dir_Watcher.bcl:


- Executes regularly on a timed basis defined in the CIMPLICITY Event Manager.

Default startup is each 10 minutes on the clock, e.g. 1:10PM, 1:20PM, and moves XML output files to the SOURCE folder.

- Copies XMLT output files (XML order files, etc.) from the specified folder (XMLT_DUMP) in chronological (XML file creation) order to the POMS input directory (SOURCE).

This ensures that all files have:

- Completed their downloading from XMLT
- Been given unique filenames.
- Removes the original files from the XMLT_DUMP directory and triggers the POMS input RCO via point update when all available files have been copied over.

 **Note:** The paths defining the XMLT_DUMP and SOURCE folders are stored in CIMPLICITY points. If desired, these points can be dynamically configured using scripts.


4. POMS.xsd Schema

Gathered .xml files are the source that is evaluated and goes through the Product Order Management System (POMS) once the trigger for the first RCO site, \$OMV_ORDER, is initiated.

An XML schema file, POMS.xsd, defines the order entry interface to POMS. This schema is used for validating XMLT output before it is accepted for processing by POMS.

POMS.xsd is the required schema definition for the .xml order files. This file is located in the ... \Program Files\Proficy\Proficy CIMPLICITY\POMS directory and defines the various elements and attributes constitute an XML-based order that the Plant Order Management System (POMS) accepts.

XML-based orders are the translated XML files that XMLT generates as output.

 **Note:** You do not code with POMS elements. XMLT automatically creates translated files that incorporate these elements. The description below is provided for debugging purposes only. For information on XMLT and XMLT Element Types see [XMLT Schema \(page 783\)](#).

```
<?xml version="1.0"?>
```

```

<Schema name="POMS" xmlns="urn:schemas-microsoft-com:xml-data"
          xmlns:dt="urn:schemas-microsoft-com:datatypes" >
  <AttributeType name="AN" dt:type="string"/>
  <AttributeType name="GN" dt:type="string"/>
  <AttributeType name="IT" dt:type="string"/>
  <AttributeType name="Id" dt:type="string"/>
  <ElementType name="A" dt:type="string" model="closed">
    <attribute type="AN" required="yes"/>
  </ElementType>
  <ElementType name="R" content="eltOnly" model="closed">
    <element type="A" maxOccurs="*" />
  </ElementType>
  <ElementType name="G" content="eltOnly" model="closed">
    <attribute type="GN" required="yes"/>
    <element type="R" maxOccurs="*" />
  </ElementType>
  <ElementType name="I" content="eltOnly" model="closed">
    <attribute type="IT" required="yes"/>
    <attribute type="Id" required="yes"/>
    <element type="G" maxOccurs="*" />
  </ElementType>
  <ElementType name="File" content="eltOnly" order="one" model="open">
    <element type="I" minOccurs="0" maxOccurs="*" />
    <element type="G" minOccurs="0" maxOccurs="*" />
  </ElementType>
</Schema>

```

POMS Element Types

These elements appear in the XMLT output file, i.e. the file that POMS takes as its input.

Element type	Description		
File	File header for the POMS XML order file (XMLT output file). Note: This file may contain one order or multiple orders depending on the configuration of the File Element Type in the translation XML file based on XMLT.xsd.		
	Content:	eltOnly	
	Order:	One	
	Model:	Open	
	Attribute Types:		
	Element Types:	I (page 778)	Optional
		G (page 778)	Optional
	Related Elements :		
	Example		

	An File element with a Date attribute: <File Date="030930">. . . </File>	
I	Item element for an order item. Note: The Item Type (IT) must be defined in the TADB and PRT databases before POMS can process the order.	
	Content:	mixed Note: Text content, i.e. Item Type, must be all uppercase.
	Order:	
	Model:	Closed
	Attribute Types:	IT (page 780)
		Id (page 780)
	Element Types:	G (page 778) Optional
	Related Elements:	File
	Example Item:.. <I IT="ORDER" Id="ABC12345X67890"> <G GN="Order"> <R> ABC12345X67890AAP2AFWAUVB0ABVCA2H83CFBE1RCWIDGET 000120001 </R> </G> </I>	
G	Group element. <ul style="list-style-type: none"> • If nested under the I element, the group is associated with an order for that item. • If not nested under an item, the group represents data for a lookup table in the TADB database. 	
	Content:	eltOnly
	Order:	
	Model:	Closed
	Attribute Types:	
	Element Types:	R (page 778)
	Related Elements:	File (page 777)
	Example A nested group associated with an order item: <I IT="ORDER" Id="ABC12345X67890"> <G GN="Order Card"> <R> ABC12345X67890AAP2AFWAUVB0ABVCA2H83CFBE1RCWIDGET 000120001 </R> </G> </I> Group data for TADB parts lookup tables: <File Date="20030311"> <G GN="Installed Parts"> <R> Warehouse Part for Assy 101 </R> </G> <G GN="Peripheral Parts"> <R> Outsourced Part for Assy 101 </R> </G> </F>	
R	Record element. This element contains one or more A elements (TADB group attributes).	
	Content:	eltOnly
	Order:	
	Model:	Closed

	Attribute Types:		
	Elements Types:	A (page 779)	
	Related Elements:	File (page 777)	
	Example A group associated with an order item: <I IT="ORDER" Id="ABC12345X67890"> <G GN="Order Card"> <R> ABC12345X67890AAP2AFWAUVB0ABVCA2H83CFBE1RCWIDGET 000120001 </R> </G> </I>		
A	Attribute element (TADB group attribute).		
	Content:	Text Only	
	dt:type:	String	
	Order:		
	Model:	Closed	
	Attribute Types:	AN (page 779)	
	Element Types:		
	Related Elements:	R (page 778)	
	Examples An TADB attribute: <R> Outsourced Part for Assy 101 </R>		

POMS Attribute Types

These attributes appear in POMS elements within the XMLT output file, i.e. the file that POMS takes as its input.

Attribute Type		
AN	Attribute name.	
	dt:type	String
	Length	Up to 40 characters.
	Default	
	Example	
	Outsourced Part for Assy 101	
GN	Group name.	
	dt:type	String
	Length	
	Default	
	Example	

	<G GN="Order Card">	
IT	Item Type (Item Type ID).	
	dt:type	String
	Length	
	Default	
	Example	
	<I IT="ORDER" Id="ABC12345X67890">	
Id	Item ID.	
	dt:type	String
	Length	
	Default	.
	Example	
	<I IT="ORDER" Id="ABC12345X67890">	

XMLT Translation Utility

XMLT Translation Utility

The XMLT conversion utility converts order data from the legacy formats to a POMS XML compliant format. This translation utility is designed to convert raw formats where the order records can be easily located and the order attributes can be mapped.

You configure the XMLT translation utility by creating an XML document, an .xml translation file, that defines how the utility parses data from the source input files.

The XML translation file you create must be based on the XML schema definition file XMLT.xsd located in the Program Files\Proficy\Proficy CIMPLICITY\POMS\XMLT directory.

Sample translation files that you can customize to convert data from your company's orders are located in ...\Program Files\Proficy\Proficy CIMPLICITY\POMS\XMLT\TranslationFiles.

- Create an XML Translation File
- XMLT Schema
- XMLT Element Types
- XMLT Attribute Types
- XMLT User Interface

Create an XML Translation File


This section describes the creation of the XML translation file. This file defines a data transduction between the text-based input files created by an external order scheduling system and the XML-based output files XMLT generates.

- XMLT Element Types are a set of custom tag elements defined in the XML schema definition file, XMLT.xsd.
- You use these custom elements to construct an XML translation file in your text editor.
- The translation file tells the XMLT translation utility how to parse data from a text-based input file in order to create the translated XML output file.
- A different schema file, POMS.xsd, defines the structure of XML output files. This schema file is also used for validating output files before accepting them for processing by POMS.

The XML translation file as a set of instructions that tell XMLT where to find the data it needs. You should be thoroughly familiar with:

- The structure and semantics of your input files before starting.
- XSD schemas, terminology and use.

It is suggested that the Order items and associated TADB Groups and attributes be defined in PRT before creating XML translation files. This guide is intended provide you with the necessary information about how XMLT structures your data to create output files acceptable to POMS.


 **Note:** You may need to create one or more translation files depending on the data you want to bring into POMS. By default, POMS accepts:

- Orders (Items with attribute groups)
- Parts lists (attribute groups only).

XML Translation File Structure

The structure of the translation file is specified by the schema definition file [XMLT.xsd \(page 783\)](#), located in the ...\\Program Files\\Proficy\\Proficy CIMPLICITY\\POMS\\XMLT directory.

The translation file consists of required and optional elements. A typical outline of the basic structure of a translation file for translating orders is shown below. There are many variations of this structure, see the XML files provided in located in ...\\Program Files\\Proficy\\Proficy CIMPLICITY\\POMS\\XMLTTranslationFiles directory for examples.

 **Note:** The attributes of these elements have been omitted for clarity. In some cases only one element, or related set of elements, is shown where several may appear in practice.

Purpose	XML Translation Element
	<?xml version="1.0" ?>

Purpose	XML Translation Element
Output file creation options.	<File>
(Optional) Output file header.	< Header (page 785) >
	< Attribute (page 790) >
	Content
	< Chunk (page 790) />
	</Attribute>
	</Header>
A new record (Item, Group Only, or Item and Group data)	< Record (page 787) >
(Optional) Where the record starts.	< StartRecord (page 787) >
	Content
	< Chunk (page 790) />
	</StartRecord>
(Optional) Where the record ends.	< EndRecord (page 788) >
	< Chunk (page 790) />
	</EndRecord>
(Optional) Item, e.g. order, information.	< ItemType (page 788) >
	Content
	< Chunk (page 790) />
	</ItemType>
(Optional) Group information.	< GroupName (page 789) >
	Content
	</GroupName>
(Required if GroupName appears) Group attributes.	< ItemData (page 789) >
TADB group attribute	< Attribute (page 790) >
	Content
	< Chunk (page 790) />
	</Attribute>
	</ItemData>
	</Record>
(Optional) Line count validation	< Trailer (page 786) >
	< CountField (page 786) >

Purpose	XML Translation Element
	<Chunk (page 790) />
	</Trailer>

XMLT Schema

An XML schema file, XMLT.xsd, defines the structure of XML translation files configure XMLT to translate text-based input files into XML-based output files. This schema includes Element Types for each type of record that can occur in the XMLT output file (the XML item order and/or group data suitable that POMS accepts as input). Its Attribute Types tell the XMLT translator where to parse information from the customer source file (the file to be translated) and how to format it for the various elements so that the translated file can be successfully validated by POMS.

```
<?xml version="1.0"?>
<Schema name="xmlt" xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt="urn:schemas-microsoft-com:datatypes" >
  <AttributeType name="Line" dt:type="ui2" default="0"/>
  <AttributeType name="Offset" dt:type="ui2"/>
  <AttributeType name="Length" dt:type="ui2"/>
  <AttributeType name="Pad" dt:type="enumeration" dt:values="True False"
    default="False"/>
  <AttributeType name="Pattern" dt:type="string"/>
  <AttributeType name="CountExtra" dt:type="ui2"/>
  <AttributeType name="CollectLines" dt:type="enumeration" dt:values="True
    False"/>
  <AttributeType name="RecordLength" dt:type="ui2"/>
  <AttributeType name="EachLine" dt:type="enumeration" dt:values="True False"
    default="False"/>
  <AttributeType name="DateFormat" dt:type="enumeration" dt:values="YYYYMMDD
    YYMMDD YYYYDDMM YYDDMM YYYYDDD YYDDD MMDDYY MMDDYYYY DDMYY DDMYYYY"/>
  <AttributeType name="SplitFile" dt:type="enumeration" dt:values="True
    False" default="True"/>
  <AttributeType name="SplitType" dt:type="enumeration" dt:values="Item Size"
    default="Size"/>
  <AttributeType name="SplitSize" dt:type="ui2" default="20000"/>
  <ElementType name="Chunk" content="empty" order="many">
    <attribute type="Line"/>
    <attribute type="Offset" required="yes"/>
    <attribute type="Length" required="yes"/>
    <attribute type="Pad"/>
  </ElementType>
  <ElementType name="Attribute" order="many" model="closed">
    <attribute type="DateFormat"/>
    <element type="Chunk"/>
  </ElementType>
  <ElementType name="CountField" content="eltOnly" order="many"
    model="closed">
    <attribute type="CountExtra"/>
  </ElementType>
```


```

    <element type="Chunk" />
</ElementType>
<ElementType name="Header" content="eltOnly" order="many" model="closed">
    <attribute type="Pattern" required="yes" />
    <element type="Attribute" minOccurs="0" />
</ElementType>
<ElementType name="Trailer" content="eltOnly" model="closed">
    <attribute type="Pattern" required="yes" />
    <element type="CountField" />
</ElementType>
<ElementType name="StartRecord" model="closed">
    <element type="Chunk" />
</ElementType>
<ElementType name="EndRecord" model="closed">
    <element type="Chunk" />
</ElementType>
<ElementType name="ItemType" model="closed">
    <element type="Chunk" />
</ElementType>
<ElementType name="GroupName" content="textOnly" dt:type="string"
    model="closed" />
<ElementType name="ItemData" content="eltOnly" order="many" model="closed">
    <attribute type="EachLine" />
    <element type="Attribute" />
</ElementType>
<ElementType name="Record" content="eltOnly" order="many" model="closed">
    <attribute type="CollectLines" />
    <attribute type="RecordLength" />
    <element type="StartRecord" minOccurs="0" maxOccurs="1" />
    <element type="EndRecord" minOccurs="0" maxOccurs="1" />
    <element type="ItemType" minOccurs="0" maxOccurs="1" />
    <element type="GroupName" maxOccurs="1" />
    <element type="ItemData" maxOccurs="*" />
</ElementType>
<ElementType name="File" content="eltOnly" order="many" model="closed">
    <attribute type="SplitFile" />
    <attribute type="SplitType" />
    <attribute type="SplitSize" />
    <element type="Header" minOccurs="0" />
    <element type="Record" />
    <element type="Trailer" minOccurs="0" />
</ElementType>
</Schema>

```

XMLT Element Types

These elements appear in [XMLT translation files \(page 775\)](#), and specify the data to extract from an XMLT input files for various [XMLT data structures \(page 783\)](#). XMLT uses these structures to build the XMLT output file.

 **Note:** Unless otherwise stated, the examples below would occur in a translation XML file based on XMLT.xsd. Examples from other types of files, e.g. input and output files, are indicated where used.

Element Type	Description		
File	<p>Each input file can contain multiple order/group records. The attributes of this element configure XMLT for various types of output. Output files can be split if:</p> <ul style="list-style-type: none"> • The resulting file will be greater than a configured size, or • The resulting file will contain more than one order (item). 		
	Content	eltOnly	
	Order	Many	
	Model	Closed	
	Attribute Types	SplitFile (page 794)	
		SplitType (page 795)	
		SplitSize (page 795)	
	Element Types	Header (page 785)	Optional
		Record (page 787)	Required
		Trailer (page 786)	Optional
	Related Elements		
	<p>Example An File element configures XMLT to split output files in order to obtain individual XML output files for each order item: <File xmlns="x-schema:..xmlt.xsd" SplitFile="True" SplitType="Item"> . . . </File></p>		
Header	(Optional) Specifies header information. XMLT uses this information to generate attributes for the <File> header in the XML output file.		
	Content	EltOnly	
	Order	Many	
	Model	Closed	
	Attribute Types	Pattern (page 794)	Required
	Element Types	Attribute (page 790)	Optional

	Related Elements	File (page 785)	
	<p>Example An Attribute element within a Header element extracts six characters from the default (0th) line of the header (the line starting with the letters "HEADR"), starting at offset 6 (the 7th character): <code><Header Pattern="HEADER"> <Attribute DateFormat="YYMMDD">Date <Chunk Offset="6" Length="6"></Chunk> </Attribute> </Header></code> The extracted text becomes a Date attribute in the Output XML file header, in YYMMDD format : <code><File Date="030310"> . . .</code></p>		
Trailer	(Optional) Specifies trailer information and validates the input file.		
	Content	eltOnly	
	Order		
	Model	Closed	
	Attribute Types	Pattern (page 794)	Required
	Element Types	CountField (page 786)	
	Related Elements	File (page 785)	
	<p>Example An CountField element within a Trailer element obtains seven characters from the default (0th) line of the trailer (the line starting with the letters "TRAILR"), starting at offset 34 (the 35th character): <code><Trailer Pattern="TRAILR"> <CountField CountExtra="2"> <Chunk Offset="34" Length="7"></Chunk> </CountField> </Trailer></code> The value obtained is adjusted by adding the CountExtra value, 2. XMLT compares the adjusted value with its internal count of the number of lines actually processed.</p>		
CountField	Specifies the number of lines in the input file, not including the header or trailer, for validation purposes. The number of lines actually processed by XMLT has to match this number (adjusted by the optional CountExtra attribute) or the translation fails.		
	Content	EltOnly	
	Order	Many	
	Model	Closed	
	Attribute Types	CountExtra (page 791)	Optional
	Element Types	Chunk (page 790)	
	Related Elements	Trailer (page 786)	
	<p>Example An CountField element within a Trailer element obtains six characters from the 0th line of the trailer (the line starting with the letters "TRAILR"), starting at offset 18 (the 19th character): <code><Trailer Pattern="TRAILR"> <CountField> <Chunk Offset="18" Length="6" Line="0" Pad="False"/> </CountField> </Trailer></code> XMLT compares the value obtained with its internal count of the number of lines actually processed.</p>		

Record	Specifies processing options for records. <ul style="list-style-type: none"> • Each line can be processed as an individual record, or data from multiple input lines can be collected as a single record,. • The number of characters to be collected for each record can be specified in cases where no end-of-line/line feed character is used in the input file. 		
	Content	eltOnly	
	Order	Many	
	Model	Closed	
	Attribute Types	CollectLines (page 791)	Optional
		RecordLength (page 794)	Optional
	Elements Types	StartRecord (page 787)	
		EndRecord (page 788)	
		ItemType (page 788)	Optional
		GroupName (page 789)	Optional
		ItemData (page 789)	
	Related Elements	File (page 785)	
	<p>Example An Record element with no attributes (defaults to CollectLines=False) treats each line of input as a separate record. Note that the Chunk elements below have no Line attributes; the Item ID, at offset 10, and all attributes at offsets 243 and 249, are extracted from the same single input line. <Record> <ItemType>ORDER <Chunk Offset="10" Length="17"></Chunk> </ItemType> <GroupName>Feature Codes</GroupName> <ItemData> <Attribute>Features <Chunk Offset="243" Length="5"></Chunk> </Attribute> </ItemData> <ItemData> <Attribute>Feature Code <Chunk Offset="249" Length="5"></Chunk> </Attribute> </ItemData> </Record></p>		
StartRecord	(Optional) Specifies the characters that mark the start of a record in the input file. XMLT finds the record start marker by searching the input file for a string of characters, specified by the text content of this element, at a character offset specified by its associated Chunk . Note: The record start defines line 0 for elements contained within the previous Record element.		
	Content	mixed	
	Order		
	Model	Closed	
	Attribute Types		

	Element Types	Chunk (page 790)	
	Related Types	Record (page 787)	
	<p>Example A StartRecord element defines that characters that mark the start of a new record.</p> <ul style="list-style-type: none"> • The text content, e.g. UH, marks the record start. • Chunk defines the offset and length of this marker. <pre><Record CollectLines="True"> <StartRecord>UH <Chunk Offset="8" Length="2"></ Chunk> </StartRecord> Lines, in the input file, where with the characters "UH" occur at the offset 8 (9th character) start a new record: 00000001UH01 GBB43D0930801000001 00000002UD01 5310 CNDSBAAATEST FUNCTION 000002.00000 00000003UD01 5310 CNDSBJ9XEC00040101904 000002.00000 00000006UT01 GBB43D09308010000004 00000007UH01 GBBW3D1511301000002 00000008UD01 5310 CNDSBAAATEST FUNCTION 000002.00000 00000009UD01 5310 CNDSBGJVEC00040101904 000002.00000 00000012UT01 GBBW3D15113010000004</pre>		
EndRecord	<p>(Optional) Specifies the characters that mark the end of a record in the input file. XMLT finds the record end by searching the input file (downward from the current record start marker) for a string of characters specified by the text content of this element, at a character offset specified by its associated Chunk.</p>		
	Content	mixed	
	Order		
	Model	Closed	
	Attribute Types		
	Element Types	Chunk (page 790)	
	Related Elements	Record (page 787)	
	<p>Example An EndRecord element defines that characters that mark the end of a the current record.</p> <ul style="list-style-type: none"> • The text content, e.g. UT, marks the end of the record. • Chunk defines the offset and length of this marker. <pre><EndRecord>UT <Chunk Offset="8" Length="2"></Chunk> </EndRecord> An example of the input file is shown above for StartRecord (page 787) .</pre>		
ItemType	<p>(Optional) Identifies a record as an item, e.g. as an order belonging to the ORDER Item Type, specifies the unique identifier for this specific instance of that Item Type. The text content of this record identifies the name of the Item Type (or Item Type ID) for the item. Note: The Item Type you specify must be defined in the TADB and PRT databases.</p>		
	Content	mixed	Note: Text content, i.e. Item Type, must be all uppercase.
	Order		
	Model	Closed	
	Attribute Types		

	Element Types	Chunk (page 790)	
	Related Elements	Record (page 787)	
	<p>Example An ItemType element identifies the following record as an item of the ORDER Item Type. The Chunk on the following line specifies a unique identifier (ABC12345X67890) for the current item at offset 9 (10th character) of line 1 of the current record. <code><Record CollectLines="True"> <StartRecord>1 <Chunk Offset="0" Length="1"></Chunk> </StartRecord> <ItemType>ORDER <Chunk Line="1" Offset="9" Length="14"></Chunk> </ItemType> XML Output: <IT="ORDER" Id="ABC12345X67890"> . . . </I></code></p>		
GroupName	<p>(Optional) Associates one or more Attribute elements as a group data portion of the current record. The text content of this record identifies the name of the group. Note: The group name must be defined in the TADB database. Group data can follow an ItemType element or can follow a Record (or StartRecord) element directly.</p> <ul style="list-style-type: none"> • If following the ItemType (page 788) Element Type, the GroupName is associated with an order for that item type. • If the GroupName does not follow an ItemType, then the group represents data for a lookup table in the TADB database. 		
	Content	Text Only	
	dt:type	String	
	Order		
	Model	Closed	
	Attribute Types		
	Element Types		
	Related Elements	Record (page 787)	
	<p>Example An GroupName element identifies the following ItemData element as group data belonging to the TADB group Order Card. <code><ItemType>ORDER <Chunk Line="1" Offset="9" Length="14"></Chunk> </ItemType> <GroupName>Order Card</GroupName> <ItemData EachLine="False"> <Attribute>pVIN <Chunk Line="4" Offset="28" Length="4" Pad="False" /> </Attribute> </ItemData></code></p>		
ItemData	Specifies TADB attributes for a particular TADB group.		
	Content	eltOnly	
	Order	Many	
	Model	Closed	
	Attribute Types	EachLine (page 792)	
	Child Elements	Attribute (page 790)	

	<p>Example An ItemData element specifies that each line of the current record represents a complete group record. Note that a Chunk belonging to this element does not need to have a Line attribute; the line is specified by a previous StartRecord (page 787) element. <code><GroupName>Parts</GroupName> <ItemData EachLine="True"> <Attribute>Base <Chunk Offset="12" Length="9"></Chunk> </Attribute> <Attribute>Prefix <Chunk Offset="21" Length="7"></Chunk> </Attribute> </ItemData></code></p>		
Attribute	<p>An Attribute element. This element specifies name and value for an "attribute." Note: This element is used to generate both XML tag attributes and TADB group attributes, depending on the context in which you use it.</p>		
	Content	mixed	<p>Within a Header element, the content consists of:</p> <ul style="list-style-type: none"> • text - the name of an attribute for the output <code><File></code> tag, and • Chunk (one or more) - the value for this attribute. <p>Within an ItemData element, the content consists of:</p> <ul style="list-style-type: none"> • text - the value of the output <code><A></code> tag's AN attribute, and • Chunk (one or more) - the content for the output <code><A></code> tag. <p>Note: The <code><A></code> tag occurs in the XML output file generated by XMLT. It defines a TADB group element, not an attribute of an XMLT element. Important: The output file has its own set of tag elements, defined by the schema file POMS.xsd. You never code with POMS tags. They are discussed here only to provide a better understanding of how XMLT elements process your data.</p>
	Order	Many	
	Model	Closed	
	Attribute Types	DateFormat (page 792)	
	Element Types	Chunk (page 790)	
	Related Elements	Header (page 785) , ItemData (page 789)	
	<p>Examples An Attribute element within a Header element extracts six characters from the 0th line of the input header (the line starting with the letters "HEADR"), starting at offset 5 (the sixth character): <code><Header Pattern="HEADR"> <Attribute DateFormat="YYMMDD">Date <Chunk Offset="5" Length="6" Line="0" Pad="False" /> </Attribute> </Header></code> These characters become a value for the Date attribute of the <code><File></code> tag (the file header) in the XML output file: <code><File Date="030930"></code> Note that the output date is formatted as Year-Month-Day.</p>		
	<p>An Attribute element within an ItemData element extracts 14 characters from lines nine and ten of the current record (a record start marker defines line 0), starting at offset 10 (the eleventh character) in one case, and the first character in the other: <code><ItemData EachLine="False"> <Attribute>PIN <Chunk Line="9" Offset="10" Length="5" Pad="False" /> <Chunk Line="10" Offset="0" Length="9" Pad="False" /> </Attribute> </ItemData></code> These characters become the content of an <code><A></code> tag in XML output file. The text content of the Attribute element, "PIN" becomes a value for the AN attribute of the <code><A></code> tag in the XML output file: <code>ABC12345X67890</code></p>		
Chunk	<p>Specifies a string of contiguous characters obtained from a single line of the input text file.</p>		
	Content	Empty	
	Order	Many	

	Model	Closed	
	Attribute Types	Line (page 793)	Optional, defaults to 0.
		Offset (page 793)	Required
		Length (page 793)	Required
		Pad (page 794)	Optional, defaults to False
	Element Types		
	Related Elements	StartRecord (page 787) , EndRecord (page 788) , ItemType (page 788) , Attribute (page 790) , CountField (page 786)	
	Example A Chunk specifies a string of eight characters on input line 1, starting with the twenty-fourth character from the left. <code><Chunk Line="1" Offset="24" Length="8" /></code> Note: Line numbers start at 0. Numbering starts at the line specified by the previous Header (page 785) , StartRecord (page 787) , or Trailer (page 786) element.		

XMLT Attribute Types

XMLT Attribute Types are attributes of XMLT Element Types. These attributes specify how their parent Element Type process data from the input file.

Attribute Type	Description
CollectLines	<p>When True, allows multiple lines of the input file to be processed as a single record. All lines after the record start marker are considered as belonging to the same record until XMLT encounters :</p> <ul style="list-style-type: none"> • The record end marker, • The another record start marker, or • The last line of the input file. <p>When False, each line after the file header is processed as a complete record. In this case no record start marker is required and the default Line value (0) for ItemType Chunks can be omitted.</p>
	dt:type Enumeration
	dtvalues True, False
	Default False
	Attribute of Record (page 787)
	<p>Example <code><Record collectLines="True"> <StartRecord>IT_NO_06796C <Chunk Offset="10" Length="6"></Chunk> </StartRecord></code> This example states that the input text file has data for a particular item data on multiple lines. Note: The first line of any item data record always starts with the record start marker, i.e. the characters "IT_NO_" at character offset 10.</p>
CountExtra	(Optional) Specifies the value to add to the value, extracted by CountField before XMLT verifies the number of records in the input file.

Attribute Type	Description	
	dt:type	ui2
	Range	0 to 65,535
	Default	
	Attribute of	CountField (page 786)
	Example <Trailer Pattern="TRAILR"> <CountField CountExtra="2"> <Chunk Offset="34" Length="7"></Chunk> </CountField> </Trailer> This example above states that the number of lines XMLT processed should be incremented by a value of 2 before validating the CountField line count.	
DateFormat	(Optional) Specifies a date format for an Attribute Element Type. Use DateFormat to specify the date format that the characters in your input file represent.	
	dt:type	Enumeration
	dtvalues	YYYYMMDD, YYMMDD, YYYYDDMM, YYDDMM, YYYYDDD, YYDDD, MMDDYY, MMDDYYYY, DDMMYY, DDMMYYYY
	Default	No formatting applied
	Attribute of	Attribute (page 790)
	Example <Header Pattern="HEADER"> <Attribute DateFormat="YYMMDD">Date <Chunk Line="0" Offset="6" Length="6" Pad="False" /> </Attribute> </Header> This example specifies that the input characters extracted for the header are a date represented in a 2-digit Year/Month/Day format. The extracted date will appear in a standardized date format in the output file.	
EachLine	<p>Specifies whether each line of input represents a complete group record from which data can be extracted for the various TADB group attributes.</p> <ul style="list-style-type: none"> • True, processes each input line as an individual group. <p>Note: The Line attribute can be omitted from Chunk elements within the respective ItemData element. Data is obtained from the default line (line 0) specified by the previous RecordStart.</p> <ul style="list-style-type: none"> • False, allows multiple input lines to be processed as a group. • Usually you will set EachLine to True if the attributes extracted from the input file are represented as a TADB List group in POMS. If the attributes are represented in POMS as a Normal TADB group, set EachLine to False. 	
	dt:type	Enumeration
	dtvalues	True, False
	Default	False
	Attribute of	ItemData (page 789)

Attribute Type	Description
	<p>Example <code><GroupName>Parts</GroupName> <ItemData EachLine="False"></code> <code><Attribute>Prefix <Chunk Offset="21" Length="7"></Chunk> </Attribute></code> <code><Attribute>Suffix <Chunk Offset="28" Length="8"></Chunk> </Attribute> </</code> <code>ItemData></code> This example states that group item data records (attributes of a particular group) occur on multiple lines. Note that each Chunk has a Line attribute specifying the line number static to the record start marker.</p>
Length	The number of characters in a Chunk , beginning with the indicated starting point (Offset). Note: A Length of 0 defines an empty string.
	dt:type ui2
	Range 0 to 65,535
	Default
	Attribute of Chunk (page 790)
	<p>Example <code><ItemType>ORDER <Chunk Line="4" Offset="28" Length="4"></Chunk> <Chunk Line="0" Offset="26" Length="3"></Chunk> <Chunk Line="4" Offset="32" Length="2"></Chunk></code> <code></ItemType></code> Note: The first Chunk in this example says: Go to line 4 in the current record; start at character 28 (the 29th character); pull four characters.</p>
Line	Specifies the line in the input file where data for a Chunk can be found. The line count starts at first line of the respective header, footer, or record. Note: The first line is line 0.
	dt:type ui2
	Range 0 to 65,535
	Default 0
	Attribute of Chunk (page 790)
	<p>Example <code><Record CollectLines="True"> <StartRecord>1 <Chunk Line="0" Offset="0" Length="1" Pad="False" /> </StartRecord> <ItemType>ORDER <Chunk Line="4" Offset="28" Length="4" Pad="False" /></code> Note: The last Chunk in the example above says: Go to line 4 (the fifth line) in the current record in the input text file; start at character 28th; pull four characters.</p>
Offset	The number of characters from the left (on the specified Line of the input file) that identifies a starting point for a Chunk . Note: The first character on a line is character 0.
	dt:type ui2
	Range 0 to 65,535
	Default
	Attribute of Chunk (page 790)
	<p>Example <code><ItemType>ORDER <Chunk Line="4" Offset="28" Length="4"></Chunk></code> <code><Chunk Line="0" Offset="26" Length="3"></Chunk> <Chunk Line="4" Offset="32"</code> <code>Length="2"></Chunk> </ItemType></code> Note: The first Chunk in this example says: Go to line 4 in the current record; start at character 28 (the 29th character); pull four characters.</p>

Attribute Type	Description	
Pad	Determines if a Chunk should be padded with extra spaces (if the number of characters found at the specified Offset is less than the specified Length).	
	dt:type	Enumeration
	dtvalues	True, False
	Default	False
	Attribute of	Chunk (page 790)
	<p>Example <code><Attribute>Description <Chunk Offset="15" Length="36" Pad="True"></Chunk> </Attribute></code> Note: This example says: Go to the default line (0) in the current header or record; start at character 15; pull 36 characters; and if less than 36 characters are found there, pad the result with leading zeros to obtain 36 characters.</p>	
Pattern	(Required) Specifies a string of characters appearing in the input file that XMLT uses to identify the start of a Header or Trailer . Note: the pattern string must start in the first column (0th character) of the input file.	
	dt:type	String
	Length	
	Default	
	Attribute of	Header (page 785) , Trailer (page 786)
	<p>Example An input text file has a file header line: <code>HEADER030930</code> The translation XML defines: <code><Header Pattern="HEADR"> <Attribute>Date <Chunk Offset="7" Length="6"></Chunk> </Attribute> </Header></code> This example above states that any line starting with the characters "HEADR" is the first line in a header.</p>	
RecordLength	(Optional) Specifies a length (number of characters) for a Record when the records are not separated by a CR/NL character. Note: Do not use this attribute when <code>CollectLines="True"</code>	
	dt:type	ui2
	Range	0 to 65,535
	Default	See <code>CollectLines</code> .
	Attribute of	Record (page 787)
	<p>Example <code><Record RecordLength="80"> . . . </Record></code> This example states that a new record occurs every eighty characters, starting with the first record start marker.</p>	
SplitFile	Specifies whether input files should be split up into separate output files when translated.	
	dt:type	Enumeration
	dtvalues	True, False
	Default	True

Attribute Type	Description	
	Attribute of	File (page 785)
	Example This example creates a single output file containing all orders items (or group records) parsed from the input file: <File xmlns="x-schema:..\xslt.xsd" SplitFile="False">	
SplitType	<p>If files are split (SplitFile="True") this attribute determines how output files are split up:</p> <ul style="list-style-type: none"> • Item - creates a separate output file for each ordered item. • Size - creates a separate item when the output file will exceed the SplitSize after processing of the current record is complete. <p>Note: The size of the output file may exceed SplitSize slightly</p>	
	dt:type	Enumeration
	dtvalues	Item, Size
	Default	Size
	Attribute of	File (page 785)
	Example This example creates separate output files for each orders parsed from the input file: <File xmlns="x-schema:..\xslt.xsd" SplitFile="True" SplitType="Item">	
SplitSize	If files are split (SplitFile="True") this attribute determines how output files are split up.	
	dt:type	ui2
	Range	0 to 65,535
	Default	
	Attribute of	File (page 785)
	Example This example starts writing to another output file when the last record processed causes the current output file to exceed the 20 MB: <File xmlns="x-schema:..\xslt.xsd" SplitFile="True" SplitType="Size" SplitSize="20000">	

XMLT User Interface

1. Click Start on the Windows task bar.
2. Select Run. to display the Run dialog box.
3. Enter XMLT.exe in the **Open** field.
4. Click OK.

An XMLT dialog box opens.

The fields are as follows.

Option	Description
Input file	Contains raw data for your sales order(s).
Output file	Will contain the translated orders in xml format.
Translation file	File that was created for translating the input file.
Browse	Opens a Browser to locate the correct file.


5. Click Convert to XML to translate the input file.

The XMLT Translation Utility creates a translated file that represents the order data in XML format.

3. *DIR_WATCHER.bcl*

About DIR_WATCHER.bcl

The Directory Watcher, DIR_WATCHER.bcl, is a script file that monitors the XMLT output directory (configured by the point XML.XMLT_DUMP) on a timed-basis for the appearance of XML order files. After confirming that order files have completed downloading to this directory, the Directory Watcher moves files to the POMS input directory (configured by the point XML.SOURCE).

 **Note:** A check is made to insure that no files already in the POMS input directory are overwritten.

Once order files are moved, the Directory Watcher triggers the first POMS RCO, \$OMV_ORDER, to validate and process the orders.

rect 88, 0, 363, 135 [About External Scheduling \(page 773\)](#)

rect -7, 135, 207, 295 [About XMLT \(page 773\)](#)

rect 215, 276, 566, 485 [About the Product Order Management System \(POMS\) \(page 799\)](#)

rect 1, 408, 209, 486 [About CimView Order Entry \(page 810\)](#)

rect 296, 161, 414, 260 [About the Tracker Attribute Database \(page 817\)](#)

rect 570, 393, 633, 468 [About Range Source Architecture \(page 859\)](#)

rect 421, 147, 521, 237 [About the Tracker Query Engine \(page 893\)](#)

rect 224, 490, 356, 551 [About Order Execution Management Broadcast \(page 934\)](#)

rect -6, 293, 212, 406 [1. DIR_WATCHER.bcl Overview \(page 797\)](#)

1 (page 797)	DIR_WATCHER.bcl overview.
2 (page 798)	Open the DIR_WATCHER.bcl.

1. DIR_WATCHER.bcl Overview

DIR_WATCHER.bcl manages incoming .xml source files for the POMS system and triggers POMS processing of those files.

1. XMLT:

- a. Converts legacy data into an .xml format file with a pre-defined schema.
- b. Places the .xml file in a remote directory.

2. DIR_WATCHER.bcl looks for a file specified by a point, XML.XMLT_DUMP, located under the site root directory, and copies it into another directory where the POMS order validation routine can process.

The directory where this file is copied by the DIR_WATCHER.bcl is determined by the value of another point, XML.SOURCE.


Another point, XML.ROOT, configures the path name of the root directory where the Directory Watcher looks for both the source and destination directories.

The Directory Watcher script is automatically executed on a timed-basis by the CIMPLICITY Event Manager associated with your POMS-based project.

3. By default, every 10 minutes, the DIR_WATCHER.bcl:

- Copies a file from the XML.DUMP directory to the directory specified by XML.SOURCE and
- Triggers the \$OMV_ORDER site in POMS for processing.

You can change the default for how often DIR_WATCHER.bcl runs, using the CIMPLICITY Event Editor.

 **Note:** The name of the file that \$OMV_ORDER would process is dynamically set in another point, XML.PROCESSING_FILE, by the DIR_WATCHER.bcl script.

\$OMV_ORDER validates files placed in the XML.SOURCE directory using the schema file specified in the point XML.SCHEMA_NAME.

Invalid	\$OMV_ORDER rejects the file and moves it to Rejected directory, located in the directory site_root \POMS\TBNG_POMS\OrderFiles directory.
Valid	POMS processing of accepted files then begins.

4. The CIMPLICITY Event Manager executes DIR_WATCHER.bcl on a timed basis. The file paths used are stored in CIMPLICITY points.

DIR_WATCHER.bcl:

- Copies files from the specified source directory in chronological order to the POMS input directory.
- Ensures that all files have both completed downloading and have unique filenames.

When all available files have been copied over, the Directory Watcher:


- Removes copied files from the source directory and
- Triggers the POMS input to RCO via a point (OMV_ORDER_T) update.

2. Open the DIR_WATCHER.bcl

Note: TBNG_POMS is a sample project included with CIMPLICITY Plant Edition. This project is available as a template. Do not edit it directly.

1. Open the TBNG_POMS project or a project that was created from the TBNG_POMS project.
2. Expand the Basic Control Engine folder in the Workbench left pane..
3. Select Scripts.
4. Double-click DIR_WATCHER.bcl in the right pane.

DIR_WATCHER.bcl opens in the CIMPLICITY Program Editor.

 **Note:** The script file is located in...\\Program Files\\Proficy\\Proficy CIMPLICITY\\Projects\\TBNG_POMS\\Scripts\\DIR_Watcher.bcl.

3. Customize DIR_WATCHER.bcl

Do any of the following to configure the script that feeds POMS, DIR_WATCHER.bcl.

Set the:

- Name of the POMS schema file by configuring the name set in the point XML.SCHEMA_NAME. This file is located by default in the XML.Root directory under the site root.
- Directory where XMLT puts its XML output files by configuring the point XML.XMLT_DUMP. This is the same directory DIR_WATCHER.bcl watches for new files it move to the XML.SOURCE directory location.
- Directory location where POMS picks up orders (XML output files created by XMLT) by configuring a point named XML.SOURCE.

- Path name of the root directory where DIR_WATCHER.bcl looks for the above directories by configuring the point XML.Root.

Every 10 minutes, the DIR_WATCHER.bcl:

- Copies a file from the XML.XMLT_DUMP directory to the directory specified by XML.SOURCE and
- Triggers the \$OMV_ORDER site in POMS for processing.

You can configure how often DIR_WATCHER.bcl runs using the CIMPPLICITY Event Editor.

 **Note:** The name of the file that \$OMV_ORDER would process is dynamically set in another point, XML.PROCESSING_FILE, by the DIR_WATCHER.bcl script.


\$OMV_ORDER validates files placed in the XML.SOURCE directory using the schema file specified in the point XML.SCHEMA_NAME.


Invalid	\$OMV_ORDER rejects the file and moves it to Rejected directory, located in the directory site_root\POMS\TBNG_POMS\OrderFiles directory.
Valid	POMS processing of accepted files then begins.

4. Product Order Management System

About the Product Order Management System (POMS)

The Product Order Management System (POMS) can be the hub of your Order Execution Management order management system. POMS is essentially a project that contains the basic configuration on which you can build your customized system. Order Execution Management includes a template project, TBNG_POMS, and provides a Wizard with which you can create your own POMS project.

 **Warning:** Do not open the TBNG_POMS project. This is the template the Wizard uses to create your project.

 **Important:** POMS requires:

- Order Execution Management.
- Query Engine setup.
- TADB setup.
- TADB interface setup.

POMS only processes files that have an .xml extension.

POMS Setup Wizard

About the POMS Setup Wizard

The POMS Wizard

- Creates a Order Execution Management project based on a TBNG_POMS template.
- Sets up the connections to a specified SQL database.

Step 1 (page 800)	Open the Plant Order Management Setup wizard.
Step 2 (page 801)	Select the POMS setup destination.
Step 3 (page 801)	Select the RCO database connection parameters.
Step 4 (page 802)	Select the TADB database connection parameters.
Step 5 (page 803)	Open the new POMS project.
Step 6 (page 804)	Create a SEQATTRIB Query expression.

! **Warning:** TBNG_POMS is a template for projects that you create. It is configured precisely for that purpose. Do not open TBNG_POMS or you may find some incorrect configuration in the new project.

Step 1. Open the Plant Order Management Setup Wizard

! **Important:** If you have meta-data .xml files, that will be used for your TADB database, copy them to the Wizard directory before you open the Plant Order Management Setup Wizard. A default meta-data file, POMS_TADB.xml, is located in that directory.

Method 1

1. Click Start on the Windows task bar.
2. (All) Programs>Proficy HMI SCADA - CIMPLICITY version>Order Management Tracking Application Template Wizard.

Method 2

Do the following.

A	Expand ...\\Proficy CIMPLICITY.
B	Expand POMS.
C	Select Wizard.
D	Double-click pomswiz.exe.

The Plant Order Management (POMS) Setup Wizard splash screen opens.

Step 2. Select the POMS Setup Destination

1. Click Next on the Plant Order Management Setup Wizard splash screen.

The POMS Wizard destination dialog box opens.

2. Do one of the following.

Method 1

- a. Click the Browse button to the right of the **Destination and Filename** field.

A Save As dialog box opens.

- a. Create a folder for your project.
- b. Enter a project name in the **File name** field.
- a. Click Save.

The path to the folder and the file name you just entered display in the Destination and Filename field.

Method 2

Enter the path to an existing folder and the name of the project you want the Wizard to create In the **Destination and Filename** field.

The POMS wizard will set up the specified project in the selected location.

Step 3. Select the RCO Database Connection Parameters

1. Click Next in the POMS Wizard Destination dialog box.

The POMS Wizard RCO Database Connection Parameters dialog box opens.

2. Provide the following data, as follows.

rect 31, 68, 322, 90 ([page 802](#))
 rect 71, 164, 202, 186 ([page 802](#))
 rect 10, 140, 295, 162 ([page 802](#))
 rect 10, 114, 295, 136 ([page 802](#))
 rect 10, 90, 401, 112 ([page 802](#))

Specification	Description		
Server	The node on which the SQL database is located. Note: If you are using CIMPLICITY SQL, the server will have \CIMPLICITY in its identification.		
	Opens Data Link Properties dialog box, in which you can find or set up the server if it is not already set up. Note: Microsoft help is available for your configuration.		
Database	The SQL database that will contain the Tracker RCO data. Important: The RCO database requires SQL Server 2000 and service pack 2 for SQL Server 2000.		
	<table border="1"> <tr> <td>Create Database</td> <td> If an RCO database does not exist: <ol style="list-style-type: none"> Enter a name in the Database field. Click Create Database. The POMS Wizard will create a database with the name you entered. A message will tell you if the operation was successful. Note: You cannot select an existing RCO database. </td> </tr> </table>	Create Database	If an RCO database does not exist: <ol style="list-style-type: none"> Enter a name in the Database field. Click Create Database. The POMS Wizard will create a database with the name you entered. A message will tell you if the operation was successful. Note: You cannot select an existing RCO database.
Create Database	If an RCO database does not exist: <ol style="list-style-type: none"> Enter a name in the Database field. Click Create Database. The POMS Wizard will create a database with the name you entered. A message will tell you if the operation was successful. Note: You cannot select an existing RCO database.		
User ID	Login ID required to access the database.		
Password	Password required to access the database.		

3. Click Test Connection.

A message tells you if the connection succeeded.

The RCO database will be available for your project's RCO data. The Next button, which has been disabled, is enabled.

Step 4. Select the TADB Database Connection Parameters

1. Click Next after you have successfully created the RCO database.

The Select TADB database connection options dialog box opens.

2. Provide the following data, as follows.

rect 29, 65, 325, 91 [\(page 803\)](#)
 rect 20, 89, 395, 113 [\(page 803\)](#)
 rect 19, 114, 296, 140 [\(page 803\)](#)
 rect 16, 138, 296, 163 [\(page 803\)](#)
 rect 71, 164, 294, 205 [\(page 803\)](#)
 rect 71, 204, 242, 224 [\(page 803\)](#)

Specification	Description	
Server	The node on which the SQL database is located. Note: If you are using CIMPLICITY SQL, the server will have \CIMPLICITY in it's identification.	
	Default	Node that was entered for the RCO database.
	Opens Data Link Properties dialog box, in which you can find or set up the server if it is not already set up. Note: Microsoft help is available for your configuration.	
Database	The SQL database that will contain the Tracker TADB data. The database can be the same one selected for or different from the RCO database. Important: The TADB database requires SQL Server 2000 and service pack 2 for SQL Server 2000.	
	Create Database	If an RCO database does not exist: a. Enter a name in the Database field. b. Click Create Database. The POMS Wizard will create a database with the name you entered. A message will tell you if the operation was successful.
User ID	Login ID required to access the database.	
	Default	User ID entered for the RCO database
Password	Password required to access the database.	
	Default	Password entered for the RCO database.
Meta data File	The names of meta data files that you copied to the Wizard directory are available in the drop-down list, in addition to the POMS default. The meta data file sets up the tables that will contain your project's Item type, Group and Attribute data.	

3. Click Test Connection.

A message tells you is the connection succeeded. The Finish button, which has been disabled, is enabled.

4. Click Finish.

The POMS Wizard creates your POMS project.

Step 5. Open the New POMS Project

A message tells you when the POMS wizard has completed configuring your POMS project.


1. Click OK to open the project.

A message tells you that the configuration of this project is out-of-date.

2. Click OK to upgrade the project.

CIMPLICITY upgrades and opens the new POMS project.

Step 6. Create a SEQATTRIB Query Expression

 **Important:** A SEQATTRIB query expression is required for the output logic script to schedule order items for production.

Creating this expression requires:

- Knowledge of how to create a basic expression in the Tracker Query Engine.
- An existing TADB attribute assigned to holding a sequence value.

The attribute may have been defined in any of the following ways.

- In the selected meta data file when you created the POMS project.
- In an existing database that you selected in the TrackerCfg_UI and merged into the POMS project when you re-opened the TrackerCfg_UI.
- In the PRT>Advanced>Item Types section of the TrackerCfg_UI.

1. Open the TrackerCfg_UI.
2. Make sure the group and attribute that will be used in the SEQATTRIB query expression are defined.
3. Open the Tracker Query Expression Browser.
4. Create a [new \(page 901\)](#) expression named SEQATTRIB.

Result: This [expression \(page 913\)](#) will query and display an item's sequence.

 **Note:** A simple sequence query expression is `Group.Attribute`


Example

5. A group in a PCOrder Item Type is named Production.
6. An attribute in the Production group is Sequence. This attribute will hold the sequence value for an item.

7. The `SEQATTRIB` query expression is `Production.Sequence`.

POMS Order Regions Overview

The POMS templates defines a number of PRT regions for processing electronic orders. These are virtual regions because they contain virtual items, Orders, rather than physical regions that Product items are routed through.

 **Note:** There is no difference between virtual and physical regions in the PRT model. The practical difference is that the transition point between one virtual region and another must also be virtual, i.e., virtual points set by mouse-clicks, scripts or system events.

rect 97, 58, 218, 85 [\(page 805\)](#)

poly 415, 289, 435, 279, 453, 275, 465, 276, 481, 277, 513, 285, 515, 291, 497, 300, 479, 303, 426, 297, 417, 291 [6. \\$OMX_PLANT \(page 809\)](#)

poly 415, 46, 443, 37, 463, 35, 483, 35, 509, 41, 521, 49, 497, 59, 475, 61, 441, 59, 416, 49 [5. \\$OMX_MASTER \(page 809\)](#)

poly 106, 143, 82, 132, 55, 127, 30, 134, 9, 138, 4, 145, 57, 156, 102, 147 [1. \\$OMV_ORDER \(page 807\)](#)

poly 423, 93, 433, 108, 432, 124, 434, 142, 429, 177, 425, 191, 421, 195, 413, 185, 413, 172, 411, 147, 412, 122, 417, 97 [4. \\$OMX_UPDCOM \(page 808\)](#)

poly 258, 288, 287, 278, 311, 275, 331, 280, 350, 283, 357, 288, 342, 297, 317, 301, 299, 301, 287, 301, 274, 297, 258, 290 [7. \\$OMX_RETURN \(page 809\)](#)

poly 229, 159, 237, 185, 241, 205, 241, 223, 236, 251, 227, 262, 221, 251, 219, 237, 217, 223, 217, 201, 219, 181, 225, 164, 223, 167 [3. \\$OMX_ORUPDT \(page 807\)](#)

poly 229, 21, 239, 41, 242, 71, 239, 97, 235, 113, 229, 121, 224, 118, 219, 101, 217, 75, 217, 41, 227, 21 [2. \\$OMX_ORNEW \(page 807\)](#)

rect 264, 315, 381, 344 [\(page 806\)](#)

rect 97, 275, 215, 302 [\(page 806\)](#)

rect 100, 196, 216, 226 [\(page 805\)](#)

rect 410, 314, 529, 344 [\(page 806\)](#)

rect 408, 231, 523, 255 [\(page 806\)](#)

rect 259, 228, 380, 255 [\(page 806\)](#)

rect 258, 197, 380, 224 [\(page 806\)](#)

rect 260, 165, 380, 192 [\(page 806\)](#)

rect 259, 95, 381, 118 [\(page 806\)](#)

rect 260, 60, 380, 85 [\(page 806\)](#)

rect 261, 24, 378, 49 [\(page 806\)](#)

The virtual order regions in any project based on the TBNG_POMS project template are:

Region	Virtual holder for:
\$OM_ORDER_NEW	New order items; no decision has been made.
\$OM_ORDER_UPDT	Update order items; no decision has been made.

\$OM_ORNEW_ACPT	Accepted new orders
\$OM_ORUPD_ACPT	Accepted update orders.
\$OM_ORNEW_RJCT	Rejected new orders.
\$OM_ORUPD_RJCT	Rejected update orders.
\$OM_ORNEW_DETND	New orders that are being detained; they have been neither accepted nor rejected.
\$OM_ORUPD_DETND	Update orders that are being detained; they have been neither accepted nor rejected.
\$OM_ORDER_RETND	Manually returned orders.
\$OM_MASTER_BLND	Orders involved in master blend (ideal) sequencing.
\$OM_PLANT_BLND	Orders involved in custom plant blend sequencing; this is the last region before actual production.
DETAINMENT	System detainment region.

POMS Routing Overview

POMS Routing Overview

POMS comes with RCO's that provide the basis for routing order items through the validation and categorization process.

These RCO's are:

1 <i>(page 807)</i>	\$OMV_ORDER
2 <i>(page 807)</i>	\$OMX_ORNEW
3 <i>(page 807)</i>	\$OMX_ORUPDT
4 <i>(page 808)</i>	\$OMX_UPDCOM
5 <i>(page 809)</i>	\$OMX_MASTER
6 <i>(page 809)</i>	\$OMX_PLANT
7 <i>(page 809)</i>	\$OMX_RETURN

1. \$OMV ORDER

1. Flags the order item as Update
2. Creates an .xml file with the update order information.

2. \$OMX ORNEWrect 106, -2, 321, 67 [\(page 807\)](#)rect 14, 81, 462, 476 [\(page 807\)](#)

1	An order arrives in the \$OM_ORDER_NEW region.		
2	The Routing Logic Module (function blocks) processes the new order.		
	The validation result, based on business logic built into a Query expression, can be:		
	IGNORE_ORDER	Status	Based on business logic built into a Query expression.
		Action	The Order item is deleted from both PRT and TADB.
	PENDING_ORDER	Status	Not enough information to automatically accept, reject or ignore the order item.
		Destination	\$OM_ORNEW_DET
		Action	Manually rejected
			The Order item is deleted from both PRT and TADB.
			Update accepted through \$OMX_UPDCOM
			Moves back to \$OM_ORDER_NEW
	ACCEPT_ORDER	Status	\$OM_ORNEW_RJCT A TADB tracking record exists and has a sequence attribute.
		Destination	\$OM_ORNEW_ACPT
		Action	Triggers \$OMX_MASTER.
	REJECT_ORDER	Status	Based on business logic built into a Query expression.
		Destination	OM_ORNEW_RJCT
		Action	The Order item is deleted from both PRT and TADB.

3. \$OMX ORUPDTrect 162, -5, 325, 71 [\(page 807\)](#)rect -4, 72, 533, 423 [\(page 807\)](#)

1	An order arrives in the \$OM_ORDER_UPDT region.		
2	The Routing Logic Module (function blocks) processes the update order.		
	The validation result, based on business logic built into a Query expression, can be:		
	IGNORE_ORDER	Status	Ignore the current order item.

	Action	The order item is deleted from both PRT and TADB.	
PENDING_ORDER	Status	Not enough information to automatically accept, reject or ignore the order item.	
	Destination	\$OM_ORUPD_DET N	
	Action	Manually accepted	Triggers \$OM_ORUPD_ACPT
		Manually rejected	The Order item is deleted from both PRT and TADB.
		No action	Stays in \$OM_ORUPD_DET N
ACCEPT_ORDER	Status	The update is accepted.	
	Destination	\$OM_ORUPD_ACPT	
	Action	Triggers the \$OMX_ORUPDT RCO.	
REJECT_ORDER	Status	Based on business logic built into a Query expression.	
	Destination	OM_ORUPD_RJCT	
	Action	The Order item is deleted from both PRT and TADB.	

4. \$OMX UPDCOM

rect 230, -5, 406, 71 [\(page 808\)](#)


rect 230, 69, 404, 128 [\(page 808\)](#)

rect 49, 129, 402, 228 [\(page 808\)](#)

rect 56, 236, 537, 427 [\(page 808\)](#)

rect 3, 427, 525, 600 [\(page 808\)](#)

1	Orders arrive in the \$OM__ORUPD_ACPT.	
2	The Routing Logic Module (function blocks) commits the accepted update order.	
3	.xml retrieval	Deletes the update only item.
	Succeeds	Looks for original Item ID.
4	The original Item ID is retrieved.	
	Not found in PRT, but found in TADB.	Order item is deleted from TADB.
	Not found in TADB, but found in PRT.	Order item is deleted from PRT.
	Found in PRT and TADB attributes	Item ID is retrieved from PRT.
	Action	The update item is merged with the original. The update item is deleted from PRT and TADB.
5	The original item is in \$OM_ORNEW_DET N:	
No	Action	None; update has been merged with the original order.
Yes	Destination	\$OM_ORDER_NEW
	Action	Triggers \$OMX_PLANT

 **Note:** If the PRT Order item for the order to be updated is in any other region, its TADB record will be updated without moving the item.

5. \$OMX MASTER

rect 157, 0, 336, 71 [\(page 809\)](#)

rect 38, 73, 351, 467 [\(page 809\)](#)


1	An order arrives in the \$OM_ORDNEW_ACPT region.	
2	The Routing Logic Module (function block) processes the update order.	
	Decision is:	
	SCHEDULE_ORDER	Orders are scheduled in an "ideal" sequence.
	Destination	\$OM_MASTR_BLND sequence process. The output logic script schedules order items for production according to their sequence attribute value, SEQATTRIB , by moving them to the correct location in the \$OM_MASTR_BLND region.
	If the location process is successful:	
	Yes	The order item continues moving toward production. Triggers \$OMX_PLANT.
	No	The script moves the order item to the \$OM_ORNEW_DET_N region if an error is encountered. Moves to \$OM_ORNEW_DET_N.

6. \$OMX PLANT

rect 2, -3, 150, 67 [\(page 809\)](#)

rect 2, 63, 150, 320 [\(page 809\)](#)

1	An order arrives in the first location \$OM_MASTR_BLND region.		
2	The Routing Logic Module (function block) processes the update order.		
	Decision is:		
	SCHEDULE_ORDER	Status	The order item is scheduled for production.
		Destination	Last location in the \$OMX_PLANT region.

 **Note:** This RCO is where your custom plant sequencing function blocks and/or scripts should be added.

7. \$OMX RETURN

1. Generates a log status message that includes the Order item ID (PRT_ITEM_ID).
2. Deletes the return order item.

5. POMS CimView Order Entry

About CimView Order Entry

Most orders will be entered into your POMS database through .xml files. However, you may find that you have to manually edit an order item or correct an inadvertent error.

POMS contains CimView order entry screens for these type of entries.

rect 79, 2, 343, 133 [About External Scheduling \(page 773\)](#)

rect 224, 490, 356, 551 [About Order Execution Management Broadcast \(page 934\)](#)

rect 419, 147, 519, 237 [About the Tracker Query Engine \(page 893\)](#)

rect 562, 391, 631, 468 [About Range Source Architecture \(page 859\)](#)

rect 295, 168, 414, 258 [About the Tracker Attribute Database \(page 817\)](#)

rect 216, 275, 554, 480 [About the Product Order Management System \(POMS\) \(page 799\)](#)

rect 4, 317, 205, 391 [About DIR_WATCHER.bcl \(page 796\)](#)

rect 4, 125, 192, 305 [About XMLT \(page 773\)](#)

POMS CimView Order Entry Usage

POMS provides CimView screens so if you need to make a manual entry to create or edit an order, you can. Data that you enter is written to PRT (item ID's only) and TADB.

 **guide:** Keep in mind the following when you make manual entries.

If you create a new or update order, make sure you:

- Enter data in ALL fields that do not allow NULLS.
- Adhere to your system's rules for data entry.

For example, the Item ID needs to follow the rules of construction and sequencing.

If you edit a new order, make sure you:

- Modify the item ID according to your system's rules.

An Item ID must be unique in the database. Therefore, when an update is entered, before it is merged with the original order, it must have an identifier that flags it as an update and modifies the item ID so the update can exist in the database. When the update order goes through the process automatically, POMS automatically flags the file.

Your system will have rules governing the construct of the flag. You must manually adhere to these rules when you edit an existing order.

- Enter data in ALL fields that do not allow NULLS.
- Adhere to your system's rules for data entry.

Step 1 (page 811)	Open POMS_MAIN.cim.
Step 2 (page 811)	Create an order.
Step 3 (page 813)	Enter attribute values.
Step 4 (page 814)	Browse order queues.
Step 5 (page 816)	Edit an order.
Step 6 (page 816)	Clone a selected order.

Step 1. Open POMS_MAIN.cim

1. Start a POMS project.
2. Double-click POMS_MAIN.cim in the Workbench right pane.


A POMS project login box may open the first time you try to open POMS_MAIN.cim after the project starts running.

The POMS_MAIN.cim CimView screen opens.

3. Enter a correct **User ID** and **Password**.
4. Click OK.

The POMS_MAIN.cim screen opens displaying an Order Management Menu.

Step 2. Create an Order

 **Note:** Order data will almost always be entered through **.xml** files; it is likely that CimView will never be used to create a new order. However, CimView does offer the capability.

1. Click Create Order on the Order Management Menu.

The DATA_ENTRY.cim screen opens.

2. Select options in the following sequence to create the order.

rect 8, 65, 223, 92 ([page 812](#))

rect 299, 61, 364, 96 ([page 813](#))

rect 160, 168, 393, 231 ([page 813](#))

rect 158, 237, 390, 306 ([page 813](#))

rect 228, 59, 299, 96 ([page 813](#))

rect 239, 98, 384, 127 ([page 812](#))

rect 9, 98, 225, 156 ([page 812](#))

	Option	Description
A	Order Type	Corresponds to the item type in PRT and TADB.
		The Order type that you select is placed in the item type category in PRT and TADB. When you make a select a message will ask you if you are sure. Click OK to continue.
B	Order Id	Becomes the Item ID in PRT and the SQL database.
		Important: This number must be unique.
		Note: The number may also be an attribute of the selected order.
C	Queue	Places the order in the queue for the specified region.
		Options are:
	Accepted New	New order that can go to production.
	Pending New	New order on hold.
	Rejected New	New order that will not be produced.
	Accepted Update	Order with changes to an existing order that have already been approved for production.
	Pending Update	Order with changes to an existing order waiting for approval, rejection or return.
	Rejected Update	Update request is rejected.
	Master Blend	It is not advisable to create orders for the Master Blend or Plant Blend regions.
	Plant Blend	
	New	New order that has not been evaluated.

			Update	Order with changes to an existing order that have not been evaluated.
Up (page 812)			Return	If there is more than one POMS project or plant, this queue holds orders that do not belong in the current POMS project and need to be rerouted
	D	Save		Saves the new order in both PRT and the TADB (page 817) database.
				Note: A message asks for confirmation. Click Ok to save the entries.
	E	Attribute Fields		Attributes that are in Normal groups. All fields that do not allow NULLS must be filled in.
	F	List Group Member buttons		Clicking any Attribute List button opens LIST_EDIT.cim, the Attribute List screen, on which you can enter list attribute values.
	G	Exit		Closes the DATA_ENTRY.cim. Create order screen. Returns you to the POMS_MAIN.cim.

3. Click Exit to close.

Step 3. Enter Attribute Values

You can create an attribute list for a new item or modify it for an existing item through a LIST_EDIT.cim screen.

Screen features and options are as follows.

rect 8, 132, 417, 320 [\(page 813\)](#)

rect 8, 107, 428, 130 [\(page 814\)](#)

rect 8, 81, 431, 107 [\(page 813\)](#)

rect 220, 59, 432, 83 [\(page 813\)](#)

rect 10, 61, 217, 83 [\(page 813\)](#)

	Feature/ Option	Description	
A	Order	Item ID of new or selected order item.	
B	Group	Group that includes the selected attribute list.	
C	Attribute Lists	Each column is an attribute in the selected group. All attributes that do not allow NULL values need to have at least one entry.	
D	Toolbar buttons	Add New	
		Delete	Deletes a selected row or rows.
		Save	Saves the screen and adds new values to the TADB database.
		Cancel	Cancels your entries.
		Select All	Selects all the rows on the Attribute List screen.

		Clear Selected	Clears the selected values.
E	Value Entry fields	Screen opened:	
		The first time.	Each writable field corresponds to the column underneath it. Your entry will be added to the: <ul style="list-style-type: none"> • CimView list when you click • Attribute column in the TADB when you click Save.
		For edit.	The values for the item's attributes in the TADB database display when you open the Attribute List. You can continue adding more attributes the same way you do for a new screen.

guide: Guidelines

1. Use the Backspace key then type in new values to correct or change entries in the order you are creating.

Note: You can use this method to create new orders without exiting the Create Order screen.

2. Do not use the number pad to enter numbers; use the keyboard numbers.

Step 4. Browse Order Queues

1. Click Browse Order Queues on the Order Management Menu.

A Browse orders screen opens.

You can use this screen to view and make manual decisions for existing orders in any selected region.

Screen features and options are as follows.

rect 6, 108, 407, 319 ([page 814](#))

rect 7, 63, 408, 95 ([page 815](#))

rect 7, 35, 296, 67 ([page 814](#))

	Feature/Option	Description	
A	Browse Orders In	The drop-down list contains the regions in your project. Select the region with the orders you want to view.	
B	Order Lists	The following columns display the orders in and overview details for orders in the selected region.	
		Location	The first column displays the location of the order in the region. Columns display in the order of the location sequence beginning with 0.

		Order ID	The ID of the order items in the region. A single order's sequence in the list corresponds to its location in the region.
		Reference Id	If an order item has a reference ID, its reference ID displays on the same row.
		Class	The class of the order item on the same row.
C	Toolbar buttons	Accept	Sends selected:
			<ul style="list-style-type: none"> • New orders to the \$OM_ORNEW_ACPT region. • Update orders to the \$OM_ORUPD_ACPT region.
		Reject	Sends selected: <ul style="list-style-type: none"> • New orders to the \$OM_ORNEW_RJCT region. • Update orders to the \$OM_ORUPD_RJCT region.
		Ignore	Designates selected orders to be Ignored. POMS deletes ignored orders.
		Return	Sends an order to the \$OM_ORDER_RETN region.
		Edit	DATA_ENTRY.cim Edit Order (page 816) screen
		Clone	Opens a DATA_ENTRY.cim Create Order (page 816) screen
		Select all	Selects (and highlights) all of the order items that are listed in the selected region.
		Create New	Opens a Create New (page 811) screen.
		Filter/Refresh	Filters and refreshes (page 815) the screen.
		Clear Selected	De-selects a selection. Clears the highlight.
		Exit	Closes the ORDERS.cim. Returns you to the POMS_MAIN.cim.

Filter/Refresh

Filter the list

2. Enter an Order ID in the Filter/Refresh field located on the ORDERS.cim toolbar.
3. Click Filter/Refresh.

Result: The specified order is the only order that displays in the list of orders.

Refresh the list

4. Clear the Filter/Refresh field.
5. Click Filter/Refresh.

A list of all of the orders in the region displays.


Step 5. Edit an Order

1. Select an order on the ORDERS.cim screen.
2. Click Edit on the ORDERS.cim toolbar.

A DATA_ENTRY.cim Edit Order screen opens with data from the selected order.

You can create an update order as follows.

A	Modify the original Order Id (Item ID) according to your system's Update Order rules.
B	Edit or add attribute values where necessary.

 **Important:** Change any values that must be unique.

3. Click Save to save the update order item.

The update order item is entered in PRT and TADB; the item's attribute values are entered in TADB.


Step 6. Clone a Selected Order

1. Select an order on the ORDERS.cim screen.
2. Click Clone on the ORDERS.cim toolbar.

A DATA_ENTRY.cim Create Order screen opens with data from the selected order.

You can create a new order as follows:

A	Fill in the Order Id field.
B	Edit or add attribute values where necessary.

 **Important:** Change any values that must be unique.

3. Click Save to save the new order item.

The new order item is entered in PRT and TADB; the item's attribute values are entered in TADB.

6. Tracker Attribute Database

About the Tracker Attribute Database

The Tracker Attribute Database (TADB) stores comprehensive data about items, including orders and product components.

rect 88, 4, 347, 131 [About External Scheduling \(page 773\)](#)
 rect -2, 129, 194, 317 [About XMLT \(page 773\)](#)
 rect 5, 333, 189, 385 [About DIR_WATCHER.bcl \(page 796\)](#)
 rect 217, 272, 562, 487 [About the Product Order Management System \(POMS\) \(page 799\)](#)
 rect 6, 389, 196, 486 [About CimView Order Entry \(page 810\)](#)
 rect 564, 397, 631, 462 [About Range Source Architecture \(page 859\)](#)
 rect 271, 152, 414, 268 [Tracker Attribute Database Overview \(page 817\)](#)
 rect 423, 149, 523, 239 [About the Tracker Query Engine \(page 893\)](#)
 rect 224, 490, 356, 551 [About Order Execution Management Broadcast \(page 934\)](#)

Tracker Attribute Database Overview

The Tracker Attribute Database (TADB) stores Tracker data, including PRT backing file data.


- TADB features/configuration.
- PRT Collector in TADB overview.
- Region with an item allocated capacity requirements

TADB Features/Configuration

rect 185, 13, 208, 39 [1. TADB Enabled \(page 819\)](#)
 rect 459, 217, 482, 243 [1. TADB Enabled \(page 819\)](#)
 rect 221, 150, 247, 180 [2. TADB SQL Table Structure Overview \(page 823\)](#)
 rect 283, 208, 397, 226 [Option 4.2.2.1. Set Item Attributes \(except ASSOCIATE attributes\) \(page 221\)](#)
 rect 284, 240, 390, 255 [Option 1.4. Use a Criteria Set as a Range Source \(page 872\)](#)
 rect 283, 224, 389, 241 [About Range Source Architecture \(page 859\)](#)
 rect 499, 66, 525, 90 [3. TADB Meta-Data Configuration \(page 828\)](#)
 rect 182, 192, 209, 213 [4. TADB/PRT Backing File Synchronization and Recovery \(page 848\)](#)
 rect 27, 50, 59, 71 [4. TADB/PRT Backing File Synchronization and Recovery \(page 848\)](#)
 rect 16, 177, 43, 198 [4. TADB/PRT Backing File Synchronization and Recovery \(page 848\)](#)

1 (page 819)	TADB Enabled
---	--------------

2 (page 823)	TADB SQL Table Structure Overview
3 (page 828)	TADB Meta-Data Configuration
4 (page 848)	TADB/PRT Backing File Synchronization and Recovery

 **Note:** In most cases the groups and attributes will be defined at the Enterprise Resource Planning (ERP) level. As the system designer, you can take these categories and enter the information into the TADB configuration.

PRT Collector in TADB

The PRT Collector writes to the PRT backing files and the TADB database.

The PRT Collector:


- Gets point updates to move items, requests to .add items, delete items
- Responds back only after it successfully updates the database.

If the database connection is lost or the project is in the process of failing over a SQL cluster the PRT Collector will:

- Continue to retry updating the database until the connection is restored.
- Generating an alarm that it has lost the database connection.

The alarms raise an alert as follows.

Alarm	Message
PRT_DB_COMM	PRT database command failed. Check the status log for details.
PRT_DB_CONN_DOWN	PRT has lost the database connection

 **Note:** The PRT Collector writes to the TADB database. It is important to bring the database server back online to restore a lost connection as quickly as possible. If not, in extreme cases, data can be lost.

Region with an Item Allocated Capacity Requirements

(On a node configured to use TADB) if you configure a region with an allocated item capacity

(Locations * Location_Capacity * 2) greater than 32767, Tracker will not run correctly on its next startup.

Example

- Configure a region with 8191 locations 2 items per location, the `alloc_item_capacity` ([page 826](#)) is 32764.

Your project starts without complaint and Tracker has no problem.

- Re-configure the region to have 8192 locations and restart the project.


The core status log receives the message TADB Item count 0 exceeds Item capacity -32768 for region.

Subsequently, an attempt to open PRT_UI will cause PRT_DS to crash with an application error.

Looking directly at the `TrkRegion` table in the TADB shows the `alloc_item_capacity` field has the number 32768.

1. TADB Enabled

1. TADB Enabled

 **Note:** All projects must use the same TADB database.

The Tracker Collector (TrkCollector) uses the database configuration for the first project that starts on the server.

- The process the Tracker Collector follows to determine where to store TADB data is as follows.

1. If the project is connected to a TADB database, the TADB database is used.
2. If Tracker Collector does not find a TADB database it looks in the `Trkattribute.cfg` file to determine if there is a TADB database or if it should use backing files.
3. If there is no direction in the `Trkattribute.cfg` file TADB will not store TADB data.
 - Tracker Collector uses the same storage (or no storage) for all projects that start after the first project.
 - When a project starts, the PRT Collector:
4. Determines what storage vehicle the Tracker Collector uses
5. Uses the same storage vehicle.

The following two steps enable TADB.

Step 1 (page 820)	Enable TADB in the CIMPLICITY project.
Step 2 (page 820)	Create a TADB database.

Step 1. Enable TADB in the CIMPLICITY Project

1. Click Project>Properties on the Workbench menu bar.

The Project Properties dialog box opens.

2. Select the General tab.
3. Check TADB in the **Options** box.

The Tracker Attribute Database is enabled and ready for configuration.

- If Tracker and Order Execution Management were not previously checked, they are checked automatically.
- The TADB option is checked and changed to read-only if the Query Engine is checked.

Step 2. Create a TADB Database

- Important Notes before creating a TADB Database
- Create a TADB Database

Important Note Before Creating a TADB Database

This procedure needs to be done on SQL Servers where Tracker Query is going to be installed or upgraded, including both nodes of a cluster and also on a Historical TADB server.

Tracker Query requires a file, mfc90.dll, in order to operate correctly.

Important: Mfc90.dll must be installed before Proficy CIMPLICITY is installed.

Do the following.

1. Insert the Proficy CIMPLICITY CD in the CD drive on the SQL Server server.
2. Expand ...\\Setup\\Microsoft\\VisualStudio.
3. Double-click vcredist_x86.exe
4. Follow any installation instructions.

Result: The required file will be installed.

Note: You can also search for vcredist_x86.exe at the Microsoft Web site.

Mfc90.dll is installed automatically on machines where CIMPLICITY is installed.

Create a TADB Database

5. Install Proficy SQL

Note: If you used the Wizard to create a POMS project this step is completed. However, you can also follow all of these steps or start with 3 in this step, if you want to create and assign or assign a different existing TADB database from the one you assigned when you used the Wizard.

6. Create a new database in the SQL Server that will be used for the TADB.

7. Open the TrackerCfg_UI in an existing project.

Note: The project can be on any node.

If this is the first time you open the TrackerCfg_UI, an Options dialog box displays, in which you select the RCO database.

Once you have selected the RCO database, a TADB Datasource Specification dialog box opens.

8. Fill in the fields, as follows.

Field	Description
Driver	SQL Server.
Server	Name of the server that has the database.
Database	Database that was created in the SQL Enterprise
User ID	User who has access to the database.
Password	Required to access the database.
Build	Opens Data Link Properties dialog box, in which you can find or set up the server if it is not already set up. Note: Microsoft help is available for your configuration.
Enable Historical Database (page 837)	At least a Standard edition of Microsoft SQL Server is required..

9. Click OK.

Tracker searches for and connects to the database.

10. Actions in the remainder of this step depend on the status of the connected database.

The database is not an established TADB or Query Engine database.
The database is an existing TADB database with data.

The database is not an established TADB or Query Engine database.

The following messages display.

- a. The current database...is not a TADB database. Do you want to create one now?

Click Yes to create the selected database as a TADB database.

- a. The current database...is not a Query Engine database. Do you want to create one now?

Click Yes to create the selected database as a Query Engine database.

- a. A Configuration Login dialog box opens.
- a. Enter a user name and password that has privileges to configure Tracker.
- a. Click OK.

The TrackerCfg_UI window becomes available for configuration.

[Up \(page 822\)](#)

The database is an existing TADB database with data.

1. Click the Save button on the TrackerCfg_UI toolbar to save the configuration.
2. Close the TrackerCfg_UI.
3. Click the Configuration Update button on the Workbench toolbar.
4. Re-open the TrackerCfg_UI.

If data has been imported into the TADB database a message displays.

Options are:

A	Does not allow you to modify the TADB database.
	If you choose this option, this message will display every time you open the TrackerCfg_UI until the PRT Item Types configuration is the same as the configuration in the SQL database.
B	Updates the data in PRT Item Types. TADB configuration is used if there is a conflict or differences.

1. Check (B) Merge display and enable modification...to update PRT Item Types with the TADB configuration.
2. Click OK.

3. Select PRT>Advanced>Item Types in the TrackerCfg_UI left pane.

The TADB meta data categories display in the TrackerCfg_UI PRT>Advanced>Item Types right pane.

2. TADB SQL Table Structure Overview

2. TADB SQL Table Structure Overview

The TADB tables and relationships are as follows.

rect 65, 219, 222, 308 [A. ItemTypes Table \(page 823\)](#)

rect 228, 81, 377, 181 [B. Items Table \(page 824\)](#)

rect 217, 1, 369, 65 [C. TrkAttributes Table \(page 825\)](#)

rect 267, 219, 421, 428 [D. TrkRegions Table \(page 825\)](#)

rect 427, 110, 575, 171 [E. TrkGroups Table \(page 827\)](#)

rect 0, 110, 149, 163 [F. RelatedItems Table \(page 827\)](#)

rect 11, 167, 119, 204 [G. SQL Table Relationships \(page 827\)](#)

A (page 823)	ItemTypes
B (page 824)	Items
C (page 825)	TrkAttributes
D (page 825)	TrkRegions
E (page 827)	TrkGroups
F (page 827)	RelatedItems
G (page 827)	Relationships

A. ItemTypes Table

Column	Data Type	Length	Description	Allow Nulls
--------	-----------	--------	-------------	-------------

P	pKey	bigint	8	Item Type primary key <ul style="list-style-type: none"> Automatically generated Internally used only 	
	id	varchar	16	Item Type identifier	
	description	varchar	40	User description of the Item Type	
	code	char	10	Item Type code used for Type Detector in Translation Type Code.	
	class	varchar	16	Item Type class	
	deleted	bit	1	Flag to indicate the Item Type has been deleted (Internally used only in Historical TADB configuration)	
	tracking_type	int	4	Item Type Tracking Type code	
				1	Serialized
				2	Non-serialized

P = Primary key

B. Items Table

	Column	Data Type	Length	Description	Allow Nulls
P	pKey	bigint	8	Items primary key. <ul style="list-style-type: none"> Automatically generated. Internally used only. 	
	id	varchar	20	Item unique tracking ID.	
	refId	varchar	20	Item reference ID of Item. Note: The reference ID can be seen through the PRT_UI.	
	fItemtype	bigint	8	Foreign Key to the Item's Item Type; Item Type is found in the ItemTypes (page 823) table.	
	created	datetime	8	Date and time the Item was created.	
	modified	datetime	8	Date and time the Item was last modified.	
	active	bit	1	Flag to indicate the Item is actively in the production process. Internally used <ul style="list-style-type: none"> For Historical TADB configuration only To support the Archive method. 	
	externalHoldActive	bit	1	Flag indicating if an External Hold is active on the Item	

groupID	varchar	20	Foreign key to the Tracking Group Id; Group ID is found in the TrkGroups (page 827) table
internalHoldActive	bit	1	Flag to indicate if an Internal Hold is active for the Item
internalHoldReason	varchar	40	Reason for an active Internal Hold
itemStatusHI	int	4	High word of Item status bits.

P = Primary key

C. TrkAttributes Table

	Column	Data Type	Length	Description	Allow Nulls
P	fItems	bigint	8	Foreign Key to Item; Item is found in the Items table.	
P	name	varchar	32	Attribute name.	
P	class	int	4	Attribute class.	
			0	Extended	
			1	Standard	
			2	Hold	
	[value]	sql_variant		Attribute value.	
	pos	tinyint	1	Attribute's position in the Standard Attribute array.	
			1-20	Standard Attributes	
			0	Extended and Holds	
	[\$timeStamp]	datetime	8	Date and time of last modification (Internal)	

P = Primary key

D. TrkRegions Table

	Column	Data Type	Length	Description	Allow Nulls
P	pKey	bigint	8	Region primary key. Internally used only	
	regionID	varchar	16	Region unique ID.	
	region_status	int	4	Region status word.	

tracking_type	varchar	16	Region tracking type. Types are: <ul style="list-style-type: none"> • Sequential • Pool • Shifting • Detainment 	
region_type	int	4	Region sub type	
			1	Normal
			2	Combine
			3	Disperse
cfg_locations	int	4	Number of locations that are configured.	
cfg_items_per_location	int	4	Configured location capacity.	
cfg_item_capacity	int	4	Configured Item capacity The capacity is: <code>cfg_locations*cfg_items_per_location</code>	
alloc_locations	int	4	Number of locations allocated. The number can be up to: <code>2 * cfg_locations</code> .	
alloc_item_capacity	int	4	Number of Items allocated. The number is: <code>alloc_locations*cfg_items_per_location</code>	
total_items_in_region	int	4	Number of Items currently in the Region	
occupied_locations	int	4	Number of Locations occupied by at least 1 Item	
det_region_id	varchar	16	Recursive foreign Key to the TrkRegions table; identifies the Detainment Region for this Region	
auto_assoc_region	bit	1	Flag to indicate if this is an Auto Association Region	
tran_region_id	varchar	16	Recursive foreign Key to TrkRegions table; identifies the Transition Region for this Region. Important: tran-region_id is valid only for Shifting regions.	
total_items_exiting_region	int	4	Number of Items that have exited this Region	
total_items_to_repair	int	4	Number of Items that have exited this Region and been moved to Detainment.	
PRTC_VALSTATS_MODIFY_MASK	int	4	Internal validation statistics	
PRTC_VALSTATS_TOTAL_VALIDATIONS	int	4	Internal validation statistics	
PRTC_VALSTATS_INVALID_COUNT	int	4	Internal validation statistics	

	PRTC_VALSTATS_TOTAL_INVALID	int	4	Internal validation statistics	
	PRTC_VALSTATS_TOTAL_DBL_TRIGGER	int	4	Internal validation statistics	
	PRTC_VALSTATS_TOTAL_MISMATCH	int	4	Internal validation statistics	
	PRTC_VALSTATS_CONS_MISMATCH	int	4	Internal validation statistics	
	PRTC_VALSTATS_TOTAL_NOREAD	int	4	Internal validation statistics	
	PRTC_VALSTATS_CONS_NOREAD	int	4	Internal validation statistics	
	[\$timeStamp]	datetime	8	Datetime of last modification (Internal)	

P = Primary key

E. TrkGroups Table

	Column	Data Type	Length	Description	Allow Nulls
P	pKey	bigint	8	Group primary key. Internally used only	
	groupID	varchar	20	Tracking Group unique ID.	
	hold_checking_enable	bit	1	Flag to indicated if an External Hold is active for the Tracking Group	
	group_hold_reason	varchar	40	Text reason for an active External Hold	
	[\$timeStamp]	datetime	8	Date and time of the last modification (Internal)	

P = Primary key

F. RelatedItems Table

	Column	Data Type	Length	Description	Allow Nulls
P	pltems	bigint	8	Parent Item key in the relationship	
P	cltems	bigint	8	Child Item key in the relationship	
	[\$timeStamp]	datetime	8	Date and time of the last modification (Internal)	

P = Primary key

G. SQL Table Relationships

The relationships and primary and foreign keys among the basic TADB tables are as follows.

	Relationship Name	Primary key Table	Primary Key	Foreign key table	Foreign Key
i	FK_Items_ItemTypes	ItemTypes	pKey	Items	fltemType

ii	Fk_TrkAttributes_Items	Items	pKey	TrkAttributes	fltems
iii	FK_Items_TrkRegions	TrkRegions	regionID	Items	region1ID
iv	FK_Items_TrkGroups	TrkGroups	groupID	Items	groupID
v	FK_Relateditems_Items_Child	Items	pKey	RelatedItems	cltems
	FK_Relateditems_Items_Parent	Items	pKey	RelatedItems	pltems

3. TADB Meta-Data Configuration

3. TADB Meta-Data Configuration


3.1 (page 828)	Configure the TADB meta-data through the TrackerCfg_UI.
3.2 (page 848)	Use the TADB import/export utility.

3.1. Configure the TADB Meta-Data through the TrackerCfg_UI

3.1. Configure the TADB Meta-Data through the TrackerCfg_UI

Step 1 (page 828)	Enter Item Types in the TrackerCfg_UI.
Step 2 (page 830)	Configure groups for the TADB.
Step 3 (page 830)	Configure attributes for TADB.
Step 4 (page 836)	Update the TADB configuration.
Step 5 (page 836)	View TADB revisions (updates)
Step 6 (page 837)	Configure a Historical database

Step 1. Enter Item Types in the TrackerCfg_ui

 **Note:** The Item Types component will be blank if you are creating the TADB configuration for the first time. Entries will display if you attached an existing TADB database and merged the data. You can follow these steps to edit existing configurations.

1. Expand the PRT>Advanced folders in the TrackerCfg_UI left pane.
2. Double-click **Item Types**.

The **Types** box (and its subordinate boxes) display in the TrackerCFG_UI right pane.

Note: If XML meta-data has been imported, the imported Item Types will display after you merge the databases.


3. Click the New button above the Types box.

The PRT Item Type Configuration dialog box opens.

4. Fill in the fields as follows.

Field	Description	
Item Type ID	Names the item that will be tracked by PRT and stored in the SQL database.	
Description	Informative description of the item.	
Item Type Code	An internally used reference.	
	Default	The next higher number from the last assigned code. Note: If you delete the item to which a code is assigned, that number is not re-assigned.
Class	Class to which the item belongs. A class can have one or more items. This category enables you to get or make decisions that apply to related items.	
Tracking Type	Serialized (page 130)	Check if you want to positively identify (and track) each item as it moves through the process.
	Non-serialized (page 131)	Check if the item does not have to be tracked.

5. Click OK.
6. Click the Save button on the TrackerCfg_UI toolbar to save your new entries.

 **Note:** The POMS project includes a default item type, **ORDER**. You can edit the name or delete it if you don't want to use it.

The type you created displays in the Types box. Item data will be stored in TADB when you do a [TADB Configuration \(page 836\)](#) Update and PRT when you do a CIMPLICITY configuration update..

Step 2. Configure Groups for the TADB

 **Important:** Groups are entered into the TADB database only. They are not recorded in PRT.

1. Select an Item Type in the Types box.
2. Click above the **Groups** box.

The TADB Attribute Group Configuration dialog box opens.

3. Fill in the fields, as follows.

Field	Description	
Item Type		(Read only) Displays the selected item Type ID (page 828) .
Name		Name of the attribute to match. If you fill in a name without an expression, then the attribute is returned if it exists.
Type	List	An attribute associated with the group can consist of more than one unit. For example, a computer may have an attribute that represents a hard drive. However, a user may order two hard drives, not just one.
	Normal	Each attribute in the group must be a single entity.

4. Click OK.
5. Click the Save button on the TrackerCfg_UI toolbar to save your new entries.

The groups will be written to the TADB database when you do a [TADB Configuration \(page 836\)](#) Update.

Step 3. Configure Attributes for the TADB

- Open the TADB Attribute Configuration dialog box.
- Configure TADB attributes.
- Important TADB attribute notes.

1. Open the TADB Attribute Configuration Dialog Box

Data for these attributes can be accessed through Order Execution Management Query.

1. Select a group in the **Groups** box.
2. Click above the **Attributes** box.

A blank TADB Attribute Configuration dialog box opens.

Configure TADB Attributes

Fill in the fields, as follows.

rect 23, 25, 279, 60 [\(page 832\)](#)
 rect 23, 59, 279, 94 [\(page 832\)](#)
 rect 23, 94, 279, 114 [\(page 832\)](#)
 rect 22, 117, 386, 137 [\(page 832\)](#)
 rect 18, 143, 275, 163 [\(page 832\)](#)
 rect 17, 167, 197, 187 [\(page 833\)](#)
 rect 20, 193, 391, 213 [\(page 833\)](#)
 rect 19, 220, 386, 240 [\(page 833\)](#)
 rect 15, 244, 352, 273 [\(page 834\)](#)

A (page 832)	Item Type
B (page 832)	Attribute Group
C (page 832)	Name
D (page 832)	Description
E (page 832)	Data Type
F (page 833)	Length
G (page 833)	Default Value
H (page 833)	F Group
I (page 834)	Check boxes

A	Item Type
---	-----------

(Read only) Displays the selected [Item Type ID \(page 828\)](#) .

[Up \(page 831\)](#)

B	Attribute Group
---	-----------------

(Read only) Displays the [attribute group \(page 830\)](#) that is selected in the Group box.

C	Name
---	------

Attribute name.

[Up \(page 831\)](#)

D	Description
---	-------------

Description of the attribute.

E	Data Type
---	-----------

Choose one of the following SQL Server data types.

Note: Microsoft MSDN is the source for the following definitions.

	bigint	"Integer (whole number) data from -2^{63} (-9,223,372,036,854,775,808) through $2^{63}-1$ (9,223,372,036,854,775,807). Storage size is 8 bytes. bigint is intended for special cases where the integer values may exceed the range supported by the int data type. The int data type remains the primary integer data type in SQL Server."
	bit	Integer data type 1, 0, or NULL.
	char	Fixed-length non-Unicode character data with length of n bytes. n bytes must be a value from 1 through 8,000. Storage size is n bytes. The SQL-92 synonym for char is character.
Up (page 831)	datetime	Date and time data from January 1, 1753 through December 31, 9999, to an accuracy of one three-hundredth of a second (equivalent to 3.33 milliseconds or 0.00333 seconds). Values are rounded to increments of .000, .003, or .007 seconds.
	int	"Integer (whole number) data from -2^{31} (-2,147,483,648) through $2^{31} - 1$ (2,147,483,647). Storage size is 4 bytes. The SQL-92 synonym for int is integer."
	nchar	Fixed-length Unicode character data of n characters. n characters must be a value from 1 through 4,000. Storage size is two times n bytes. The SQL-92 synonyms for nchar are national char and national character.
Up (page 831)	nvarchar	Variable-length Unicode character data of n characters. n characters must be a value from 1 through 4,000. Storage size, in bytes, is two times the number of characters entered. The data entered can be 0 characters in length. The SQL-92 synonyms for nvarchar are national char varying and national character varying.

	real	Is a floating point number data with the following valid values: $-3.40E + 38$ through $-1.18E - 38$, 0 and $1.18E - 38$ through $3.40E + 38$. Storage size is 4 bytes. In SQL Server, the synonym for real is float(24). Note Microsoft® SQL Server™ 2000 treats n as one of two possible values. If $1 \leq n \leq 24$, n is treated as 24. If $25 \leq n \leq 53$, n is treated as 53.
	smalldatetime	Date and time data from January 1, 1900, through June 6, 2079, with accuracy to the minute. smalldatetime values with 29.998 seconds or lower are rounded down to the nearest minute; values with 29.999 seconds or higher are rounded up to the nearest minute.
Up (page 831)	varchar	Variable-length non-Unicode character data with length of n bytes. n bytes must be a value from 1 through 8,000. Storage size is the actual length in bytes of the data entered, not n bytes. The data entered can be 0 characters in length. The SQL-92 synonyms for varchar are char varying or character varying."

F	Length
---	--------

The following data types require a specified length.

The default length is 10. Review the data type definition to see the maximum allowed value.

- [char \(page 832\)](#)
- [nchar \(page 832\)](#)
- [nvarchar \(page 832\)](#)
- [varchar \(page 833\)](#)

[Up \(page 831\)](#)

G	Default Value
---	---------------

The value that is used, if no other value is specified.

H	F Group
---	---------

(When F Group is used the attribute has to be a [bigint \(page 832\)](#) data type)

If the group to which this attribute is related needs to be associated with another group, F Group enables you to specify the other group.

Example

You have two item types:

Body components	<ul style="list-style-type: none"> • Includes the related groups and attributes associated with a vehicle that is being manufactured. • Has an installed parts group that includes the parts actually installed in the vehicle.
Vehicle Order	<ul style="list-style-type: none"> • Includes the related groups and attributes that are ordered for a vehicle. • Has a parts group that specifies the parts that are to be installed in the vehicle.

In order to keep track of parts, you need to associate the parts that were actually installed in a body component with the parts specified in a vehicle order.

You can:

- Create a primary key attribute for the Body Component Parts group that is the primary key in the Vehicle Order>Parts relational database and
- Specify the Vehicle Order>Parts group as the F Group.

Result: The body component parts can now be associated with the Vehicle Order parts.


[Up \(page 831\)](#)

I	Check Boxes
---	-------------

Check any of the check boxes as follows.

- Allow Nulls

If null values are not allowed, you must enter a default value.

 Tip: If it is not required, make sure you check Allow Nulls. This can avoid entry errors, if there is no value for the attribute when an order is entered through [.xml \(page 773\)](#) or [CimView \(page 810\)](#) .

- Indexed

Check to index the attribute if it will be included frequently in a search, to speed up the search.

- Unique Part

Like a primary key, all of the attributes checked as a unique part are combined to form a unique constraint in a group.

Example

- a. First name and last name are checked as unique attributes.

Joe is a first name.

Smith is a last name.

There can't be more than one Joe Smith.

There can be three Joe's and three Smiths.

There may, in fact, be more than one Joe Smith.

- a. Middle name and Street Address are also checked as Unique Parts to resolve this potential problem.

Now there can't be more than one individual with the name Joe Frank Smith, who lives at 1 Columbia Street.

- Lookup

(For list groups) The Lookup feature takes advantage of the relational database capability, using the primary key to eliminate duplicate reporting of attribute values.

This can be a powerful benefit, particularly when there are several records or several attributes related to a single group.

Example

A Feature Codes List group for a Vehicle Order item type includes an attribute, Description. Because this attribute is a description of a feature code, it can appear repeatedly in vehicle orders.

Lookup is checked.

As a result, instead of taking the space in the table line and in the database the value is stored in a single instance in the relational database.

[Up \(page 831\)](#)

3. Click OK.
4. Click the Save button on the TrackerCfg_UI toolbar to save your new entries.

Results:

- The attributes display when you select the group to which they are associated.
- Attribute configuration is written to the TADB database when you do a TADB Configuration Update.
- Values for these attributes can be retrieved through Tracker Query for analysis and decision making.

Important: Important TADB Attribute Notes

- TADB attributes are entered into the TADB database only. They are not recorded in PRT.
- If your system is configured to use the [Historical \(page 837\)](#) TADB, it is bound by limits on the Microsoft SQL Server that Tracker Configuration does not enforce.

The limits are that a TADB attribute group can have the following maximum number of attributes.

Attribute	Maximum Number
Non-lookup	250
Lookup (page 834)	250
Total	500

Step 4. Update the TADB Configuration

1. Do one of the following:

Method 1

Click on the TrackerCfg_UI toolbar.

Method 2

- a. Select Tools on the TrackerCfg_UI menu bar.
- b. Select TADB Configuration Update from the menu.

Method 3

Press Ctrl+U on the keyboard.

The Update Configuration dialog box opens when you use either method.

2. Enter a comment in the to explain what is being updated.

3. Click OK.

The database configuration is updated.

A message reports if there are either:

- No unreported errors or
- Configuration update failed.

Step 5. View TADB Revisions (Updates)

1. Click Tools on the TrackerCfg_UI menu bar.

2. Select View TADB Revisions.

A Revisions browser opens.

Column	Description
Rev	Sequence in which revision was made; 1 is the oldest.
Date	Date of the revision
UID	User identification
Comment	Comment entered by the user when the revision was made.

3. Click Details to display more details about a selected revision.
4. Click Rollback to open the selected revision of the TADB database.

Step 6. Configure a Historical Database

Step 6. Configure a Historical Database

- Historical Database Overview
- Configuration/Maintenance Steps

Historical Database Overview

- A Historical Database requires at least a Standard Edition of Microsoft SQL Server.
- When you use a historical database your project data is stored as follows.

The SQL TADB database:

- Holds the following data during production: Standard attributes, Extended attributes, and TADB data.
- Deletes most items when production is complete.
- The Historical database:
- Receives all the data that passes through the TADB database. There may be an 8 second or more delay before the Historical database receives the data. The exact amount of time depends on your system configuration.
- Retains all the data that passes through the TADB database. When data is entered into this database it is read-only; it can be [removed \(page 840\)](#) from the Historical TADB based on **time constraints** or **item data based rules**.
- Keeps an account of all the changes made within any single record.

Therefore, as long as you keep the Historical Database running, you can reference data for any item, which has not been removed, from the time it started.

Configuration/Maintenance Steps

Steps to create and use the Historical Database are as follows.

Step 6.1 (page 838)	Create a database in SQL Server
Step 6.2 (page 838)	Specify the database to be the Historical Database.

Step 6.3 (page 840)	Start/stop Historical Database activity.
Step 6.4 (page 840)	Perform TADB/Historical TADB Maintenance

Step 6.1. Create a Database in SQL Server

1. Open the SQL Server Management Studio.
2. Do the following.

A	Right-click Databases .
B	Select New Database on the Popup menu.

A Database Properties dialog box opens.

3. Do the following.

A	Select General in the New Database dialog box left-pane.
B	Enter the Historical database name in the Database name field.
C	Click OK.

- a. Result: The database that you will hold historical data is created.

Step 6.2. Specify the Database to be the Historical Database

! **Important:** You must select a historical database that was created for the TADB database that you selected for your project. Do not select a historical database that was created and used for a different TADB database, even if that TADB database has abandoned it.

1 (page 838)	Enable the TADB Historical Database
2 (page 839)	Create the Historical TADB Data Source

Enable the TADB Historical Database

1. Open the TrackerCfg_UI.
2. Select the following.

1	Tools on the TrackerCfg_UI menu bar.
---	---

2	TADB on the Tools menu.
3	TADB Datasource on the TADB Extended menu.

A TADB Datasource Specification dialog box opens displaying the TADB data source specifications.

1. Do the following.

1	Check Enable Historical Database.
2	Click OK.

A Blank TADB Historical Datasource Specification dialog box opens.

Create the Historical TADB Data Source

1. Do the following.

A	Enter specifications for the Historical TADB database. Important: Make sure you enter the exact name for the database you created in the SQL Server Management Studio.
B	Click OK.

A message may open reporting that the SQL Server agent must be started.

1. Open the Microsoft Services window before you respond to the message.
2. Do the following.

A	Select SQL Server Agent (SIMPLICITY) . Note: Startup is set to Manual.
B	Click the Run button.
C	Close the Services window when the service is running.

The TADB Historical Datasource Specification dialog box re-displays.

1. Click OK.

A message box opens reporting that the selected TADB database is not historical database and asks if you want to create one.

1. Click Yes.

Another message box opens alerting you the to fully enable the Historical TADB database may take a long time.

1. Click Yes.

Result: The Historical TADB database is created.

Either a:

- Report opens reporting the errors when creating the database.
- Message opens reporting that the database was created with no errors.


Step 6.3. Start/Stop Historical Database Activity

Start Historical Database activity.
Stop Historical Database activity.

Start Historical Database activity

Check Tools>Start/Stop Historical Database activity when the project is stopped.

Result: Data starts getting stored in the Historical Database. Storage will continue as long as it is running.

 **Note:** Under normal conditions, once you start Historical Database activity, it will continue running with no interruptions.

Stop Historical Database activity

Clear Start/Stop Historical Database activity when the project is running.

Result: Data is no longer stored in the Historical Database. There will be no historical record of production that occurs while Historical Database activity is stopped.

Step 6.4. Perform TADB/Historical TADB Maintenance

Following is a summary of Historical Tracker Attribute Database (Historical TADB) and Tracker Attribute Database (TADB) maintenance features.

- TADB/Historical TADB Maintenance: Summary
- TADB/Historical TADB Maintenance: Overall Architecture
- Maintenance: TADB


- Maintenance: Historical TADB

TADB/Historical TADB Maintenance: Summary

- When items are deleted from **PRT_UI**, if archiving is not indicated, TADB causes items to be without regions; they are not deleted
- Items in the production **TADB** that are not assigned to a region, may accumulate. This behavior is by design, since items that disappear from PRT may reappear in another region. TADB items that are in no PRT region can be removed using a rule based alternative.

Note: The items are not removed from the Historical TADB; they continue to be archived.

- The Historical TADB, which has the capability to retain Tracker item data long after the items have left Tracker, also has a mechanism to remove archived item data from the Historical TADB.

 **Note:** Everything that is in TADB is in Historical TADB; Historical TADB runs about 8 seconds behind TADB.

TADB/Historical TADB Maintenance: Overall Architecture

PRT only deletes items from the TADB when the archive disposition is used.

- PRT archiving:

1. Deletes the item from PRT.
2. Sets the **Active** column for the item in TADB to 0.
3. Deletes the item from TADB.
 - The 0 value for the item's **Active** column is sent to the Historical TADB; the item, which is then ignored, remains archived.

 **Note:**

- All items are archived to the Historical TADB to prevent accidental deletion of item data. The archive disposition only indicates when they are archived, not if they are archived.
- Archiving misplaced or deleted items in TADB and archived items in Historical TADB is handled with rule based SQL Agent jobs that are configured and started by the normal CIMPLICITY project upgrade processes.
- Following are definitions for terms that help to indicate how data is maintained in PRT, TADB and Historical TADB.

Term	Definition
Remove	Generically explains when data, in the context of Archive, Delete and Purge, is no longer there, without reusing the other terms and creating circular definitions.

Archive	Delete from PRT and TADB, but not from Historical TADB. Note: Scripts, for example, can archive an item at anytime	
Delete	The definition for delete is revised slightly after the Historical TADB/TADB Maintenance enhancements have been implemented.	
	Before enhancement	Deleted data from PRT, TADB, and Historical TADB. Note: Deleting is valuable for temporary items that do not need to be kept for a long time.
	After enhancement	Delete from PRT and remove the region from the TADB item. Note: This mimics behavior when there is a region overflow.
Purge	Delete old previously archived items in Historical TADB.	

Maintenance: TADB

TADB items without regions can be archived from the TADB (to the Historical TADB), based on time constraints or item data based rules.

- The TADB Maintenance Job:
- Is a memory resident SQL Agent job that sleeps when there is nothing to do.

Note: This is similar to the replication jobs that are used to make the Historical TADB function.

- (Optionally) uses configuration values held in both of the following to determine when to archive items that no longer exist in a region to the Historical TADB.

TADBConfig table in the production TADB.

Tracker Query Engine named expressions.

- If a Historical TADB is not configured items are deleted.
- Includes the following.
- TADB Maintenance: Configuration
- TADB Maintenance: Tracker Query Named Expression
- TADB Maintenance: Behavior

TADB Maintenance: Configuration



A TADB Maintenance dialog box enables a user to change the settings for the TADB maintenance job.

4. Open the Tracker Configuration window.
5. Select Tools>TADB> TADB Maintenance on the Tracker Configuration menu bar.

Result: The TADB Maintenance dialog box opens.

Options are as follows.

Option	Description
--------	-------------

Expired days to retain	(Days) Number of days an item in the TADB with no region set may exist without modification in the Production TADB before the agent job automatically archives it. The range of valid values is as follows.	
	0	Disables archiving by time  Note: If archiving by time is disabled, only Tracker Query Engine filtering can take place.
	95,342	Maximum number of days.
	1	Minimum number of days for archiving by time to be enabled.
	Default	14
SQL batch size	Number of items that can be archived in a single operation. Values that are:	
	Larger:	Archive applicable items more efficiently.
	Smaller:	Have a smaller performance impact on other database operations.
	The range of valid values includes the following.	
	2,147,483,647	Maximum number of items.
	1,000	Minimum number of items.
	Default	1,000
Maximum idle wait time (seconds)	Number of seconds to sleep when there are no items to process.	
	 Note: <ul style="list-style-type: none"> When Tracker Query Engine expressions are not configured: It is not necessary to set this value; the maintenance script can predict how long it must wait for more items to be ready for archiving. When the Tracker Query Engine expression TADBArchiveRuntime (page 844) is defined: Setting the Maximum idle wait time value will ensure that maintenance is performed more frequently than 1 time per day. Options are as follows. 	
	0	Represents 86,400 seconds (1 day).
	86,400	Maximum number of seconds.
	120	Minimum number of seconds. (2 minutes).
	Default	0 (86,400 seconds)
Enabled	Whether or not maintenance activity can take place at the current time.	
	Checked	Enables maintenance.
	Clear	Suspends maintenance.
	Default	Enabled or 1

	<p>! Important: A stored procedure, <code>cimsp_MaintenanceState</code>, which is provided.</p> <ul style="list-style-type: none"> • Can enable, disable, or query whether or not maintenance should take place at the current time. • Takes a single optional integer parameter to indicate that maintenance activity should be enabled or disabled. • Returns a value that indicates the current status. <p>Returned values are:</p>	
	1	Enables maintenance.
	0	Suspends maintenance.
	Default	Current status Note: If no value is passed, no action is taken.

TADB Maintenance: Tracker Query Named Expression

A Tracker Query Engine named expression, `TADBArchiveRuntime`, can be defined for each item type for which a more specific rule is desired.

Example

```
Trk.COLOR='BLUE'
```

Items will be deleted by the job if both their:

- Region is blank.
- Attribute COLOR is BLUE.

Note: The **Expired days to retain** is not applied to these expressions, so it is possible to archive some items almost immediately after they leave PRT.

However, expression logic such as `TADB.modified<'Ud:-2'` can still be used to specify a retention period.

TADB Maintenance: Behavior

The TADB Maintenance enhancements have changed TADB Maintenance behavior, as follows.

Behavior before Tracker TADB Maintenance Enhancements

The following table displays how the affected Tracker features reacted, before the TADB maintenance enhancements, when the listed actions occurred.

Action	PRT Behavior	TADB Behavior	Historical TADB Behavior
PRT Region Overflows	Item was deleted.	Item's region name was emptied.	No action.
PRT Delete Item	Item was deleted.	Item was deleted.	Item was deleted.
PRT Archive Item	Item was deleted.	Item was deleted.	Item's active column was set to 0.

SQL Agent Job (did not exist)	None.	None.	None.
-------------------------------	-------	-------	-------

Behavior after Tracker TADB Maintenance Enhancements

The following table displays how the affected Tracker features react, after the Tracker TADB Maintenance enhancements are installed, when the listed actions occur.

Action	PRT Behavior	TADB Behavior	Historical TADB Behavior
PRT Region Overflows	Item is deleted.	Item's region name is emptied.	No action.
PRT Delete Item	Item is deleted.	Item's region name is emptied.	No action.
PRT Archive Item	Item is deleted.	Item is deleted.	Item's active column is set to 0.
*SQL Agent Job action	Not applicable.	Item is deleted.	Item's active column is set to 0.

*The SQL Agent job deletes expired TADB items that have no region.

Maintenance: Historical TADB

Archived Item data can be removed from the Historical TADB based on **time constraints** or **item data based rules**.

The Historical TADB Maintenance Job:

- Is a memory resident SQL Agent job that automatically sleeps when there is nothing to do.

Note: This is similar to the replication jobs that are used to make the Historical TADB function.

- Uses configuration values held by both of the following to determine when to delete archived items that no longer exist in the production TADB.
- `TADBConfig` table in the Historical TADB.
- Tracker Query Engine named expressions.
- Includes the following.
- Historical TADB Maintenance: Configuration
- Historical TADB Maintenance: Tracker Query Named Expression
- Historical TADB Maintenance: Behavior


Historical TADB Maintenance: Configuration



A Historical TADB Maintenance dialog box enables a user to change the settings for the Historical TADB maintenance job.

6. Open the Tracker Configuration window.
7. Select Tools>TADB>Historical TADB Maintenance on the Tracker Configuration menu bar.

The Historical TADB Maintenance dialog box opens.

Options are as follows.

Option	Description	
Expired days to retain	<p>(Days) Number of days an archived item in the Historical TADB that no longer exists in the Production TADB may exist without modification in the Historical TADB before the agent job automatically purges it. Note: This also designates the maximum time noncurrent historical rows are allowed to exist:</p> <ul style="list-style-type: none"> • For active items in the following tables: • TItems • TTrkAttributes • TRelatedItems • TGroupAttribs^<item type>^<attribute group> • in the following tables. • TCritSets • TTrkGroups • (New) TGroupAttribs^<item type>^<attribute group>_lookup • TTrkRegions • TRanges • TRangeRanges <p>The range of valid values is as follows.</p>	
	0	<p>Disables purging by time.</p> <p> Important:</p> <ul style="list-style-type: none"> • If purging by time is disabled, only Tracker Query Engine filtering can take place and only items will be deleted. • If archiving by time is enabled, items will be purged by time even if Tracker Query Engine filtering is enabled.
	95,342	Maximum number of days.
	1	Minimum number of days for purging by time to be enabled.
	Default	0
SQL batch size	Number of items that can be archived in a single operation. Values that are:	
	Larger:	Archive applicable items more efficiently.
	Smaller:	Have a smaller performance impact on other database operations.
	The range of valid values includes the following.	
	2,147,483,647	Maximum number of items.
	10,000	Minimum number of items.
	Default	10,000
Maximum idle wait time (seconds)	(Seconds) Number of seconds to sleep when there are no items to process.	

 Note: <ul style="list-style-type: none"> When Tracker Query Engine expressions are not configured: <p>It is not necessary to set this value; the maintenance script can predict how long it must wait for more items to be ready for deletion.</p> <ul style="list-style-type: none"> When the Tracker Query Engine expression <code>TADBPurgeHistorical (page 847)</code> is defined: <p>Setting the Maximum idle wait time value will ensure that maintenance is performed more frequently than 1 time per day. The range of valid values is as follows</p>	
0	Represents 86,400 seconds (1 day)
86,400	Maximum number of seconds.
120	Minimum number of seconds. (2 minutes)
Default	0 (86,400 seconds)
Enabled	Whether or not maintenance activity can take place at the current time.
Checked	Enables maintenance.
Clear	Suspends maintenance.
Default	Disabled or 0
 Important: A stored procedure, <code>cimsp_MaintenanceState</code> , which is provided. <ul style="list-style-type: none"> Can enable, disable, or query whether or not maintenance should take place at the current time. Takes a single optional integer parameter to indicate that maintenance activity should be enabled or disabled. Returns a value that indicates the current status. <p>Returned values are:</p>	
1	Enables maintenance.
0	Suspends maintenance.
Default	Current status Note: If no value is passed, no action is taken.

Historical TADB Maintenance: Tracker Query Named Expression

A Tracker Query Engine named expression, `TADBPurgeHistorical`, may be defined for each item type for which a more specific rule is desired.

Example

```
Trk.COLOR='BLUE'
```

Archived items will be deleted by the job if their attribute `COLOR` is `BLUE`.

Note: The **Expired days to retain** is not applied to these expressions, so it is possible to delete some archived items almost immediately after they leave the production TADB.

However, expression logic such as `TADB.modified<'Ud:-2'` can still be used to specify a retention period.

Historical TADB Maintenance: Behavior

The ability to purge archived data from Historical TADB did not exist before. The Historical TADB Maintenance job provides that ability; it can purge data following rule based criteria.

3.2. TADB Configuration Import/Export Utility

3.2. TADB Configuration Import/Export Utility

TADB Configuration Import/Export Utility provides a way to export the SQL database configuration associated with your Order Execution Management project to an XML file.

The exported file can either be imported into a SQL database directly, or selected when you [create a new TADB database \(page 802\)](#) through the POMS wizard.

Step 1 (page 848)	Export a TADB configuration file.
Step 2 (page 848)	Import a TADB configuration file.

Step 1. Export TADB Configuration

1. Check this option.
2. Enter the User name and Password previously assigned to the selected database.

Step 2. Import TADB Configuration

1. Check this option.
2. Enter the User name and Password previously assigned to the selected database.

4. TADB/PRT Backing File Synchronization and Recovery

4. TADB/PRT Backing File Synchronization and Recovery

- Logical discrepancies between Tracker and TADB.
- Synchronization and recovery overview.
- Differential report timestamps.
- Synchronization report and implementation steps.
- Backing file recovery when there are decreases in PRT configuration components.

Logical Discrepancies between Tracker and TADB

Whenever an item is added to TADB or modified in TADB (which almost always is when an out of sync issue will occur):

1. SQL Server returns all the data being stored for the saved item.
2. PRT compares the values returned from the TADB to those in Tracker.
3. PRT compares every field returned to its internal value and sends a descriptive error message to the same synchronization error handler that currently logs a message and raises an out of sync alarm.

Result: If errors are found, they are reported using the same, but greatly extended TADB error message.

Synchronization and recovery overview

The PRT Collector manages discrepancies or corruption in the PRT backing files as follows.

- Synchronize data

When the project is not running a user can have the PRT Collector synchronize PRT and TADB.

During synchronization PRT data overwrites TADB data in most cases.

Note: When items in TADB that are assigned to regions in TADB, but are not in PRT, by default they will be archived unless archiving is not indicated; in that case the item will be made regionless.

- Recover data

When a project starts PRT Collector validates the data between PRT backing files and TADB.

If PRT backing files are corrupt the PRT Collector will:

4. Delete the corrupt backing files.
5. Create new blank backing files.
6. Query the TADB database for the data needed to recover the backing files.
7. Recover the data from the database.

Differential Report Timestamps

The PRT_TADB_Diff report displays the last modified timestamps in both TADB and PRT for the representation of an item.

- Knowing the last modified times can help determine which version of data is true, if there are differences between TADB and PRT for an item.
- Because the timestamps will most often be different, if they are the only difference between PRT and TADB for an item, that item will not be listed in the report.

Synchronization report and implementation steps

Step 1 (page 852)	Create/Open PRT vs. TADB Validation Reports
Step 2 (page 857)	Review and Copy Report Data
Step 3 (page 858)	Synchronize PRT Backing Files and TADB

Backing File Recovery when there are Decreased in PRT Configuration Components

When the PRT configuration changes after items have been logged to the database involve reductions, e.g. fewer regions, fewer locations, fewer items in a location, recovery attempts to:

- Maintain the item sequence.
- Maintain item associations.
- When warranted, move items to the next location or region.
- Report all changes and issues in the Status Log.

The following examples portray two possible situations.

Example 1

REGION1's capacity initially is 200.

The capacity is reduced to 150.

Recovering items from the database starts at Location 1.

Items are recovered from 1-150.

At 150, the revised capacity is full; there is no more room for items.

Item 151 is not recovered.

An initial warning and failure are logged to the status log that:

- Item count %d <200> exceeds Item capacity %d<150> for region <REGION1>.
- Location %d<151> for Item %s<widgets> exceeds region <REGION1> locations %d<150>.

Reports about all items over 151 are also logged to the Status Log.

Note: The recovery maintains the initial integrity of items/location. If the location allows 2 items and only 1 item is in the location at the time of recovery, the location will continue to have only 1 item.

Example 2

The location capacity in REGION1 is initially is 3 items.

The capacity is reduced to 2 items.

If the database contains 3 items in a location recovery will attempt to fulfill three requirements.

- Maintain the item sequence.
- Maintain item associations.
- Reduce the number of items in the location to 2 or 1.

The recovery starts with the first item in the location.

The following table displays how database items in locations that previously allowed 3 items are allocated during recovery.


Location	Database Items	Recovered Items
1	1A, 1B, 1C	1A, 1B
2	2A, 2B, 2C	1C
3	3A, 3B, 3C	2A, 2B
4	4A, 4B	2C
5	5A, 5B, 5C	3A, 3B
6	6A, 6B	3C

- For each instance that an item is re-assigned to the next location a report is logged in the Status Log.

Adjusted %s<1C> location from %d<1> to %d<2> during recovery. Updating TADB to match.

- For each instance that there is an attempt to re-assign an item and the region/location capacity is exceeded, a report is logged in the Status Log.

Adjusted location %d<7> from %d<4> for %s<4A> exceeds %s<REGION1> %d<6> locations.

 **Note:** If an unforeseen error occurs while recovering an item from the TADB, the following message is logged to the status log.

Failed recovery %s to %s at adjusted location %d, %d orig.

Where the message components are:

Failure to recover <item> to <region> at adjusted location <attempted location>, <database location>.

Step 1. Create/Open PRT vs. TADB Validation Reports

Step 1. Create/Open PRT vs. TADB Validation Reports

New reports can be created and existing reports opened to evaluate details about regions or items.

1.1 (page 852)	Validation Reports: Selected Regions and/or Locationless Items
1.2 (page 856)	Validation Report: Selected Items

1.1. Validation Reports: Regions and/or Locationless Items

1.1. Validation Reports: Selected Regions and/or Locationless Items

A PRT vs. TADB Validation Report window is available to evaluate selected regions or items that are not assigned regions in the TADB (locationless items). The window for these reports is opened through the Workbench.

- Open the PRT vs. TADB Validation Report window
- Create/Open a Report for Selected Regions or Locationless Items

Open the PRT vs. TADB Validation Report window

1. Make sure the project is not running.

Note: A report can be generated if the project is running; however you may see illegitimate diffs that muddy the report because items are moving; in addition the extra activity could slow down production.

2. Select **Project>Tracker Configuration>PRT_TADB_DIFF** in the Workbench left pane.
3. Select **PRT_TADB_Diff** in the Workbench right pane.
4. Do one of the following.

A	Click Edit>Properties on the Workbench menu bar.
B	Click the Properties button on the Workbench toolbar.
C	In the Workbench left pane:

	Either	Or
	Double-click PRT_TADB_Diff .	a. Right-click PRT_TADB_DIFF . b. Select Properties on the Popup menu.
D	In the Workbench right pane:	
	Either	Or
	Double-click PRT_TADB_Diff .	a. Right-click PRT_TADB_Diff . b. Select Properties on the Popup menu.
E	Press Alt+Enter on the keyboard.	

5. Right-click **PRT_TADB_DIFF**.
6. Select Properties on the Popup menu.
7. Right-click **PRT_TADB_Diff**.
8. Select Properties on the Popup menu.

1.1.1. Validation Report: Selected Regions

Validation reports can be opened and created through the Workbench for selected regions.

- Create a Selected Region Report
- Open an Existing Selected Regions Report

Create a Selected Region Report

1 <i>(page 853)</i>	Create a Selected Region Report File
2 <i>(page 854)</i>	Select Desired Regions

1. Create a Selected Region Report File

1. Click View on the PRT vs. TADB Validation Report menu bar to select the generated report view.

Options are:

	Report view	Report displays
1	Expanded View	All details, including data that is identical in the PRT backing files and TADB.
2	Compact View	(Default) Only the differences between PRT backing file and TADB data.

1. Do any of the following.

- Click the New button on the PRT vs. TADB Validation window toolbar.
- Click File>New on the PRT vs. TADB Validation window menu bar.
- Press Ctrl+N on the keyboard.

A Save As dialog box opens when you use any method.

1. Do the following.

1	Select a directory for the Diff report file.
2	Enter a filename for the report file in the File name field. Note: The file will be an XML file.
3	Click Save.

Result: A Select Desired Regions browser opens.

1. Select Desired Regions


Do the following.

A	Select one or more regions. Tip: The Browser recognizes the standard selection keys.	
	Key	Selects
	Ctrl	Regions not in sequence.
	Shift	Regions in sequence.
	Default	All regions are selected.
B	Click OK.	

Result: PRT Collector:

Compares the PRT backing files with TADB data for the selected regions.

Reports the differences in the compact or expanded view, whichever was selected.

 **Note:** You can change the [view \(page 853\)](#) at any time; all of the information was collected so if you selected the Compact view it is available for you to use the Expanded view.

Compact View: Displays only the differences.

Expanded View: Displays all details in PRT backing files and TADB.


Open an Existing Selected Regions or Selected Item Report

1. Do one of the following
 - Click the Open button in the PRT vs. TADB Validation window toolbar.
 - Click File>Open on the PRT vs. TADB Validation window menu bar.
 - Press Ctrl+O on the keyboard.

The Open dialog box opens in the last location in which you saved or opened a report file using any of the methods.

2. Select the report file to open.
3. Click Open.

Result: The selected report opens.

 **Note:** If the window appears to be blank, simply expand it slightly; the report details will display.


Open an Existing Selected Regions or Selected Item Report

4. Do one of the following
 - Click the Open button in the PRT vs. TADB Validation window toolbar.
 - Click File>Open on the PRT vs. TADB Validation window menu bar.
 - Press Ctrl+O on the keyboard.

The Open dialog box opens in the last location in which you saved or opened a report file using any of the methods.

5. Select the report file to open.
6. Click Open.

The selected report opens.

 **Note:** If the window appears to be blank, simply expand it slightly; the report details will display.

1.1.2. Validation Report: Locationless Items

1. Click File>New Locationless on the PRT_TADB Validation window menu bar.

A Save As dialog box opens.

2. Do the following.

1	Select a directory for the Locationless Item report file.
2	Enter a filename for the report file in the File name field. Note: The file will be an XML file.
3	Click Save.

Result: The PRT vs. TADB Validation window opens with the Locationless Item report.

- The report, which is in Expanded view by default, provides details about the found items. However, It does not compare these details with items in PRT that might have the same item IDs.
- The Expanded view provides database headers for the columns.

Open an Existing Locationless Item Report

3. Do one of the following


- Click the Open button on the PRT vs. TADB Validation window toolbar.
- Click File>Open on the PRT vs. TADB Validation window menu bar.
- Press Ctrl+O on the keyboard.

The Open dialog box opens in the last location in which you saved or opened a report file using any of the methods.

4. Select the report file to open.

5. Click Open.

The selected report opens.

 **Note:** If the window appears to be blank, simply expand it slightly; the report details will display.

1.2: Validation Report: Selected Items

1. Start the project.

2. Open the PRT_UI

3. Do the following.

A	Select the region in which the item to evaluate is located.
B	Right-click the item.
C	Select Diff with TADB on the Popup menu.


Result:

- No differences are found.

A No Differences report opens listing the item's details in PRT and TADB.

- Differences are found.

A Differences Found report opens listing the item's details in PRT and TADB and highlighting the differences.

 **Note:** If you select more than one item and then select Diff with TADB on the Popup menu, the first item selected will be evaluated.

Step 2. Review and Copy Report Data

- Find data in a report.
- Copy data in the report.

Find data in a report

1. Click Edit>Find on the PRT vs TADB Validation window menu bar.

A Find dialog box opens.

2. Enter the find criteria.

Option	Description	
Find	(Required) String that is located in one or more single cells in the table.	
Match whole word only	The found string is exactly as it is entered. Example 365 is entered in the Find field.	
	Checked	365 is not found in a customer order CO788736522.
	Clear	365 is found in a customer order CO788736522.
Match case	Only the string that is in the case entered in the Find field will be selected. Example co788736522 is entered in the Find field.	
	Checked	The actual order CO788736522 is not found.
	Clear	The actual order CO788736522 is found.
Highlight all matches	Highlights all matches in the report. Example 365 is entered in the Find field.	
	Checked	Highlights all matches.
	Clear	Highlights the next or previous selected match.
	Note: Matches are determined by which check boxes are checked or clear.	
Buttons	Next	TADB goes to the next match in the Diff report.

	Previous	TADB goes to the previous match in the Diff report.
Note: Matches are determined by which check boxes are checked or clear.		

Copy Data in the Report

3. Do any of the following to select data in the report.

- Select the Entire Report
- Select a Section in the Report

Select the Entire Report

Click Edit>Select All on the PRT vs. TADB Validation Report window menu bar.

The entire report is selected.

Select a Section in the Report

- a. Hold the right-mouse button down.
- b. Drag the mouse over the section of the report you want to select.

The sections the mouse was dragged over are selected.

4. Do one of the following.

- Click the Copy button on the PRT vs. TADB Validation Report window toolbar.
- Click Edit>Copy on the PRT vs. TADB Validation Report window menu bar.
- Press Ctrl+C on the keyboard

PRT vs. TADB validation copies your selection.

The copied data can be pasted into any text application, e.g. Notepad.

Step 3. Synchronize PRT Backing Files and TADB

1. Create a new report, selecting the regions to evaluate.
2. Click Edit>Synchronize on the PRT vs. TADB Validation window menu bar.
3. Create a new report for the selected regions to confirm that the data is synchronized.

PRT and TADB data is synchronized.

TADB Identifier Restrictions

The following identifiers are allowed in any order with restrictions.

A-Z, a-z, 0-9, _ and <space>

Identifier exceptions are:

- An identifier may not begin or end with a space.

Example

Valid	A<space>A
Not Valid	<space>A and A<space>

- A space cannot be followed by another space in identifier names.

Example

Valid	A<space>A
Not Valid	A<space><space>A

- An identifier may not be all numbers.

Example

"4x"

Valid	"4<space>_"
Not Valid	"4"

- Reserved identifiers are

Identifier	Reserved by the Query Engine syntax to reference:
"Trk"	Tracker Standard, Extended, and System attributes.
"TADB"	TADB System attributes (e.g. id, refid).

7. Range Source Architecture

About Range Source Architecture

Range Source Architecture (RSA) enhances the traditional RCO concept of a the Tracker source (source region).

In RSA a routing logic module, when triggered, can now:

- Execute RSA logic to dynamically choose what sources it will process.
- Weight and eliminate items at those sources according to various rules in order to process them in the most efficient or logical sequence.

rect 79, 12, 337, 143 [About External Scheduling \(page 773\)](#)

rect -3, 140, 192, 299 [About XMLT \(page 773\)](#)

rect 1, 314, 183, 395 [About DIR_WATCHER.bcl \(page 796\)](#)

rect 208, 276, 562, 486 [About the Product Order Management System \(POMS\) \(page 799\)](#)

rect 0, 400, 187, 493 [About CimView Order Entry \(page 810\)](#)

rect 293, 161, 410, 258 [About the Tracker Attribute Database \(page 817\)](#)

rect 562, 360, 639, 472 [Range Source Architecture Overview \(page 860\)](#)

rect 421, 148, 521, 238 [About the Tracker Query Engine \(page 893\)](#)

rect 224, 490, 356, 551 [About Order Execution Management Broadcast \(page 934\)](#)

Range Source Architecture Overview

Range Source Architecture Overview

Range Source Architecture (RSA) complements and works alongside the traditional RCO decision-elimination logic.

- Traditional Routing Logic.
- Range Source Architecture.

Traditional Routing Logic

Items are evaluated when they are located in a first or last location in a source region.

The Routing Logic, follows a definite sequence.

Example

The assembly floor of a computer manufacturer has three regions.

Computers travel from the first location in one region to the last location in the next.

They are assembled sequentially in each region as they travel from last to first place.

Range Source Architecture

Range Source Architecture enables decisions to be made for items in any location in as many regions as you want.

This architecture is designed for items that do not require linear movement such as orders or items stored in an automated system retrieval system (ASRS) in which items can be randomly selected. The actual location is immaterial.


The initially intended purpose of a range is to evaluate, select, sequence, substitute, hold or eliminate items to be produced by evaluating the purchase orders.

Since orders can be grouped into pre-defined categories (virtual regions), not physical locations, the criteria for selecting the range to be evaluated is limited only by what you specify. There are no outside constraints.

Ranges simply take advantage of this non-restrictive capability.

A range can be:

- A single region.
- Several regions from one or more projects.
- A subset of sequential locations from one or more regions, in one or more projects.
- A query that returns, on-the-fly, any combination of regions and/or region locations.

 **Note:** The TBNG_POMS template enables you to create a project with pre-defined regions that can be used exactly how they are configured or modified to your system's needs.

Example

The computer factory evaluates purchase orders to manage if, when and in what sequence manufacturing will fulfill the order.

- The range in which evaluation will take place includes all location in the \$OM_MASTR_BLND region.
- \$OM_MASTR_BLND contains orders that were accepted for production.
- The goal is to select the next order to move to the \$OM_PLANT_BLD region.

This can be accomplished in the range as RSA:

1. Evaluates the orders using a combination of business rules.
2. Eliminates order that fail.
3. Selects the single remaining order as the next item to process.
4. Moves the selected order to \$OM_PLANT_BLD.
 - The order can be any one in the range. Its location is irrelevant.

Note: The colors of the orders in this graphic represent the color of the computer that is being ordered.

L1, L2...L20 = Locations

P01,P02...P20 = Purchase orders

Range Source Architecture (RSA) Modules

- RSA module definition.
- Differences between RSA and base RCO function blocks.
- Range Source manipulation.
- Configure RSA Modules.
- RSA Applications.

RSA module definition

RSA function blocks eliminate or weight items instead of decisions. Items that remain after an elimination are available to the function blocks that follow. An RSA module:

- Is a series of function blocks that starts with a range-defining block and usually ends with an item selection.
- May be part of a routing logic or output logic module.

The RSA module, in effect, is pre-processing logic that:

- Selects an item and
- Sets or stores data about the selected item.

This data can then be used by base RCO function blocks to make routing decisions and generate output. If your RSA module starts in the routing logic, the range source list (or selected item) is available to the output logic of any remaining decision, where you can continue your RSA module.

Differences between RSA and base RCO function blocks.

RSA function blocks are different from Routing and other base RCO function blocks in the following ways:

	Traditional RCO	Range Source Architecture
Item Source	Source Region	Range Source (non persistent)
Eliminates	Decisions	Items


Item Destination	Destination Region	<p>If the RSA module is in an RLM:</p> <ul style="list-style-type: none"> • Item(s) not eliminated from the range source are available for further processing to subsequent function blocks within the same routing or output logic module. These subsequent blocks may execute traditional RCO elimination logic to choose a Decision. • Changes made to an item's standard attributes, extended attributes, or to the TADB persist and remain available when the item is processed by other routing or output logic modules. • A selected item is available to, and may be routed by, the remaining Decision's Output Logic Module.
------------------	--------------------	--


Range Source Manipulation

In traditional RCO, one or more source regions may be routed to one or more destination regions. But the order in which items are processed by a routing logic module is primarily determined by the order in which items are queued. Depending on how decisions are eliminated, one or another item at the head of a decision's source region will be processed next. If necessary, an RCO may even reject an item by routing it to a Detainment region. But the item, once it has left its source, must go somewhere.

However, there are situations where it is advantageous to apply logic before items leave their source regions. In those situations, RSA provides supplemental logic to:

Set the range of sources by range, criteria or solve (query).
Select, or sequence, the next item to process through elimination and weighting.
Assign sequence numbers, e.g. rotation numbers, to items in the range source.
Substitute the order associated with an item in the range source for the order of another item at a particular region location.
Apply a hold to items that take effect when they enter a range.

 **Note:** A routing logic module (RLM) that uses RSA function blocks does not need to determine any routing decisions. The RLM can be used merely to modify attributes of items in the range source. For example, a range of items can be placed on hold or assigned rotation numbers. The modification persists after the control cycle executing the function blocks is completed without necessarily moving any items. In this case there is no decision elimination. Therefore, make sure to include only one decision in the RCO's Decisions folder, or configure a Multiple output module.

 **Important:** RSA function blocks with Attribute Name and Attribute Value parameters support standard and extended attributes.

TADB attributes may be accessed using the:

- RSA function blocks that support named solves (queries) or
- RSA function blocks that support solve (query) expressions (using the reserved name SOLVE instead of the attribute name) or

- [TADB function blocks \(page 584\)](#) to access TADB attributes directly.

Configure a Range Source Architecture Module

Configure a Range Source Architecture Module

When you create a range source output, it is recommended that you divide your criteria into two logic modules.

Add blocks to:

Step 1 (page 864)	Define the Source Range
Step 2 (page 872)	Add a Set Block Weights Block
Step 3 (page 872)	Add evaluation blocks to the RSA module.
Step 4 (page 873)	Add historical evaluation to the RSA module.
Step 5 (page 886)	Apply Weights to the Evaluation Blocks
Step 6 (page 887)	Add a Function Block to Break Ties
Step 7 (page 888)	Use Range Source Applications

Step 1. Define the Source Range

Step 1. Define the Source Range

The first block, when you create a Routing Logic Module (RLM) must define the source range in which the item will be evaluated.

Example

A computer factory creates a Routing Logic Module to select purchase orders for computers that will be classed as CLASS1 computers.

Where and/or what orders to be evaluated needs to be defined.

The following options define a basic range source.

Option 1.1 (page 865)	Use a range as a source.
Option 1.2 (page 869)	Use a spread range as a source.
Option 1.3 (page 869)	Use a Query as a range source.

Criteria sets include the range in the definition, you can also use it as the first block in the RLM.

Option 1.4 (page 872)	Use a criteria set as a range source.
---------------------------------------	---------------------------------------

Option 1.1. Use a Range as a Source

Option 1.1. Use a Range as a Source

Ranges include:

- One or more locations in
- One or more regions in
- Any running project to which you have access.

You can configure ranges through any function block that contains a range name. Once you have defined it, the range is available for any other range source architecture that you configure.

Steps to select a range as a source include.

1 (page 866)	Open the P1 dialog box.
2 (page 866)	Use P1 dialog box features.
3 (page 867)	(Optional) Use Range Browser features.
4 (page 867)	(If an existing range is not acceptable) Configure the range.

1. Open the P1 dialog box.

1. (Recommended) Create a new Routing Logic Module.
2. Double-click the Range Name or Hold Active Range parameter in function blocks including:

Set Source Range
Begin Spread Range Block Module
Rotation Pull Ahead
Apply Hold to Range.
Apply Hold To Item

A P1 dialog box opens.

The P1 dialog box provides you with the Browse, add and editing options that other RCO function blocks offer.

[Up \(page 865\)](#)

1. Use P1 dialog box features.

Do any of the following through the P1 dialog box.

Option	P1 Dialog Box	
Range Name	Identifies the range to: <ul style="list-style-type: none"> • Use in the block. • Be edited if Edit is selected on the popup menu. 	
Browse buttons	Opens the Range Browser	
Popup Menu button	Displays the popup menu	
	Selections are:	
	New	Opens a new Range Viewer dialog box.
	Edit	Opens the Range Viewer for a range that is entered in the Range Name field.
	Browse	Opens the Range Browser.
	Recent ranges	Uses the range that is selected in the list.
OK	Closes the P1 dialog box; displays the range as the parameter value.	

The Range Browser lists all of the ranges that have been configured for the project.

[Up \(page 865\)](#)

1. (Optional) Use the Range Browser.

If you opened the Range Browser you can use the buttons to do any of the following.

Button	Description
OK	Closes the Range Browser and enters the selected range into the Range Name field in the P1 dialog box.
Cancel	Closes the Range Browser without selecting a range.
Add	Opens a new Range Viewer dialog box.
Edit	Opens the Range Viewer for the selected range.
Delete	Deletes a selected range.

The Range Viewer dialog box is either blank or contains regions and locations that are included in a selected range.

[Up \(page 865\)](#)

1. Configure the range.

Depending on whether you are adding or editing a range, the Range Viewer dialog box is blank or contains regions and locations that are included in a selected range.

1. Do the following to configure the range.

1. Add to or edit the list as follows.

1. Click Add.

A blank Range Editor dialog box opens.

1. Select a region in the list and click Edit.

The Range Editor dialog box opens with the current configuration.

1. Enter or edit the ranges as follows.

Field	Description
Range name	(Read only) Name entered in the Range Viewer Range name field.

Region	Region to be included in the range.
Project	Name of a running project from which a region is selected.
Region start (page 868)	Region location or location code for the first location to include in the range.
Region end (page 868)	Region location or location code for the last location to include in the range.

1. Click OK.

The Range Viewer displays the region and selected location codes in the Region list.

Example

The example computer factory:

- Wants to be able to select the next order that should be sent to the \$OM_PLANT_BLND region.
- Creates a range that includes all locations in the \$OM_MASTR_BLND region.


1. Continue to add, edit or delete regions until the range has all of its regions/locations.
2. Click OK.

The P1 dialog box opens with the selected range in the Range Name field.

Click OK.

The range displays as the value for the Range Name parameter.

Region Location Codes

 **Note:** The Range Source Set Point With Attribute or Range Source Set Variable With Attribute function block can be used to store the region and region location of the selected item to virtual points, or as RCO, variables that can be referenced by the output logic controlling how an item moves.

Ranges can be made up of multiple regions or specific locations within those regions. When a dialog box field supports entry of a subset of locations within a range, use the following codes to specify the locations you want:

Location Code	Returns Items From
>0	Location that corresponds to the entered value.
=-1	LAST location in the region.
=-2	FIRST location in the region.

=-3	ALL locations in the region.		
	StartLocation	EndLocation	Returns Items From
=0	=-3	=-3	All locations in the region.
	=-2	=-2	FIRST location in the region.
	=-1	=-1	LAST location in the region.
	=-1	>0	Currently undefined. This will return an empty set.
	>0	=-1	A StartLocation that is >0 to the end of the region.
	>0	<n	Between StartLocation and EndLocation. Note: StartLocation should be less than or equal to the EndLocation.
	=1	=-1	All locations in the region.

Option 1.2. Use a Spread Range Source

A range can include multiple passes.

! **Important:** A spread range source includes either a range or query as the range source in its definition. Therefore, you must have at least one range source defined before you create a criteria set.

The blocks are:

Begin Spread Range Block Module marks the start of a multiple-pass Spread Range Block Module in the routing or output logic module.
End Spread Range Block Module marks the end of a multiple-pass Spread Range Block Module that starts with a Begin Spread Range Block Module.

If no items or no unweighted items remain after an evaluation, these blocks can expand the range and provide multiple passes through the module.

When you add these blocks, the:

1. Number of items included in the evaluation set is incremented
2. Expanded set is re-evaluated by a second pass by the Spread Range Block Module logic.
3. Evaluation repeats multiple times, up to the number of passes you specify, until at least one unweighted item remains.

Do not put a count block inside the spread range logic.

Keep logically separate Record OLM vs. Eliminate RLM

Option 1.3. Use a Query as a Range Source

A [query \(page 893\)](#) can define a range that is included in any criteria you want. The criteria does not have to be locations. It can, for example, be a list of attributes.

Example

- A range can be orders for computers with selected top-of-the-line features, that are listed in the TADB database as attributes for a Bill of Materials.
- These orders can be identified through a query for those attributes.
- The query that includes the range of features can be used as the range source.

Steps to select a range as a source include.

1 (page 870)	Open the P1 dialog box.
2 (page 870)	Use P1 dialog box features.
3 (page 871)	(Optional) Use Expression Browser features.
4 (page 871)	(If an existing query range is not acceptable) Configure the named query.

1. Open the P1 dialog box.

Double-click Solve Name in the Set Source Solve function block.

A P1 dialog box opens.

The P1 dialog box provides you with the Browse, add and editing options that other RCO function blocks offer.

[Up \(page 870\)](#)

1. Use P1 dialog box features.

Do any of the following through the P1 dialog box.

Option	P1 Dialog Box
Solve Name	Identifies the named query to: <ul style="list-style-type: none"> • Use in the block. • Be edited if Edit is selected on the popup menu.
Browse buttons	Opens the Expression Browser

Popup Menu button	Displays the popup menu	
	Selections are:	
	New	Opens a New Expression dialog box. 1. Enter a name for the new query. 2. Click OK. Opens the Expression Editor.
	Edit	Opens the Expression Editor for a query that is entered in the Solve Name field.
	Browse	Opens the Expression Browser.
	Recent named queries	Uses the named query that is selected in the list.
OK	Closes the P1 dialog box; displays the named query as the parameter value.	

[Up \(page 870\)](#)

1. Use Expression Browser features.

If you opened the Expression Browser you can use the same steps to [create or edit \(page 894\)](#) a query as you would for any query.

[Up \(page 870\)](#)

1. Configure the Query range.
 1. Use the Expression Editor syntax to create or edit an expression that will determine what items will be included in the range.

Example

- The computer factory has a class of computers called CLASS 1.
- The computers have a 19" monitor, ergonomic keyboard and are not blue in color.
- A CLASS1 query will select all of the purchase orders that list these attributes.
- The selected orders will be the orders in the query range.

1. Apply the query and close the Expression Editor when the expression is ready.

The Expression Browser opens.

1. Select the Query.
2. Click OK.

The P1 dialog box opens with the named query in the **Solve Name** field.

Click OK.

The named query displays as the value for the Range Name parameter.

Option 1.4. Use a Criteria Set as a Range Source

1. Enter a name for the new query.
2. Click OK.
Opens the Expression Editor.
3. Only the criteria range and, if specified, the location pass through the filter.
4. Of the Items the pass through the first filter, only items that are included in the named hold remain.
Hold blocks include:

Step 2. Add a Set Block Weights Block

A Set Block Weights function block:

- Must always be second in an RSA module.
- Assigns a priority to each of the RSA function blocks that will be added to evaluate items.

Most likely you will come back and apply the weights after you add the evaluation blocks.


Example

Note: The computer factory Tracker engineers have

- Selected MASTRBLEND to use as range for the CLASS1 module.
- Created a Set Blocks Weight block named CLASS!Weights.

However they have not yet entered values.

Once RSA blocks are added to evaluate attributes such as color and monitor size the weights will be entered in the CLASS1Weights function block.

 **Note:** This block can be added after you have added the remaining blocks. However, make sure you move it up to second place.

Step 3. Add Current Evaluation Blocks to the RSA Module

Function blocks after a range selection and Set Block Weights function block can establish the criteria that will ultimately pass only successful items.

These blocks will:

- Eliminate failing items from the range source, or
- Apply the weight that is assigned by the Set Block Weights function block to failing items.

Blocks based solely on current data include:

Eliminate By Weight.
Eliminate Attribute Mismatch
Eliminate Attribute Value Mismatch.
Range Source Eliminate By Solve
Eliminate Capability Mismatch

Example

The computer factory has three part numbers for processors in the [TADB \(page 817\)](#) : Pro111, Pro112 and Pro113.

- Orders that specify Pro113 have priority over the other orders.
- The first block added to the CLASS1 RLM is Range Source Eliminate by Solve.
- This block will assign a weight to all orders that do not specify Pro113.

The Tracker engineers:

1. Select the Range Source Eliminate by Solve block in the Range Source list.
2. Open the P1 dialog box for the Solve ID/Expression.
3. Select a query that identifies orders with PRO113.
4. Close the P1 dialog box.

The named query displays as the value for the Solve ID/Expression parameter.

5. Close the Range Source Eliminate by Solve function block.

The block is the first evaluation block in the RLM list and will provide the first criteria for evaluation.

Step 4. Add Historical Evaluation to the RSA Module

Step 4. Add Historical Evaluation to the RSA Module

Historical evaluation compares current data in the context of a historical record of selections.

The exact type of comparison/calculation/evaluation depends on the block that is being used.

Each historical evaluation requires two blocks.

They are:

Historical Evaluation Block	Record Block
Eliminate Sources Based on Attribute Spacing	Record Attribute Spacing.
Eliminate Sources Based on Rule	Record Attribute Rule.
Eliminate Source By Percentages	Record Attribute Percentages.
Eliminate Source By Ratio	Record Attribute Ratios.
Eliminate Source By Pattern	Record Attribute Pattern.

Most frequently the:

- Historical evaluation blocks are added to a Routing Logic Module.
- Record blocks are added to an Output Logic Module.

Examples

The example computer factory adds three historical evaluations to the Routing Logic Moduls and corresponding record blocks to an Output Logic Module.

Example 4.1 (page 874)	Define an Eliminate Sources Based on Rule evaluation.
Example 4.2 (page 878)	Define an Eliminate Source By Percentages evaluation.
Example 4.3 (page 882)	Define an Eliminate Source By Ratio evaluation.

Example 4.1. Define an Eliminate Sources Based on Rule Evaluation.

Example 4.1. Define an Eliminate Sources Based on Rule Evaluation.

The computer factory has three values for the COLOR attribute:

- BLUE
- BLACK

- BEIGE

Comparison is case sensitive:

- Build a maximum of 3 BLACK computers consecutively.
- Space BEIGE computers so at least a minimum of two other colors are built between each BEIGE computer.

Note: The Spacing rule does not say that there should be 1 beige computer out of every three. Results from other rules (and what items are in the range) may place 5 computers with other colors after a beige computer. This is perfectly acceptable.

Rule Type	Attribute Name	Attribute Value	Rule
G	COLOR	BLACK	3
S	COLOR	BEIGE	2

Configuring this rule is done in three segments.

Example 4.1.1 (page 875)	Create an Eliminate Sources Based on Rule function block.
Example 4.1.2 (page 877)	Enter values for the rule definition arrays.
Example 4.1.3 (page 878)	Add a Record Attribute Rule function block.

Example 4.1.1 Create an Eliminate Sources Based on Rule Function Block

1. Open the Range Source function block Eliminate Sources Based on Rule.
2. Name the block COLORRule.
3. Configure the first array point as follows.
 - a. Opens the P1 dialog box for Rule Type Array.
 - b. Creates a new virtual, text array point.

Note: The array point will identify the rule types for the evaluation.

- a. Enters 2 in the Elements on the Point Properties dialog box General tab.

Note: The 2 elements will hold the two rule types, Grouping and Spacing.

- a. Select Saved in the in the **Initialization** field on the Virtual tab in the Point Properties dialog box.

Note: The value that is set in the array will be saved even if the project stops.

4. Configure the next array parameters that define the evaluation rules.

Note: Each element in the array holds a value that refers back to and defines the corresponding element in the Rule Type array.

Parameter	Point Type	Elements	Point Name
Attribute Name Array	Text	2	
Attribute Value Array	Text	2	
Rule Array	Integer	2	
Count Array	Integer	2	
Upper Release % Target	-	-	
Lower Release % Target	-	-	-
Breakable Array	-	-	-

The % and Breakable parameters will not be used in this block.

5. Configure the two parameters that will provide the count data.

Parameter	Point Type	Elements	Point Name
Attribute Release Count Array	Integer	2	
Total Release Count Point	Integer	1	

A brief description of these two parameters in this example is as follows.

- a. Attribute Release Count Array will reset to 0 according to the rule that is used.

Black Grouping Rule		
The count: <ul style="list-style-type: none"> • Is reset to 0 when a non-black item is selected. • Stays at 0 until a black item is selected. 		
When black items are selected the count is increments as follows.		
	Black	Count goes to
	1	1
Consecutive	2	2
Consecutive	3	3
<ul style="list-style-type: none"> • When 3 black items are selected consecutively the black limit has been met. • No more black items can be selected until another color is selected. 		
Beige Spacing		
The count: <ul style="list-style-type: none"> • Is reset to 0 when a beige item is selected. 		
When non-beige items are selected the count increments as follows.		

	Non-beige	Count goes to
	1	1
Consecutive	2	2
<ul style="list-style-type: none"> • After 2 non-beige items are selected, a beige can be selected. • If more non-beige colors are selected before the beige, the count remains at 2. 		

a. The Total Count is a parameter that is used throughout the Range Source Architecture.

The parameter uses the same point; the point counts the total number of released items.

6. Close the block.

7. Compile the RLM.

The RLM will use this block to evaluate the rules for grouping and spacing the black and beige colors.

 **Note:** The Percent rule type was not used in this block.

A Percent rule type will be configured. However the rule requires a lower weight than this rule, so it will be configured in Eliminate Source By Percentages.

Example 4.1.2. Enter Values for the Rule Definition Arrays

1. Stop the project; does a configuration update and re-starts the project.
2. Open the Point Control Panel.
3. Add the configuration array points for the Eliminate Sources Based on Rule function block.
4. For each point:
 - a. Double-click the point to open its Point Control Panel Point Properties dialog box.
 - b. Select the Array tab.
 - a. Apply values to each array element as follows

	Parameter	Description		Point Array Values
		Value	Element	
1	Rule Type	G	0	RULEGS
		S	1	
2	Attribute Name	COLOR	0	COLORCOLOR
		COLOR	1	

3	Attribute Value	BLACK	0	COLORSNAME
		BEIGE	1	
4	Rule	3	0	RULEGS_COLOR
		2	1	

The initial values (Element 0) display in the Point Control Panel list.

The configuration for Eliminate Sources Based on Rule evaluation is almost ready for runtime. RSA will enter the count values during runtime.

Note: The weight that will be assigned to failures still needs to be entered in the Set Block Weights function block.

Rule Type	Attribute Name	Attribute Value	Rule	Attribute Release Count	Total Release Count
G	COLOR	BLACK	3	TBD	TBD
S	COLOR	BEIGE	2	TBD	TBD

Example 4.1.3. Add a Record Attribute Rule Function Block

1. Create a new decision, named CLASS1OLM
2. Create a new output logic module, named CLASS1RECORD.
3. Open the Record Attribute Rule function block.
4. Enter the same points for the record parameter values as were used in the Eliminate Sources Based On Rule block.

The Record Attribute Rule is the first block in the Output Logic Module.

Example 4.2. Define an Eliminate Source By Percentages Evaluation

Example 4.2. Define an Eliminate Source By Percentages Evaluation

1. MEM128
2. MEM256 and
3. MEM512.

Based on a variety of factors, production wants to apply maximum and minimum percent targets to each of the part; the purchase orders will be evaluated accordingly.

Attribute Name	Attribute Value	Upper Release %	Lower Release %
RAM	MEM128	30	20
RAM	MEM256	40	25
RAM	MEM512	50	40

Configuring this rule is done in three segments.

Example 4.2.1 (page 879)	Create an Eliminate Source By Percentages function block.
Example 4.2.2 (page 881)	Enter values for the percentage arrays.
Example 4.2.3 (page 882)	Add a Record Attribute Percentages function block.

Example 4.2.1. Create an Eliminate Source By Percentages Function Block

1. Open the Range Source function block Eliminate Sources Based on Percentages.
2. Name the block RAMPercent.
3. Configure the first array point as follows.
 - a. Open the P1 dialog box for Attribute Name Array.
 - b. Create a new virtual, text array point.

Note: The array point will identify the attribute whose values will be evaluated for percentage distribution.

- a. Enter 3 in the Elements on the Point Properties dialog box General tab.

Note: Each of the 3 elements will hold the attribute name that will correspond with the value in the Attribute Value Array. For this function block each Attribute name element will have RAM as the value.

- a. Select Saved in the in the **Initialization** field on the Virtual tab in the Point Properties dialog box.

Note: The value that is set in the array will be saved even if the project stops.

4. Configure, in addition to the Attribute Name Array, the next four array parameters that define the evaluation rules.

Note: Each element in the array holds a value that refers back to and defines the corresponding element in the Percent array.

Parameter	Point Type	Elements	Point Name
Attribute Name Array	Text	3	RAMATTNAME
Attribute Value Array	Text	3	RAMATTVALUES
Upper Release % Array	Integer	3	RAMUPPERP
Lower Release % Array	Integer	3	RAMLOWERP

5. Configure the two parameters that will provide the count data.

Parameter	Point Type	Elements	Point Name
Attribute Release Array	Integer	2	RAMATTRELEASE
Total Release Count	Integer	1	COUNTTOTAL

A brief description of the percent calculation in this example is as follows.

The actual number of items that can be released for each attribute value is constantly re-evaluated based on the percentage of the constantly increasing total count.

Attribute Value	Upper Release %	Lower Release %	Attribute Release	Total Release Count	Calculations
MEM128	30	20	27		30%
MEM256	40	25	25	90	28%
MEM512	50	40	38		42%

- a. Attribute Release is the actual number of items released for the respective attribute value.

This count takes place during runtime.

- The actual Attribute Release numbers are divided by the Total Release Count, which is the sum of all the released items.
- In this example, the MEM128 item has reached its upper release limit.

No more can be selected until the Total Release Count increases enough to bring the acceptable release number back inside the boundary of the Upper Maximum Release %.

- Based on upper release targets, MEM256 needs to increase release more than the others.

Depending on other criteria, the additional percentage may be taken from MEM128; as a result, the MEM128 percent release will decrease.

- MEM512 is in good shape.
- a. The Total Count is a parameter that is used throughout the Range Source Architecture.

The parameter uses the same point; the point counts the total number of released items.

6. Close the block.
7. Compile the RLM.

The RLM will use this block to target the minimum and maximum percent for each RAM part number.

Example 4.2.2. Enter Values for the Percentage Arrays

1. Stop the project; does a configuration update and re-starts the project.
2. Open the Point Control Panel.
3. Add the configuration array points for the Eliminate Sources Based on Rule function block.
4. For each point:
 - a. Double-click the point to open its Point Control Panel Point Properties dialog box.
 - b. Select the Array tab.
 - a. Apply values to each array element as follows

	Parameter	Description		Point Array Values
		Value	Element	
1	Attribute Name	RAM	0	RAMATTNAME
		RAM	1	
		RAM	2	
2	Attribute Value	MEM128	0	RAMATTVALUE
		MEM256	1	
		MEM512	3	
3	Upper Release %	30	0	RAMUPPERP
		40	1	
		50	2	

4	Lower Release %	20	0	RAMLOWERP
		25	1	
		40	2	

The initial values (Element 0) display in the Point Control Panel list.

The configuration for Eliminate Source Based on Percentages evaluation is almost ready for runtime. RSA will enter the count values during runtime.

Note: The weight that will be assigned to failures still needs to be entered in the Set Block Weights function block.

Attribute Name	Attribute Value	Upper Release %	Lower Release %	Attribute: Release	Total Count
RAM	MEM128	30	20	TBD	TBD
RAM	MEM256	40	25	TBD	TBD
RAM	MEM512	50	40	TBD	TBD

Example 4.2.3. Add a Record Attribute Percentages Function Block

1. Re-open the CLASS1RECORD Output Logic Module.
2. Open the Record Attribute Percentages function block.
3. Enter the same points for parameter values as were used in the Eliminate Source by Percentages block.

The Record Attribute Percentages block is the second block in the Output Logic Module.

Example 4.3. Define an Eliminate Source By Ratio Evaluation

Example 4.3. Define an Eliminate Source By Ratio Evaluation

The computer factory has three monitor part numbers: MONO15, MONO17 and MONO19.

MONO17 has a unique stand that requires more packing time than the other two.

The number of MONO17 units that can go on line need to be limited to 1 out 5.

Eliminate Source by Ratio provides a rule that will limit selection of purchase orders that include MON17 to a 1:5 ratio.

Attribute Name	Attribute Value	Block Size	Block Limit	Attribute: History
MONITOR	15	30	20	TBD

Configuring this rule is done in three segments.

Example 4.3.1 (page 883)	Create an Eliminate Source By Ratio function block.
Example 4.3.2 (page 885)	Enter values for the ratio arrays.
Example 4.3.3 (page 886)	Add a Record Attribute Ratios function block.

Example 4.3.1. Create an Eliminate Source By Ratio Function Block

1. Open the Range Source function block Eliminate Source by Ratio.
2. Name the block MONITOR15Ratio.
3. Configure the first array point as follows.
 - a. Open the P1 dialog box for Attribute Name Array.
 - b. Create a new virtual, text array point.

Note: The array point will identify the attribute whose values will be evaluated for percentage distribution.

- a. Enter 1 in the Elements on the Point Properties dialog box General tab.

Notes

- There is only 1 attribute value in the that needs to be evaluated for the ratio, MON015. Therefore, there only needs to be 1 attribute name to pair with the 1 value. For this function block the Attribute name element will have MONITOR as the value.
- CIMPLICITY treats all points as array points. Therefore, this point is an array point with one element.
 - a. Select Saved in the in the **Initialization** field on the Virtual tab in the Point Properties dialog box.

Note: The value that is set in the array will be saved even if the project stops.

4. Configure, in addition to the Attribute Name Array, the next four array parameters that define the evaluation rules.

Note: Each element in the array holds a value that refers back to and defines the corresponding element in the Percent array.

Parameter	Point Type	Elements	Point Name
Attribute Name Array	Text	1	MON_NAME
Attribute Value Array	Text	1	MON_VALUE
Block Size Array	Integer	1	MON_BLOCKSIZE
Block Limit Array	Integer	1	MON_LIMIT

5. Configure the two parameters that will provide the count data.

Parameter	Point Type	Elements	Point Name
Attribute History Array	Text	1	MON_HISTORY

A brief description of the ratio calculation in this example is as follows.

An Attribute History Array point is a string that keeps a record of the selections within the block size.

This example says that a maximum of 1 in 5 items should have a 15" monitor.

At one evaluation pass:

- CIMPPLICITY takes the:

Last 4 items that were selected +the **1 current** item under evaluation = **5 items** in the block size.

- CIMPPLICITY stores the attribute values in a string.

Value	Description
1	Matches the array value (MONO15).
0	Does not match the array value (Not MONO15)

- When no value matches, the string is 0000+0.

CIMPPLICITY looks at the five selections and evaluates a "What if" If the 5th (current) element is selected will the 1:5 ratio be broken.

Possible outcomes are as follows.

"Moving" evaluation	Last 4 Selected	Current	1:5 Decision
1	0000	0	Succeeds
2	0000	1	Succeeds
3	0001	0	Succeeds

4	0010	1	Fails
---	------	---	-------

6. Close the block.

7. Compile the RLM.

The RLM will use this block to limit Monitor 15 production to a 1:5 ratio.

Example 4.3.2. Enter Values for the Ratio Arrays

1. Stop the project; does a configuration update and re-starts the project.
2. Open the Point Control Panel.
3. Add the configuration array points for the Eliminate Sources Based on Rule function block.
4. For each point:
 - a. Double-click the point to open its Point Control Panel Point Properties dialog box.
 - b. Enter a value on the Detail tab.

Note: The entered value is actually the 0 element in the array.

Because there is only 1 element, the Array tab is not needed in the dialog box.

- a. Apply values to each parameter as follows

	Parameter	Description		Point Array Values
		Set Value	Element	
1	Attribute Name	MONITOR	0	MON_NAME
2	Attribute Value	15	0	MON_VALUE
3	Block Size	5	0	MON_BLOCKSIZE

4	Block Limit	1	0	MON_LIMIT

The values (Element 0) display in the Point Control Panel list.

The configuration for Eliminate Source by Ratio evaluation is almost ready for runtime. RSA will enter the count values during runtime.

Note: The weight that will be assigned to failures still needs to be entered in the Set Block Weights function block.

Attribute Name	Attribute Value	Block Size	Block Limit	Attribute History
MONITOR	15	30	20	TBD


Example 4.3.3. Add a Record Attribute Ratios Function Block

1. Re-open the CLASS1RECORD Output Logic Module.
2. Open the Record Attribute Ratios function block.
3. Enter the same points for parameter values as were used in the Eliminate Source by Ratio block.

The Record Attribute Ratios block is the last block in the Output Logic Module.

Step 5. Apply Weights to the Evaluation Blocks

1. Re-open the `Set Block Weights` function block.
2. Assign each block a weight in order to establish priorities.

 **guide: Guidelines**

- The value you enter in a Block n Weight parameter is assigned to the function block whose order in the evaluation list is the same as the Block n Weight number.

- It is highly recommended that you assign numbers such as prime numbers in order to avoid ties between items.
- The higher the number you assign, the higher the weight.

Items with higher weights after evaluation are more likely to fail.

Example

The four elimination blocks that have been created so far are weighted as follows.

Note: None of the blocks require immediate elimination of the item. Therefore, no block has been given a zero weight.

	Block Name	Block Parameter	Assigned Weight
1	PROC3	Block 1 Weight	7
2	COLORRule	Block 2 Weight	5
3	RAMPERCENT	Block 3 Weight	3*
4	MONITOR15Ratio	Block 4 Weight	11*


* Block 3 Weight is the most important failure criterion.

* Block 4 Weight is the least important.

Step 6. Add a Function Block to Break Ties

Two blocks are available to break ties.

Block	Eliminates
Select Source by Age	All items except the item that is highest (youngest)/lowest (oldest). Note: You specify which will be selected.
Select Lowest Weight	Eliminates all items except the item with the lowest weight

 **Important:** If you have more than one weighted block in a row, you must insert a Select Lowest Weight block at the end to break the tie.

1. The selected item moves on to the next evaluation block, if there is one.
2. If more than one item has the same lowest weight, RSA picks one to move on. Since all of these blocks are equal in evaluation, it does not matter which is selected.
3. If the selection is still too broad, then more blocks need to be added to the module and/or weights can be changed for further refine the selection.

Example

The computer factory Tracker engineers

1. Select a Select Lowest Weight block in the Range Source list.

There are no parameters. Select Lowest Weight finds the one or more blocks in the range that have the lowest weight and selects one.

2. Add the Select Lowest Weight block after the weighted evaluation blocks. An item with the lowest weight is selected and can move on to be evaluated by any remaining blocks.


Step 7. Use Range Source Applications

Step 7. Use Range Source Applications

Option 7.1 (page 888)	Re-sequence items in a range.
Option 7.2 (page 891)	Manage Holds in a Range
Option 7.3 (page 892)	Substitute orders in a range.

Option 7.1. Re-sequence Items in a Range

Option 7.1. Re-sequence Items in a Range

 **Note:** The Range Source Set Point With Attribute or Range Source Set Variable With Attribute function block can be used to store the region and region location of the selected item to virtual points, or as RCO, variables that can be referenced by the output logic controlling how an item moves.

Block	Defines
Resequence Mode Begin	At what sequence number to begin re-sequencing.
Resequence Range	At what sequence number to end re-sequencing.

Example

Range Before re-sequencing.

The sequence of order from 8 through 20 needs to be re-arranged to comply with elimination rules.


Order items arriving from external sources can be sequenced in a logical way before the order fulfillment process begins. Use a Resequence Block module to assign sequence (or rotation) numbers to items "in-place." If Orders are then moved from the range source to a single virtual region, the Reorder Region function block can adjust the "locations" of these virtual items to match the rotation number..

1. Set the range of sources.
2. Within a multiple pass Resequence Block Module:
 - a. Weight or eliminate items according to one or more business rules or other criteria. Elimination effectively pre-screens items based on their attributes and historical data. Weighting helps determine the static "appropriateness" of those items remaining.
 - b. Select the most appropriate item.
 - c. Assign the next rotation number to the selected item.

The Resequence Orders function block ends the Re-sequence Block Module and assigns rotation numbers to each item in turn. A selected item is removed from consideration for the next pass.

A - C repeat in multiple passes until all items have been assigned a rotation number.

3. If the range source includes more than one region, the decision associated with this routing logic must move the resequenced Order items to a single region where they can all be collected before implementing the next step.
4. Once Order items are assigned rotation numbers and occupy a single region, the Reorder Region function block adjusts the "virtual" Region Locations "occupied" by the orders to match the sequence of their rotation numbers. This effectively sequences the orders, since RCO is designed to process items in Location order (lowest to highest).

 **Note:** The picture above shows all function blocks in a single RLM. If the range source included more than one region, the Reorder Region block would be in a separate RLM associated with routes leaving the single region where resequenced Orders were collected.

5. Orders can now be routed to a destination based on traditional RCO decision-elimination logic. For example, you can:
 - a. Add decisions to the RCO. At least one decision should have a function block in its output logic to route Order items to the configured destination:
 - A Set Point function block, e.g. SetPointByValue, to set the transition point for that decision's route. This is the preferred method for moving the item because it leaves an audit trail for the transition.
 - A MoveItem function block to force the move.
 - a. If you added more than one decision, add one or more routing function blocks to the end of your RLM to eliminate decisions.

! **Important:** Weight is not persistent data. The weight of an item is discarded after every control cycle, or after each pass of a multiple-pass block module (Spread Range Block Module or Resequence Block Module.)

Re-sequencing and Rotation Numbers

Re-sequencing in the virtual world of ranges is different from sequencing items on the factory floor.

In the:	Sequencing/re-sequencing
Factory	Specifies and may change the physical order in which items move into and through a region.
Virtual world	Evaluates and may change rotation numbers assigned to items, thereby changing their sequence in the range. However, the item does not physically move.

Example

The factory wants to re-sequence the orders to adjust their sequence in the range according to how low they are weighted. It doesn't say "Pull this one." That's sequencing. That says, "I want this one and I want it here."

Re-sequencing changes the sequence in which items get activated.

The order that is assigned Rotation 1 will be moved to the \$OM_PLANT

1. Before re-sequencing

1. Purchase orders are currently assigned a sequence that is similar to the location sequence.
2. The orders are being evaluated by the elimination rules added to the logic module.
3. The orders need to be re-sequenced to reflect how they came through the evaluation.
4. Re-sequencing is specified to start at Rotation number 1 to cover the entire range.

L1, L2...L20 = Locations

P01,P02...P20 = Purchase orders

101, 102...120 = Rotation numbers

Orders will be moved up in the sequence of selection according to how low they are weighted static to other orders.

- a. During and after re-sequencing

Note: The rotation number colors in the graphic demonstrate conformance with the Eliminate Sources by Rule function block configuration.

Re-sequence configuration overview

Two function blocks define the beginning and end of the sequencing loop.

	Resequence Mode Begin
	Resequence Range

Insert:

- Resequence Mode Begin before the block where the loop should begin.
- Resequence Range after the last block in the loop.

After assigning rotation numbers, the following function blocks can be used outside the Resequence Block Module to manipulate or use the rotation numbers previously assigned.

Set Rotation Number to Attribute

	Advance in Order
	Rotation Pull Ahead
	Set Rotation Number to Attribute

! **Important:** Weight is not persistent data. The weight of an item is discarded after every control cycle, or after each pass of a multiple-pass block module (Spread Range Block Module or Resequence Block Module.)

Option 7.2. Manage Holds in a Range

Hold Management enables you to put a hold on the following.

Hold an item
Hold a range
Criteria set

Hold an item

Two function blocks are available for you to place a hold on and release an item.

Apply Hold to Item
Release Hold for Item

Example

The factory has a Hold Item RLM that is triggered in an emergency when an item an item goes from the MASTRBLEND range into the 4\$OM_PLANTBLND region.

Hold a Range


Two function blocks are available for you to place a hold and release an entire range.

Apply Hold to Range
Release Hold for Range


Criteria Set

A [criteria set \(page 872\)](#) enables you to apply a hold to the criteria you select when you create it.

The hold is an extended attribute.

 **Note:** A named hold is an extended attribute that specifies the range in which the hold is active. The item will actually be in the internal hold state only when it physically enters the Hold Active Range.

The following function blocks apply or release a named hold on items. A named hold includes a name for the hold, the range in which that hold becomes active and the duration of the hold.

 **Note:** Named holds on items can also be applied using the Order Execution Management Web_UI.

Option 7.3. Substitute Orders in a Range

Order substitution lets you exchange one product's purchase order with another.

Substitution function blocks exchange one product item's associated order with another order. Substitution allows a “better” order to be selected for the product item currently under consideration. When making a substitution either:


- ONLY the item's order association will be swapped, or
- ALL item attributes and status bits will be swapped.

Substitute Attributes and Status
Substitute Order

Order substitution requires an association between the Order items (tracked sales orders) and the Production item, i.e. physical part or completed assembly, that the order applies to. This association is made at the Item Type level.


To create an association between order items and the production item:

1. Create an extended attribute named ASSOCIATE either for the Order item type or for the item type that represents the physical part or assembly that is ordered.

 **Note:** Add the ASSOCIATE extended attribute to the item type you are actually routing through an RCO. For example, if you are tracking widget items and wish to substitute one sales order for a widget with a another order, add the ASSOCIATE attribute to the widget item type.

2. Enter the Item Type of the associated item as the value of the ASSOCIATE attribute.

All attributes (system, standard and extended) of either item can be referenced within the program where ever the attributes of its associated item are available. The associated items attributes appear with an @ sign prefix.

 **Note:** Currently, the associate item's attributes are accessible only through CimBasic's attribute programming interface and cannot be seen through the PRT_UI, except for the ASSOCIATE attribute, which can be seen through PRT_UI.

8. Tracker Query Engine

About the Tracker Query Engine

Tracker Query Engine is a powerful high level query engine. It has its own syntax for forming queries.

It pulls data from both the Tracker Attribute Database and the Order Execution Management runtime memory map.

Queries may be named and stored for future use, or for subdividing and abbreviating complicated queries.

rect 91, 3, 353, 125 [About External Scheduling \(page 773\)](#)

rect -2, 125, 183, 310 [About XMLT \(page 773\)](#)

rect 289, 168, 411, 251 [About the Tracker Attribute Database \(page 817\)](#)

rect 416, 124, 520, 238 [Tracker Query Configuration \(page 894\)](#)

rect 210, 274, 574, 487 [About the Product Order Management System \(POMS\) \(page 799\)](#)

rect 6, 327, 190, 392 [About DIR_WATCHER.bcl \(page 796\)](#)

rect 13, 401, 207, 496 [About CimView Order Entry \(page 810\)](#)

rect 223, 497, 352, 556 [About Order Execution Management Broadcast \(page 934\)](#)

rect 572, 399, 632, 457 [About Range Source Architecture \(page 859\)](#)

Tracker Query Configuration

Tracker Query Configuration

The Query Engine server requires

- A Tracker Attribute database (TADB).
- On the SQL Server node:
- At least Display installed.

This provides the Query Engine server access to Tracker standard, extended, and system attributes.

- The CIMPLICITY Router (Viewer) must be running on the SQL Server node.

Step 1 (page 894)	Initialize the Query Engine Database
Step 2 (page 895)	Open the Expression Browser.
Step 3 (page 897)	Select an expression.
Step 4 (page 902)	Select expressions and attributes.
Step 5 (page 903)	Write an expression.
Step 6 (page 905)	Test the expression.
Step 7 (page 908)	Apply the expression or cancel the session.

Step 1. Initialize the Query Engine Database

1. Create the project.

If you select Order Execution Mgt. when you create the project you may see messages that tell you the TADB database has not been configured.

Click OK to close these messages.

2. Open the TrackerCfg_UI.
3. Follow the procedure to [attach RCO and TADB databases \(page 820\)](#).

Note: The part of the procedure you will follow to Initialize the Query Engine database depends on whether or not the connected database is an established TADB and Query Engine database.

! **Important:** Do a project configuration update after you configure the Tracker Query Engine data source in the Tracker Configuration window. The configuration update will resolve the following problems that can occur without the update.

4. A new project has a TADB data source that was configured in the Tracker Configuration window. The Data source is populated.

Problem: If the Tracker Query Engine is opened without the configuration update, it will not find any data source to use.

5. The data source is changed in the Tracker Configuration window.

Problem: If the Tracker Query Engine is opened without the configuration update, it will still look at or try to find the old data source.

Step 2. Open the Expression Browser

Step 2. Open the Expression Browser

1. Make sure the TADB database connection has been established in the TrackerCfg_UI.
2. Open the Expression browser either:

Option 2.1. (page 895)	Through RCO configuration.
Option 2.2. (page 896)	Through the Workbench.

📖 Note: You can also [re-open \(page 902\)](#) the Expression Browser when you are in the Expression Editor.

Option 2.1 Open the Expression Browser through RCO Configuration

1. Open the Select a Function Block browser through the TrackerCfg_UI RCO configuration modules, output and routing logic.
2. Open a Select Decision by Solve dialog box as follows.

A	Expand the Templates folder.
B	Expand the Routing folder.
C	Double-click Select Decision by Solve.


The Select a Decision by Solve dialog box opens.

3. Double-click **Solve expression**.

A P1 dialog box opens.

4. Double-click the Query Engine button.

The Expression Browser opens in basic view.

 **Note:** After you first open the TrackerCfg_UI and select a TADB database make sure you do a configuration update in the Workbench. This will establish the location of the database; the Query Engine will know where to look for query results. If you do not do a configuration update the Expression Browser will not open.

Option 2.2. Open the Expression Browser through the Workbench

CIMPLICITY provides several methods to open the (Query) Expression Browser.

1. Select **Project>Tracker Configuration>Tracker Query Engine** in the Workbench left pane.
2. Select **Tracker Query Engine** in the Workbench right pane.
3. Do one of the following.

A	Click Edit>Properties on the Workbench menu bar.	
B	Click the Properties button on the Workbench toolbar.	
C	In the Workbench left pane:	
	Either	Or
	Double-click Tracker Query Engine .	a. Right-click Tracker Query Engine . b. Select Properties on the Popup menu.
D	In the Workbench right pane:	
	Either	Or

	Double-click Tracker Query Engine .	a. Right-click Tracker Query Engine . b. Select Properties on the Popup menu.
E	Press Alt+Enter on the keyboard.	

Result:s

- If the Query database is configured correctly:

The Query Engine connects to the SQL database; the Expression Browser opens in basic view.

- If the TADB datasource has not been [enabled \(page 817\)](#):

An error message opens reporting that

No added projects have a valid TADB configuration.

The Expression Browser does not open.

4. Right-click **Tracker Query Engine**.
5. Select Properties on the Popup menu.
6. Right-click **Tracker Query Engine**.
7. Select Properties on the Popup menu.

Step 3. Select an Expression

Step 3. Select an Expression


Option 3.1 (page 897)	Select an existing expression.
Option 3.2 (page 901)	Create a new expression.
Option 3.3 (page 901)	Copy an expression.

Option 3.1. Select an existing Expression

Option 3.1. Select an existing Expression

Option 3.1.1 (page 898)	Browse for existing expressions.
---	----------------------------------

Option 3.1.2 (page 900)	View the history for an expression.
Option 3.1.3 (page 901)	Select an expression to edit.

 **Note:** By default, the Query Engine will try to connect to the database for 15 seconds. After 15 seconds it will stop and display a time-out message. You can change the length of time by creating a global parameter SOLVEENGINE_CONNECT_TIMEOUT.

Option 3.1.1. Browse for Existing Expressions

Option 3.1.1. Browse for Existing Expressions

Option 3.1.1.1 (page 898)	Browse for existing expressions in basic view.
Option 3.1.1.2 (page 899)	Browse for existing expressions in advanced view.

Option 3.1.1.1. Browse for Existing Expressions in Basic View

Basic view fields enable you to search for expressions based on what they are stored under.

1. Enter one or more basic fields as follows.

rect 310, 122, 382, 302 [\(page 899\)](#)

rect 18, 85, 389, 112 [\(page 898\)](#)

rect 13, 55, 388, 83 [\(page 898\)](#)

rect 47, 33, 383, 54 [\(page 898\)](#)

rect 187, 121, 309, 302 [\(page 898\)](#)

rect 100, 120, 186, 308 [\(page 898\)](#)

rect 14, 122, 98, 308 [\(page 898\)](#)

	Field/Column	Description
A	Item type	Select from the drop down list.
		Note: Available item types were entered in the TrackerCfg_UI>PRT>Advanced>Item Types>Types (page 828) box.
B	Expression name	Valid entries include:
		<ul style="list-style-type: none"> • Name that was assigned when the expression was created. • Wild cards ? *
C	Expression text	Valid entries include:

		<ul style="list-style-type: none"> • An existing expression. • Wild cards ? *
D	Revision	Number of the revision that is active (page 900) for the expression.

[Up \(page 898\)](#)

2. Click Browse.

The expressions that fulfill your criteria display.

3. Do any of the following with the results:

A (page 523)	(If you are in an RCO configuration module) double-click an expression to select it for a function block.
B (page 901)	Copy a selected expression.
C (page 900)	View the history of a selected expression.

 **Tip:** Click a column header to sort the expressions according to the selected column.

Option 3.1.1.2. Browse for Existing Expressions in Advanced View

Advanced view fields enable you to search for expressions based on what they reference.

1. Click Advanced on the Expression Browser tool bar.

The Advanced button changes to Basic<<.

2. (Optional) fill in any of the advanced fields as follows.

Field	Description
Item Type	An item type referenced by the expression.
Attribute Group	An attribute group referenced by the expression

Attribute Name	An attribute name referenced by the expression	
Start Position	A start position referenced by the expression. Note: If the start position is 0, Query ignores the Length field.	
	Default	0
Length	A length reference by the expression. Note: If the start position is 0 Query ignores any entry in this field. if you enter a number in this field.	
	Default	0

Option 3.1.2. View the History for an Expression

1. Select an expression in the Expression Browser.
2. Click History.

The Expression History window opens, displaying versions of the named expression.

Column	Description
Active	Row that has the * is the active version. When the expression is selected, the active version is used. Note: Select a version and click Activate. The * will appear in the selected row.
Revision	Revision version; 1 is the oldest.
TimeStamp	Time the expression was created; the time is based on the local server.

User	Logged in user who created or edited the version.
Expression	The expression.
Comment	Comment entered when the expression was created or edited.
Predecessor	Revision that is most recent before a selected revision.

3. Click Activate after you select the revision you want to use.
4. Click Close.
5. Browse for the expression.

The revision number for the activated revision displays in the Expression browser.

Option 3.1.3. Select an Expression to Edit

1. Select a version of an expression in the Expression History window.
2. Click Edit.
3. The Expression Editor opens

The Expression Editor opens displaying the selected Item type, name and expression.

Option 3.2. Create a New Expression

1. Click New in the Expression Browser.

The New Expression dialog box opens.

2. Fill in the fields as follows.

Field	Use
Item type	Drop down list with types that were created in the TrackerCfg_UI Time types box.
Name	Identifies the expression when users need to select one to obtain information.

3. Click OK.

The Expression Editor opens displaying the Item type and name specified in the New Expression dialog box.

Option 3.3. Copy an Expression

1. Select an expression in the Expression Browser.
2. Click Copy.

A New Expression dialog box opens.

3. Do one of the following.
 - Keep the default new name.
 - Enter another new name in the Name field.
4. Click OK.

The Expression Editor opens displaying the Item type, the new name and the copied expression.

You can now edit the new expression.

Step 4. Select Expressions and Attributes

Step 4. Select Expressions and Attributes

At any time during the session you can browse for, retrieve and use existing expressions and attributes.

You can also query the Tracker Historical database instead of the TADB database.

rect 8, 303, 90, 330 [Option 4.1. Open the Expression Browser through the Expression Editor \(page 902\)](#)

rect 10, 328, 92, 355 [Option 4.2. Open the TADB Attribute Browser \(page 903\)](#)

Option 4.1 (page 902)	Open the Expression Browser through the Expression Editor.
Option 4.2 (page 903)	Open the TADB Attribute Browser.

Option 4.1. Open the Expression Browser through the Expression Editor

Click Expressions in the Expression Editor.

Result: The [Expression Browser \(page 897\)](#) opens.

When you select an expression in the Expression Browser Order Execution Mgt. Query will insert it in the expression you are creating in the Expression Editor. The selection is a [named \(page 918\)](#) expression.

Option 4.2. Open the TADB Attribute Browser

1. Click Attributes in the Expression Editor anytime that you want to find and insert an attribute into your expression.

The Attribute Browser opens containing the following folders and attributes.

A	Item type
B	Group attribute
C	Attributes

2. Check the attributes you want to include in the expression.
3. Click OK.

The attributes you selected are inserted in the expression.

Step 5. Write an Expression

There are three basic questions you can ask the Query engine.

1 <i>(page 906)</i>	What are the values of all the attributes for an item in the same order you asked for them?
2 <i>(page 907)</i>	Does an item meet the expression criteria?
3 <i>(page 907)</i>	What items of a specified type match the expression?

rect 25, 335, 104, 356 [Option 4.1. Open the Expression Browser through the Expression Editor \(page 902\)](#)

rect 25, 359, 101, 383 [Option 4.2. Open the TADB Attribute Browser \(page 903\)](#)

rect 136, 336, 180, 358 [\(page 905\)](#)

rect 135, 362, 180, 384 [\(page 904\)](#)

rect 186, 336, 231, 357 [\(page 905\)](#)

rect 185, 363, 231, 386 [\(page 904\)](#)

rect 236, 335, 280, 358 [\(page 905\)](#)

rect 234, 362, 280, 384 [\(page 904\)](#)

rect 287, 336, 327, 359 [\(page 905\)](#)

rect 283, 362, 329, 387 [\(page 904\)](#)

rect 335, 336, 377, 359 [\(page 905\)](#)

rect 335, 363, 379, 385 [\(page 904\)](#)

rect 383, 336, 428, 356 [\(page 905\)](#)

rect 382, 361, 427, 386 [\(page 904\)](#)

rect 432, 336, 476, 356 [\(page 905\)](#)

rect 433, 363, 476, 385 [\(page 904\)](#)


rect 518, 56, 604, 166 [Step 7. Apply the Expression or Cancel the Session \(page 908\)](#)

rect 509, 198, 613, 312 [Step 6. Test the Expression \(page 905\)](#)

Operators include:

Comparison Operators	
For TADB attributes	
=	Logical EQUALS operator.
<	Logical LESS THAN operator.
>	Logical GREATER THAN operator.
<=	Logical LESS THAN or EQUALS operator.
>=	Logical GREATER THAN or EQUALS operator.
=l	Logical SQL LIKE operator. Note: The appropriate information about acceptable wild cards and pattern matching characters can be found in SQL Server Books Online: <ol style="list-style-type: none"> 1. Access the index. 2. Select the entry called LIKE comparisons. 3. Select the entry called LIKE Transact-SQL Reference.
=r	Logical regular expression comparison operator.
For non-TADB attributes	
=	Logical EQUALS operator.
=r	Logical regular expression comparison operator.

Up (page 903)	
Boolean Operators	
!	Logical NOT operator (placed before an expression)
&	Logical AND operator.
	Logical OR operator
Up (page 903)	
Other Syntax Tools	
()	To group expression elements.
''	To identify literal values
,	To form sub-string expressions.
.	To identify attributes
\$\$	To identify references to named expressions.
Up (page 903)	

 **Note:** There is no Not Comparison Operator. When an expression should be a not expression, a ! is placed before the expression.

Step 6. Test the Expression

Step 6. Test the Expression

Option 6.1 (page 906)	Get values.
Option 6.2 (page 907)	Test items.
Option 6.3 (page 907)	List items.
Option 6.4 (page 908)	Query the Tracker Historical database.

 **Note:**

1. If a query fails, the Query Engine will display a message to report the failure and discards debug information. If you want to save the debug information you can create a global parameter SOLVEENGINEDEBUG to have the Query Engine write it to a file.
2. By default, the Query Engine will try to get results to a query for 15 seconds. After 15 seconds it will stop and display a time-out message. You can change the length of time by creating a global parameter SOLVEENGINE_QUERY_TIMEOUT.
3. You can query the Tracker Historical database instead of the TADB and Tracker databases. The Tracker Historical database can be used with any of the test options.

Option 6.1. Get Values

1. Display a new or existing expression in the Expression Editor.

Example

A customer can select from options for each computer part.

An expression can be written that will display a selected item's customer and selected parts.

2. Click Get Values.

An Item ID text box opens.

3. Enter an Item ID.


4. Click OK.

An Attribute Test Results window displays the specified attribute values for the selected item.

Column	Description
Index	Index of the matching attribute.
Value	Data returned based on the expression.
Data Type	Attribute data type that was specified in the TrackerCfg_UI window.

5. Click OK.

Result: The Attribute Test Results window closes and the Expression Editor displays.

 **Tip:** You can copy and paste the attribute value to another location.

6. Select the line with the value you want to copy.

7. Click Copy Value.
8. Paste the value wherever you need to use it.

Option 6.2. Test Items

Test Items tests a selected item to determine if its values test True for the values specified in an expression.

1. Insert a new or existing expression into the Expression Editor.

Example A computer distributor may have one or more computer items with 17" monitors listed on a Bill of Materials (BOM).

An expression can be written that enables you to test a selected computer item to determine if its BOM includes a 17" monitor.

2. Click **Test Item**.

An Item ID dialog box opens.


3. Select an item from the drop down list.
4. Click OK.

If the item:

- Fulfills the expression criteria, a message notifies you the item matches the expression.
- Does not fulfill the expression criteria, a message notifies you that the item does not match the expression.

5. Click **OK**.

The message closes and the Expression Editor displays.

 **Note:** Expressions that are valid for Test Item should also be valid in [List Items \(page 907\)](#) .

Option 6.3. List Items

List items lists all the items in the database that contain the values specified in your expression.

1. Display a new or existing expression in the Expression Editor.

The expression must include a value that may apply to one or more items.


Example

A computer distributor may have one or more computer items with 17" monitors listed on a Bill of Materials (BOM).

An expression can be written that enables you to list all of those computers.

2. Click List Items.

A Results from List Items browser opens displaying the items found that test true for the expression.

 **Note:** Expressions that are valid for List Items should also be valid in [Test Item \(page 907\)](#) .

Option 6.4. Query the Historical Database

By default the Query Engine queries Tracker and TADB records.

You can also query the [Tracker Historical database \(page 838\)](#) instead of the TADB and Tracker databases.

Check the Use Historical check box.

Result: The Query Engine will query the Historical database.

Step 7. Apply the Expression or Cancel the Session

Click at any time during the session.	
Apply	Save the expression.
Apply & Activate	Save the expression and make it the active version. This version will be used when the expression is called.
Click to end the session	
OK	Close the Expression Editor. If the expression has been edited a Comment dialog box displays. <ol style="list-style-type: none"> 1. Enter a comment. 1. Click OK.
Cancel	Close the Expression Editor without saving the expression.

Tracker Query Expression Syntax

In order to work effectively with the Query Expression Editor it is important to have a working knowledge of TADB structure and Basic Tracker and vocabulary.

Rules for building Query expressions include:

Syntax conventions used for building queries.
Methods in the expression language.
Simple non-terminals.
Complex non-terminals.
Named expressions.

Syntax conventions used for building queries

The syntax conventions use different font attributes for different components of the syntax.

Syntax	Applied to	Definitions
Italic	Non-terminals.	Placeholders in the syntax and are defined elsewhere in this syntax summary.
Bold	Terminals	Literal reserved words and symbols that must be entered as shown. Characters are always case sensitive, except for NULL, which can be any possible case combination.
opt	Optional	Non-terminals followed by opt are always optional. Example { expression opt } Indicates an optional expression enclosed in curly braces.
Default typeface	Terminal characters	Characters be used as terminals in expressions.

Alternative definitions are listed on separate lines, except when prefaced with the words one of.

Methods in the expression language

The three methods that use the expression language are:

Method	Definition
GetValues(ItemID, AttributeList <i>(page 910)</i>)	Retrieve a set of values from a particular item's attributes.
ListItems(ItemType, Expression <i>(page 910)</i>)	Find out if a particular item does or does not meet a set of criteria.
TestItem(ItemID, Expression <i>(page 910)</i>)	Find all the items that do match a set of criteria.

Start Symbols

The basic building blocks that will make an expression valid are the start symbols, attribute-list and expression.

Non-terminal	Definition
(Start Symbols)	
attribute-list	list-attribute (page 912) attribute-list ; list-attribute Note: A complete attribute-list symbol must be passed to <code>GetValues</code> .
expression	term (page 912) ! opt (expression) expression & expression expression expression type-id { expression } Note: A complete expression symbol must be passed to <code>TestItem</code> or <code>ListItems</code> .

Simple non-terminals

Simple non-terminals can be combined to form more complex non-terminals.

Non-terminal	Definition
Attribute-id	Identifier (page 911)
Digit	One of 0 1 2 3 4 5 6 7 8 9
Group-id	Identifier (page 911)
Length	Number
Letter	One of _ a b c d e f g h i j k l m n o p q r s t u v w x y z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Named-expression-name	Identifier (page 911)
Number	Digit, e.g. 5 Number digit, e.g. 25
Relational	One of = < > <= >= != =r (page 903)
Space	A single blank space character
Start-pos	Number
Text	<p>Depends on the:</p> <ul style="list-style-type: none"> Operator that precedes the text and Data type of the attribute being compared. <p>Will elicit one of the following results.</p> <ul style="list-style-type: none"> The text will be converted to the correct type automatically. When either of the following operators are used: <p>=l =r the target attribute will be converted to character form before comparison.</p> <ul style="list-style-type: none"> If a type conversion cannot take place, an error condition will result and the query will not execute.
	Timestamp literals can take one of two forms.

Non-terminal	Definition	
	Absolute time stamps	' <code>yyyy-mo-dd[hh:mi:ss]</code> '
		yyyy 4 digit year
		mo 2 digit month
		dd 2 digit date
		hh hours
		mi minutes
		ss seconds
	static time stamps	' <code>M:Q'</code> '
		M Measurement unit, which is one of the following:
		yy years
		q quarters
		m months
		d days
		ww weeks
		h hours
		n minutes
		s seconds
		ms milliseconds
		Q An integer quantity of time from the current time in specified measurement units. Negative numbers are used to represent previous time stamps.
Type-id	Identifier (page 911)	
Value	Either: ' text ' <code>null min() max()</code>	
minmax	<code>.min(identifier) .max(identifier)</code>	

Complex non-terminals

Complex definitions usually have more than one rule.

Non-terminals	Definition
attribute	group-id . attribute-id start-pos <code>opt</code> length <code>opt</code>
identifier	letter digit identifier letter identifier digit identifier space identifier

Non-terminals	Definition
list-attribute	attribute group-id (sub-expression) minmax _{opt} . attribute-id start-pos _{opt} length _{opt} type-id { list-attribute }
sub-attribute	attribute-id start-pos _{opt} length _{opt}
sub-expression	sub-term sub-expression & sub-expression sub-expression sub-expression
sub-term	! _{opt} attribute Relational values ! _{opt} sub-attribute Relational values
term	! _{opt} attribute Relational values ! _{opt} group-id (sub-expression) minmax _{opt}
values	value values, value

Named expressions

A named expression may be a single symbol or multiple symbols and may not contain a text symbol except in the context of completing a value symbol.

The syntax for using a named expression is:

\$ named-expression-symbol **\$**

A named expression can be inserted anywhere except between ‘ characters.

Tracker Query Reserved Identifier Attributes

The following reserved identifiers can be used in Order Execution Mgt. group names.

- TADB.<attribute>
- Trk.<attribute>

TADB.<attribute>

TADB.<attribute> system attributes are as follows.

- id
- itemType
- itemClass
- refid
- active
- created
- modified
- externalHoldActive
- groupID
- internalHoldActive
- internalHoldReason

- itemStatusHI
- itemStatusLO
- itemTrackingType
- parentitemID
- pKey
- regEntryTime
- region1ID
- region1ID
- region1Location

Trk.<attribute>

Refers to the collection of all standard and extended Tracker attributes.

Example

Trk.COLOR

Tracker Query Expression Descriptions

Tracker Query Expression Descriptions

You can use several different types of expressions to query your Tracker and TADB data.

Descriptions and examples of expressions include:

1 <i>(page 914)</i>	Basic expression for Tracker attributes and normal TADB groups.
2 <i>(page 916)</i>	Complex expression for Tracker attributes and normal TADB groups.
3 <i>(page 917)</i>	Basic expressions for TADB list groups.
4 <i>(page 918)</i>	Named Query expressions.
5 <i>(page 923)</i>	Simple relationship expression for TADB normal group attributes.
6 <i>(page 924)</i>	Complex relationship expression for TADB normal group attributes.
7 <i>(page 925)</i>	Simple expression for attribute lists.

8 (page 926)	Demonstration only expression to understand TADB attribute lists.
9 (page 927)	Simple nested expression for TADB list group attributes.
10 (page 928)	Min/Max expressions.
11 (page 932)	Simple relationship expression using TADB attribute lists.
12 (page 932)	Nested relationship expression using TADB attribute lists.

! **Important:** Query expressions are case sensitive. For example, if you have a group named Parts, you must enter Parts in the expression, not parts.

1. Basic Expression for Tracker Attributes and Normal TADB Groups

Purpose

Test a selected section of an attribute's value.

Assumptions

There is one Attribute Group (in the expression), defined as follows.

Order Card	Attributes include A
------------	----------------------

Query Expression

Order Card.A,3,2='xy'


Where

Criteria	Is/are the:
A	Order card attribute
3	Position 3 in the A attribute value.
2	Return 2 characters in the A attribute value. Note: This is a character comparison dependant on the collation order of your database.
'xy'	Required characters in the A attribute value positions 3 and 4.

Query Expression says:


Determine if an Order Card A attribute value, positions 3 and 4 contains the characters xy.

bnf syntax Expression

 **Note:** This syntax in this expression is the basis of all expressions and can be summarized in this .bnf expression.

```
simpleExpression ::= [notOperator] group.attribute[,position[,length]]
comparisonOperator value
```

Where

notOperator	(!) When used, defines the remainder of the expression as what the value should not be.
Group	Either a <ul style="list-style-type: none"> • TADB attribute group name or • Reserved name Trk (refers to the collection of all standard and extended Tracker attributes).
position	Starting position in a value to begin the comparison.
length	Number of characters to compare.
	<p>If:</p> <ul style="list-style-type: none"> • <code>position</code> is specified and <code>length</code> is not <p>Then <code>length</code> is assumed to be 1.</p> <ul style="list-style-type: none"> • Neither <code>position</code> nor <code>length</code> are specified <p>Then The entire attribute is included in the comparison.</p>
ComparisonOperator	<p>= < > < = >= =l =r</p> <p> Note: There is no Not ComparisonOperator. When an expression should be a not expression, a ! is placed before the expression.</p> <p>Example !Order Card.A,3,2='xy'</p>

 **Tip:** You can use the keyword NULL to query NULL fields in the database.

Example

When you execute a list query on:

- `Order Card.Target Gate Release Date=l'%'`

The query returns all the items that have a release date.

- `Order Card.Target Gate Release Date=NULL`

The query returns all the items that have a <NULL> release date.

2. Complex Expression for Tracker Attributes and Normal TADB Groups

Purpose

Test a combination of values in an attribute group.

Assumptions

There is one Attribute Group (in the expression), defined with attributes as follows.

Order Card	Attributes include C and G.
------------	-----------------------------

Query Expression

`(Order Card.C,4>'3'&!Order Card.G,8='7') | Order Card.C,8,2='yy'`

Where

Criteria	Is/are the:
C	Order card attribute.
4	Position 4 in the C attribute value.
'3'	Value must be greater than 3.
G	Order card attribute.
8	Position 8 in the G attribute value.
'7'	Required character in the G attribute value position 8.
C	Order card attribute.
8	Position 8 in the C attribute value.
2	Return 2 characters in the attribute C value.
'yy'	Accepted characters in the C attribute value positions 8 and 9.

Query Expression says:

Determine:

If the Order Card group:


- Has in C attribute value, positions 3 and 4, a value greater than 3 and

- Does not have in G attribute value, position 8, a character that is not 7.

Or

If the Order Card group:

- Has in C attribute positions 8 and 9, the characters yy.

 **Note:** Parentheses, Ands and Ors can be nested within each other and appended together to any desired length.

The example above is designed to show every position in which these operators can be used.

3. Basic Expressions for TADB List Groups

Purpose

Test an item to see if it has a specified quantity of a certain part.

Assumptions

There is one List group (in the expression), defined as follows.

Parts	Attributes include quantity, id.
-------	----------------------------------

Query Expressions

1

```
Parts.quantity='7'&Parts.id='id1'
```

2

```
Parts(quantity='7'&id='id1')
```

Expression 2 is better. It often saves typing and usually yields better performance.

The group part of the syntax has been omitted from the internal sub-expression because it has already been provided outside the parentheses.

Important: Expression 2 syntax is required in attribute list expressions. In TestItem/ListItems expressions, it is optional.

Where

Criteria	Is/are the:
quantity	List group attribute.

'7'	Number of parts expression is looking for.
id	List group attribute.
'id1'	Part name.

Query Expressions say:

Return True for items that have 7 id1 parts in their Parts group

! **Important:** In this syntax no parentheses are allowed within the parentheses as shown in these examples. All other operators are allowed.

☰ **Note:** Make sure you include enough criteria in expressions to make them meaningful.

For example, the following expression is valid.

```
Parts.quantity='7'
```

However, the expression only determines if any part has a quantity of 7.

This does not tell you much.

4. Named Query Expression

4. Named Query Expression

Named expressions are a type of shorthand that can be inserted into expressions you are writing.

Named expressions can:

- Save configuration time.
- Help avoid typing errors.


4.1 <i>(page 919)</i>	Guidelines for Named Expressions
4.2 <i>(page 920)</i>	Evolution for Complex Named Expressions
4.3 <i>(page 921)</i>	Simple Named Expressions
4.4 <i>(page 922)</i>	Named Expressions inserted into Expressions

4.1. Guidelines for Simple Named Expressions

1. The syntax to surround a named expression is `$ExpressionName$`.

Example

Expression	Saved as	Insertion Syntax
Order Card	OC	<code>\$OC\$</code>

 **Note:** The `$name$` syntax works just like an `#include` directive in the “C” programming language. The `$name$` is removed; the text it represents is inserted; the expression evaluation continues.

2. A partial identifier or a partial literal will not work as a named expression.

Example

Assume

Order Card is an attribute group.

The following expression is **invalid**:

Given

<code>o =</code>	Order
<code>c =</code>	Card

This will not work because the group name is split in half.

`O C.A='xyz'` fails.

A single token in the `bnf` syntax in the detail design is the smallest an expression can be.

Fractional tokens as demonstrated in this example are not supported.

3. No substitution occurs between quotes.

The following expression is **invalid**:

Given:

<code>val=</code>	xyz
-------------------	-----

`Order Card.A='val'` fails.

The following expression is **valid**.

Given:

```
"val='xyz' "
```

```
"Order Card.A=$val$" succeeds.
```

4. Expressions are context sensitive.

4.2. Evolution for Creating a Named Expression

Although named expressions can be far more complex than the example expression used in section 4.3., this expression can help demonstrate an evolution of and guidelines for more complex expressions.

The

```
Parts($pqty$ = '7' & Parts.id = 'id1' )
```

The above named expression calls for the same results as the demonstrated [basic expressions \(page 917\)](#) for TADB List groups.

Return true for items that have 7 id1 parts in their Parts group.

The evolution is as follows:

1. Type a basic expression.

```
Parts.quantity='7' & Parts.id='id1'
```

Parts.quantity='7'	The quantity in Parts is 7.
Parts.id='id1'	The ID in Parts is id1.

2. Save time in typing.

```
Parts(quantity='7' & id='id1')
```

Parts(quantity='7'.....)	The quantity in Parts is 7.
Parts(.....&id='id1')	The ID in Parts is id1.

This says exactly the same thing as above, but it is shorter and generates a different query internally.

The group Parts in the syntax is omitted from the internal sub-expression. It has already been provided outside the parentheses.

3. Use a longhand version to move up to using a more complex named expression.

Parts(Parts.quantity='7' & Parts.id='id1')

Parts(Parts.quantity='7'...)	The quantity in Parts is 7.
Parts(...&Parts.id='id1')	The ID in Parts is id1.

This executes the same way as the second example.

! Warning: If the group outside the parentheses does not match the group inside the parentheses:

- The expression will fail to parse and
- An error will be generated.

However, the benefit of this permitted syntax is that Parts.quantity can be saved as a named expression Ptqy.

4.3. Simple Named Expressions

Purpose

Create a simple expression that can be used as a named expression.

Expression 1

Assumptions

There is one List group (in the expression), defined as follows.

Parts	Attributes include quantity
-------	-----------------------------

Query Expression

Parts.quantity

Where

Criteria	Is/are the:
quantity	List group attribute

Expression Name

`Pqty`

Expression 2

Assumptions

There is one Attribute Group (in the expression).

Order Card

Query Expression

`Order Card`

Expression Name

OC

4.4. Named Expressions inserted into Expressions

Purpose

Test an item to see if it has a specified quantity of a certain part.

Include a named expression in the expression.

Assumptions

There is one List group (in the expression), defined as follows.

Parts	Attributes include quantity, id.
-------	----------------------------------

Query Expression

```
Parts($pqty$ = '7' & Parts.id = 'id1')
```

Where

Criteria	Is/are the:
Pqty	Parts.quantity named expression.
\$\$	Syntax surrounding the named expression.
id	Nested Parts list group attribute
'id1'	Part name

Query expressions say:

Return True for items that have 7 id1 parts in their Parts group

5. Simple Relationship Expression for TADB Normal Group Attributes

1. There are two Item types (in the expression), defined as follows.

Order	Invoice information includes what the customer wants.
Customer	Information includes address and billing.

2. There is one Attribute group (in the expression), defined as follows.

Address	Attributes include <code>state</code> .
---------	---

3. When an Order item is created, it is associated with the Customer item that should receive the finished product.

Query Expression

```
TestItem("Order1", "Customer{Address.State='NY'}")
```

Where

Criteria	Is/are the:
TestItem	Sub-expression test.
Order1	Order item
State	Address group attribute
'NY'	State attribute value.

Expression says:

Find `Customer` items that have a relationship with the item `Order1`

Where the

`Customer` item's `Address` (group) contains `NY` as the value for its `State` (attribute).

Note: The relationships are looked for in the TADB's `RelatedItems` table.

.bnf syntax Expression

```
ItemType{Expression}
```

This syntax means that:

- Relationships can be combined and nested to near limitless complexity
- Parentheses can be combined and nested to near limitless complexity.

6. Complex Relationship Expression for TADB Normal Group Attributes

1. There are three item types (in the expression), defined as follows.

Component	Components that are built into the product.
Order	Invoice information includes what the customer wants, including the <code>Order Card</code> group.
Customer	Information includes the <code>Address</code> group and billing.

2. There are two Attribute groups.

Address	Attributes include <code>State</code> and <code>City</code> .
Order Card	Attributes include <code>G</code> .

Query Expression

```
TestItem("Component1", "Order {Customer {Address.State='NY' &Address.City='Albany' } &Order Card.G, 6='b' }")
```

Where

Criteria	Is/are the:
TestItem	Expression test.
Component1	Component item
State	Address group attribute
'NY'	State attribute value
City	Address group attribute
Albany	City attribute value
G	Order card attribute
6	Position 6 in the G attribute value
b	Character in position 6 in the G attribute value.

Expression says:

Determine if the Component1 item is associated with an order that:

- Has the character b in the 6th position in the Order Card's attribute G value

and where that order

- Is associated with a Customer item in Albany,NY.

7. Simple Expression for Attribute Lists

Purpose

Display all values for one attribute and values in selected positions for another.

Assumptions


There is one Attribute Group (in the expression), defined as follows.

Order Card	Attributes include A, C.
------------	--------------------------

Query Expression

Order Card.A;Order Card.C,3,2

Criteria	Is/are the:
A	Order card attribute
C	Order card attribute

 **Note:** List of attributes are separated by semi-colons

Query Expression says:

Return the Order Card A attribute and 2 characters from the C attribute value positions 3 and 4.

Query Expression Results

The results come back:

- In a 2-dimensional array.
- The columns are:
 - A 0 based index in the first column.
 - Returned values in the second column.
- Two indexed rows correspond to the expression's returned values.

The results for the above attribute list might look like this:

0	DFG7 J3356F5CS
1	G9

The indexed rows relate to the expression as follows.

Expression	Index	Result
Order Card.A	0	DFG7 J3356F5CS
Order Card.C,3,2	1	G9

Note: The index column is for when attribute from a list group is requested.

8. Demonstration only Expression to understand TADB Attribute Lists

1. There is one attribute normal group (in the expression), defined as follows.

Order Card	Attributes include A, C.
------------	--------------------------

2. There is one attribute list group (in the expression), defined as follows.

Parts	Attributes include quantity.
-------	------------------------------

Query Expression 1

`Order Card.A;Parts.quantity;Order Card.C,3,2`

Where

Criteria	Is/are the:
A	Order card attribute
quantity	List group attribute
C	Order card attribute

Important: This expression is for demonstration only. It shows how Index numbers are used. However, it is not useful because it fails to display what quantity goes with what part.

Query Expression says:

Return values as follows:

- All the values for A attribute.
- Quantity of parts in the parts list.
- C attribute value characters in positions 3 and 4.

Query Expression Results

The results come back in a 2-dimensional array with 5 rows.

- The columns are:
 - A 0 based index in the first column.
 - Returned values in the second column.
 - Five indexed rows correspond to the expression's returned values.

The results for the above attribute list might look like this:

0	DFG7 J3356F5CS
1	45
1	55
1	1
2	G9

The indexed rows relate to the expression as follows.

Expression	Index	Result
Order Card.A	0	DFG7 J3356F5CS
Parts.quantity	1	45
Parts.quantity	1	55
Parts.quantity	1	1
Order Card.C,3,2	2	G9

Quantities are returned for three parts. The results do not tell what the parts are.

9. Simple Nested Expression for TADB List Group Attributes

Purpose

Determine the number of specified parts.

Assumptions

There is one List group (in the expression), defined as follows.

Parts	Attributes include quantity, id.
-------	----------------------------------

Query Expression

```
Parts(id='id1').quantity
```

Where

Criteria	Is/are the:	
id	Nested Parts list group attribute	(Parts.id)
'id1'	Part name	

quantity	List group attribute	
----------	----------------------	--

Query Expression says:

Return the quantity of id1 parts.

Query Expression Results

The results for the above attribute list might look like this:

1	55
---	----

The quantity of the part id1 is 55.

Note: Anything you can do in the sub-expressions mentioned in [section 3 \(page 917\)](#) can be done in these sub-expressions.

10. Min/Max Expressions

Purpose

- Find items that contain specified minimum or maximum criteria
- Display attribute values for items that contain specified minimum or maximum criteria.

Assumptions

A Parts table has the following values for Order items.

id	partno	rev
i1	1	5
i1	1	2
i1	2	3
i1	3	1
i2	1	1
i2	2	2
i2	5	1

Query Expressions

Several types of query expressions can be created using the min/max functionality.

A (page 929)	<code>listitems</code> expression for normal TADB groups.
------------------------------	---

B (page 929)	<code>listitems</code> expression for list TADB groups.
C (page 930)	<code>getvalues</code> expression for normal TADB groups
D (page 931)	<code>getvalues</code> expression for list TADB groups

1. `listitems` expression for normal TADB Groups

```
listitems "orders", "parts.rev=max()"
```

Where

Criteria	Is/are the:
orders	item type
Parts	Table name
rev	Revision number
max()	Maximum value

1. The Query expression asks:

Which Order type item or items has/have the maximum revision number in the Parts table.

2. The answer is:

The maximum `rev` in the table is 5. The Order item returned is `i1`.

id	partno	rev
i1	1	5
i1	1	2
i1	2	3
i1	3	1
i2	1	1
i2	2	2
i2	5	1

a. `listitems` expression for list TADB groups

```
listitems "orders", "parts(partno='2').min(rev)"
```

Where

Criteria	Is/are the:
orders	item type
Parts	Table name
partno	Part number
2	Part number value to be included in the query.
min()	Minimum value
rev	Revision number

3. The Query expression asks:

Which Order items that have a part number of 2 have the minimum revision number?

4. The answer is:

i2, part number 2

Two items have a part number of 2; the i2 item has the minimum revision number of those part numbers, which is 2.

id	partno	rev
i1	1	5
i1	1	2
i1	2	3
i1	3	1
i2	1	1
i2	2	2
i2	5	1

a. `getvalues` expression for normal TADB groups

```
getvalues "i1", "parts(rev=max()).partno"
```

Where

Criteria	Is/are the:
i1	Part id attribute value
Parts	Table name
rev	Revision number
max()	Maximum value

partno	Part number
--------	-------------

5. The Query expression asks:

Get the part number of the `i1` item that has the maximum `rev` number.

6. The answer is:

1

partno 1 has the greatest `rev` for item `i1`, which is 5.

id	partno	rev
i1	1	5
i1	1	2
i1	2	3
i1	3	1
i2	1	1
i2	2	2
i2	5	1

a. `getvalues` expression for list TADB groups

```
getvalues "i1", "parts(partno='1').min(rev).rev"
```

Where

Criteria	Is/are the:
i1	Part id attribute value
Parts	Table name
partno	Part number
1	Part number to be queried
min()	Minimum value
rev	Revision number attribute to return

7. The Query expression asks:

What is the minimum revision number for `i1` items that have a 1 part number.

8. The answer is:

2

Records for item `i1` with `partno` 1 have 2 rev values, 2 and 5. The minimum of those values is 2.

id	partno	rev
i1	1	5
i1	1	2
i1	2	3
i1	3	1
i2	1	1
i2	2	2
i2	5	1

11. Simple Relationship Expression using TADB Attribute Lists

1. There is one Item type (in the expression), defined as follows

Customer	Information includes address and billing.
----------	---

2. There is one Attribute group (in the expression), defined as follows.

Address	Attributes include <code>state</code> .
---------	---

Query Expression

```
GetValues("Order1", "Customer{Address.State}")
```

Where

Criteria	Is/are the:
GetValues	Sub-expression command to return values.
Order1	Order item
State	Address group attribute

Query Expression says:

Return the state address for any customer items that are associated with Order1.

12. Nested Relationship Expression using TADB Attribute Lists

1. There are two Item types (in the expression), defined as follows

Order	Invoice information includes what the customer wants.
Customer	Information includes <code>Address</code> group and billing.

2. There is one Attribute group (in the expression), defined as follows.

Address	Attributes include <code>State</code> .
---------	---

Query Expression

```
GetValues("Component1", "Order{Customer{Address.State}}")
```

Where

Criteria	Is/are the:
GetValues	Expression command to return values.
Component1	Body component
State	Address group attribute

Query Expression says:

Return the state (address) for the customer that Component1 will go to when the customer's associated order is filled.


Query Engine Troubleshooting

Query Engine Troubleshooting

There may be a rare instance when the Query engine does not seem recognize the TADB when a user tries to run an expression.

This could happen, for example, if someone manually entered an expression directly into the database table instead of through the Query Engine or if someone directly changed the structure of the database.

If you cannot correct the problem any other way, you can use a Query Engine Reset feature.

 **CAUTION:** The Query Engine reset tool should be used only as a last resort.

When you use the tool it:

1. Waits for any queries that are going on to stop first.
2. Puts all queries that have not yet started on hold.
3. Will stop until the last running query is done,

The Stop Queries will last until the longest currently running query finishes, which could require a substantial amount of time.

This is the issue that particularly requires your attention before you decide to go ahead.

4. Does the following when the last query is completed.
 - a. Reads the meta-data.
 - b. Re-reads the expressions out of the database.
 - c. Allows the queries to resume when the reset is complete.

Do the following to use the Reset tool.

1	Click the Expression Browser button.
2	Select Reset on the menu.

9. *Order Execution Management Broadcast*

About Order Execution Management Broadcast

Order Execution Mgt. Broadcast is the delivery of a configurable list of product related information (including at least build options, location information, other/supporting data and subsets of the unit bill of material) to plant floor devices and to suppliers. Broadcasts may be triggered either by vehicle movement, by automation, by manual demand or by timed event.

Order Execution Management. Broadcast Components Overview

All Order Execution Mgt. Broadcast output devices are connected over the Ethernet, either via direct connection or through a serial or parallel terminal server.

Features Overview

Broadcast Form tools that include a WYSIWYG Graphical Editor. The WYSIWYG tool contains a graphical formatting tool. enables a user to create forms and data streams to define data including: Color, One and two dimensional bar codes, Text, and Graphics. A user can preview and test the functionality and output the data to configurable printers. Broadcast treats all devices as Windows print.

Broadcast device groups to associate available resources in a CIMPLICITY project with selected printers.

BCO function blocks to direct the flow of printing forms.

Queue Manager that enables a user to manage and keep track of broadcast jobs, suspend, resume and cancel broadcast jobs, resend jobs and alarm on error. Broadcast queue manager is Web enabled.

Graphic Overview


Order Execution Management Broadcast Runtime Overview

Tracker Broadcast operates as follows during runtime.

The configured Broadcast function blocks are executed.

Broadcast gathers the required item information. Information, that is selected in ASCII and/or WYSIWYG forms comes from any of the following sources: Tracker extended attributes, Tracker attributes, Tracker Attribute Database, System data, or Query engine.

Broadcast then: Merges the data with the form and generates a file with a name that is generated from the fields in the generic field configuration. This output file is queued to the specified device through the Windows Print Queue Manager. Prints the form when it is first in the queue.

1.  **Note:** If Broadcast is unable to place the job in the device queue, the following occurs:

Reason unable to place job in queue	Result
An error	Tracker redirects the job to the secondary device, if a secondary device is configured in the broadcast.
Queue size has reached its configured limit	An alarm is generated. Note: The user can configure alarms at two levels of the queue.

Enable Order Execution Management Broadcast

1. Click Project>Properties on the Workbench menu bar. The Project Properties dialog box opens.
2. Select the General tab.
3. Make sure that Broadcast is checked in the Options box.
4. Select the Options tab.
5. Make the following selections.

	Feature	Selection
A	Description	Enter an informative description.

B	Enable project broadcast	Check..
C	Enable project multicast	Check..
D	Computer	Make sure the Broadcast server name displays.
E	Configuration security	Check.

Broadcast Forms

About Broadcast Forms

Broadcast forms can be printed to any selected device.

Forms include:

- ASCII forms
- WYSIWYG forms

ASCII forms

[ASCII Forms \(page 937\)](#) are simple text data. Broadcast provides a text editor in which you can define the ASCII forms that are broadcast to output devices.

ASCII forms and their companion Control Character Token files exist in three states.

Files in each state are located in a separate folder in the following project directories.

Form/File	Directory
ASCII forms	<project name>\CimForms\ASCIIForms
Control Character Token files	<project name>\CimForms\CCharFiles

The subfolders in which the files are located according to the form's states are as follows.

Subfolder	Form State	Description
Old	Archived	Were published, but have been removed from production.
Prod	Published	<ul style="list-style-type: none"> • Currently being used in production. • Located in a predefined directory. • Available to Broadcast function blocks.
Working	Under development	<ul style="list-style-type: none"> • Only accessed from the ASCII Form editor • Tested against live data. • Not available to the Broadcast Function Blocks.

WYSIWYG forms

[WYSIWYG forms \(page 964\)](#) are created in a WYSIWYG Graphical Editor in which you can layout several types of objects including

- Text, and bar codes that can report real data when they are printed.
- Rectangle and line graphic objects to aid in form layout.
- Graphic images.

Files in each state are located in a separate folder in the following project directories.

Form/File	Directory
WYSIWYG Forms	<project name>\CimForms\WYSIWYGForms

The subfolders in which the files are located dividing the forms into functionality are as follows.

Subfolder	Form State	Description
Old	Archived	Were published, but have been removed from production.
Prod	Published	<ul style="list-style-type: none"> • Currently being used in production. • Located in a predefined directory. • Available to Broadcast function blocks.
Working	Under development	<ul style="list-style-type: none"> • Only accessed from the ASCII Form editor • Tested against live data. • Not available to the Broadcast Function Blocks.

ASCII Forms

ASCII Form Configuration

Step 1 (page 938)	Open the CimForms Text Editor.
Step 2 (page 938)	Create a Control Character Token file.
Step 3 (page 946)	Create an ASCII form.
Step 4 (page 956)	Compile an ASCII form.

Step 5 (page 957)	Add/publish an ASCII form.
Step 6 (page 960)	Test print an ASCII form.

Step 1. Open the CimForms Text Editor

1. Expand the CimForms folder in the TrackerCfg_UI left pane.
2. Right-click **ASCII Forms**.
3. Select Text Editor on the popup menu.

The CimForms Text Editor opens.

Step 2. Create a Control Character Token File

Step 2. Create a Control Character Token File

ASCII forms must contain printer control sequences, which are cryptic strings of alphanumeric characters and special characters. The control characters that you use depend on the printer to which the ASCII form will be sent. You should be able to find the specific control character syntax in your printer documentation

You can:

- Define the control sequences in a Control Character Token file.

A Control Character Token file must be created for each device type.

- Associate the Control Character Token file with one or more ASCII forms.

If you use more than one ASCII form, all of the forms can use the same control character file or any of the ASCII forms can use its own file.

 **Note:** This step is not necessary if the Control Character Token file you plan to use is already up-to-date and published.

The ability to enter control characters in Control Character Token files that are separate from the ASCII forms helps to:

- Reduce typing (control characters often have to be repeated several times).
- Improve readability of the ASCII form

- Reduce the need for the form developer to remember the control sequences.

Steps to make a Control Character Token file available for use are:

Step 2.1 (page 939)	Open a Control Character Token file.
Step 2.2 (page 941)	Write a Control Character Token file.
Step 2.3 (page 941)	Save the Control Character Token file.
Step 2.4 (page 942)	Compile the Control Character Token file.
Step 2.5 (page 943)	Add/publish the Control Character Token file.

Step 2.1. Open a Control Character Token File

Step 2.1. Open a Control Character Token File

You can create a new file or edit an existing Control Character Token file in the CimForms Text Editor.

Both the new and the updated file will be compiled and added/published to be available for Broadcast use.

Option 2.1.1 (page 939)	Open a New Control Character Token file.
Option 2.1.2 (page 940)	Open an existing control character token file.

Option 2.1.1. Open a New Control Character Token File

1. Do one of the following in the CimForms Text Editor.

Method 1

- a. Click File on the CimForms Text Editor menu bar.
- b. Select New.

Method 2

Click the New button on the CimForms Text Editor toolbar.

Method 3

Press Ctrl+N on the keyboard.

A New dialog box opens.

2. Select Control Character Token File.

A blank Control Character Token file opens.

3. (Optional) Click File>Page setup on the CimForms Text Editor menu bar.

4. Change the page's default margins.

Notes

- Measurements are in millimeters.
- You can also change measurements for an existing file.

5. Click OK to close the Page Setup dialog box.

The file is ready for writing and editing.

Option 2.1.2. Open an Existing Control Character Token File

1. Do one of the following in the CimForms Text Editor.

Method 1

- a. Click File on the CimForms Text Editor menu bar.
- b. Select Open.

Method 2

Click the Open button on the CimForms Text Editor toolbar.

Method 3

Press Ctrl+O on the keyboard.

An Open dialog box opens.

2. Open the <Project Name>\CIMForms\CCCharFiles\Working folder.

3. Select a file type in the **Files of type** field.

Options are:


ASCII Forms (* .ccf)	Files that were created and saved or were edited and saved as *.ccf files in the in the CimForms Text Editor.
All Files (*.*)	Text files that were saved with a text extension other than .ccf, e.g. .txt

4. Select a .ccf or another text file that contains the appropriate control character tokens.
5. Click Open.

The selected file opens in the CimForms Text Editor.

Step 2.2. Write a Control Character Token File

1. Name of the token (unique name for this device).
2. Actual control characters.
3. Help string/description.


 **Note:** The control character token names must be unique in the:

- Control Character Token file
- Among the names of any general-purpose token, data token, conditional token, solve name, Tracker attribute name or extended tracker attribute name.

The control characters that you use for each file depend on the selected printer.

You should be able to find the specific control character syntax in your printer documentation.

Following is an excerpt from a Control Character Token file for a Printronix printer.

 **Note:** The same CimForm Text Editor [functionality \(page 952\)](#) is available when you write a Control Character Token file as when you write an ASCII form. After a Control Character set is defined for a device, it is linked to an ASCII Form before the form is published to the Tracker project.

Step 2.3 Save the Control Character Token File

1. Do one of the following.

Save an existing file

Method 1

- a. Click File on the CimForms Text Editor menu bar.
- b. Select Save.

Method 2

Click the Save button on the CimForms Text Editor toolbar.

Method 3

Press Ctrl+S on the keyboard.

Result: The changes made to the existing file are saved. This is all that is required for this option.

Save a new file

Choose any of the three methods you use to save a new file.

Result: The Save As dialog box opens.

Save an existing file with a new name and/or extension

- a. Click File on the CimForms Text Editor menu bar.
- b. Select Save as.

Result: The Save As dialog box opens.

2. Continue to save the new file or save an existing file with a new name and/or extension as follows.
 - a. Open the <Project Name>\CIMForms\CCharFiles\Working folder.
 - b. Enter a name for the file in the **File name** field.
 - c. Save the file with a .ccf extension.

Note: You can save the file with another text file extension, e.g. .txt, and then save it as a .ccf file when you add/publish it for the Broadcast project.

The .ccf extension is required if you want to use it as a Broadcast Control Character Token file.

- a. Click Save.

The form is saved; the name you entered in the File name field displays on the form's title bar.

Step 2.4. Compile the Control Character Token File

1. Right-click control Character Token files in the TrackerCfg_UI left pane.
2. Select Compile on the popup menu.

The Control Character Token File Compile dialog box opens.

3. Right-click the button to the right of the **Control Character Token File** field.

The Select the Control Character Token File dialog box opens.

4. Open the <Project Name>\CIMForms\CCharFiles\Working folder.
5. Select the file you want to compile.

Note: File types to select from are:

CimForms Control Character Token Files (*.ccf)	Files that were created and saved or were edited and saved as *.ccf files in the in the CimForms Text Editor.
All Files (*.*)	All text files. Note: Text files that do not conform to the Control Character Token file requirements will be compiled; the content will simply be reported as errors.

6. Click Open.

The path and file name display in the **Control character Token File** field in the Control Character Token File Compile dialog box.

7. Click OK.

Broadcast compiles the Control Character Token file and reports any errors that it finds.

If there are no errors a message displays telling you that the compilation was successful.

Step 2.5. Add/Publish the Control Character Token File

Step 2.5. Add/Publish the Control Character Token File

1. Right-click Control Character Token Files in the TrackerCfg_UI left pane.
2. Select Add/Publish on the Popup menu.

The Add / Update Control Character Token File dialog box opens.

3. Click the Open button to the right of the Control **Character Token File** field.

The Select the Control Character Token File dialog box opens.

4. Open the <Project Name>\CIMForms\CCharFiles\Working folder.

5. Select the file you want to add/publish for Broadcast.

6. Click Open.

The path and file name display in the **Control character Token File** field in the Add / Update Control Character Token File dialog box.

7. (Optional) Enter a different name to save the selected file as follows.

a. Check Save As.

The Save As field is enabled.

a. Enter a new name in the **Save As** field.

8. Click OK.

The Control Character Token File Properties dialog box opens.

9. The Control Character Token File Properties dialog box provides the following information.

rect 12, 29, 248, 42 [\(page 944\)](#)

rect 12, 48, 248, 65 [\(page 944\)](#)

rect 12, 70, 412, 133 [\(page 944\)](#)

rect 12, 136, 343, 154 [\(page 944\)](#)

rect 12, 156, 247, 172 [\(page 944\)](#)

Field	Description
Control Character Token File Name	Name of the selected file or of the name entered in the Save As field.
Control Character Token File Version	Count of the number of times the file has been published.
Control Character Token File Description	Description that will help users know the content and intended use of the added file.
Published Date and Time	Last time the file was published.
Published User	Windows logged in User ID

10. Click OK.

Result: When you add/publish a Control Character Token file the Broadcast:

- a. Adds the file to the list of published files in the TrackerCfg_UI right pane, with the date and time of its publication. It can now be associated with one or more ASCII forms.
- a. Saves the file in the <Project Name>\CIMForms\CCharFiles\prod folder.

11. You can update or delete a published Control Character Token file.

Option 2.5.1 (page 945)	Update a published Control Character Token file.
Option 2.5.2 (page 946)	Delete a published Control Character Token file.

Option 2.5.1. Update Published Control Character Token Files

1. Expand CimForms in the TrackerCfg_UI.
2. Right-click Control Character Token Files.
3. Select View from the Popup menu.

The published Control Character token files display in the right pane.

4. Right-click the file you want to update.
5. Select Update from the Popup menu.

An add / Update Control Character Token File dialog box displays the name of the file in the Working folder that was used to publish the selected file.

6. Do one of the following.

Use the current version of the file that was used before.

Click OK.

Use a different file

- a. Click the Popup menu button to the right of the **Control Character Token File** field.
- b. Find the CimForms\CCharFiles\Working folder.
- c. Select the file you want to use.

The selected file name and path will display in the **Control Character Token File** field.

- a. Click OK.

A Control Character Token File Properties dialog box displays.

7. Enter the latest information in the **Control Character Token File Description** field.
8. Click OK.

Broadcast does the following:

- Updates the selected file in the CimForms\CCharFiles\prod folder.
- Creates backup copies of the updated form in the CimForms\CCharFiles\old folder.

Option 2.5.2. Delete a Published Control Character Token File

1. (If the CIMPLICITY project is running) click the Stop Project button on the Workbench toolbar.
2. Open the TrackerCfg_UI.
3. Expand the CimForms folder.
4. Right-click Control Character Token Files.
5. Select View from the Popup menu.

The published Control Character token files display in the right pane.

6. Right-click the file you want to delete in the TrackerCfg_UI right pane.
7. Select Delete from the Popup menu.

A message asks you to confirm deletion.

8. Click Yes.

A message asks you if you want to delete all backup files.

9. Click:

Yes	To delete the published and back up files.
No	To delete only the published file.

Step 3. Create an ASCII Form

Step 3. Create an ASCII Form

Step 3.1 (page 947)	Open an ASCII form.
Step 3.2 (page 948)	Design the ASCII form.
Step 3.3 (page 953)	Save the ASCII form

Step 3.4 (page 954)	Print the ASCII form configuration.
-------------------------------------	-------------------------------------

Step 3.1. Open an ASCII Form

Step 3.1. Open an ASCII Form

You can create a new file or edit an existing ASCII form in the CimForms Text Editor.

Note: When the either the new or updated file is ready they will be compiled and added/published to be available for Broadcast use.

Option 3.1.1 (page 947)	Open a new ASCII Form file.
Option 3.1.2 (page 948)	Open an existing ASCII Form file.

Option 3.1.1. Open a New ASCII Form

1. Do one of the following in the CimForms Text Editor.

Method 1

- a. Click File on the CimForms Text Editor menu bar.
- b. Select New.

Method 2

Click the New button on the CimForms Text Editor toolbar.

Method 3

Press Ctrl+N on the keyboard.

A New dialog box opens.

2. Select ASCII Form.

A blank ASCII form opens.

3. (Optional) Click File>Page setup on the CimForms Text Editor menu bar.

4. Change the page's default margins.

Notes

- Measurements are in millimeters.
- You can also change measurements for an existing file.

5. Click OK to close the Page Setup dialog box.

The file is ready for writing and editing.

Option 3.1.2. Open an Existing ASCII Form

1. Do one of the following in the CimForms Text Editor.

Method 1

A. Click File on the CimForms Text Editor menu bar.

B. Select Open.

Method 2

Click the Open button on the CimForms Text Editor toolbar.

Method 3

Press Ctrl+O on the keyboard.

An Open dialog box opens.

2. Open the <Project Name>\CIMForms\ASCIIForms\Working folder.

3. Select a file type in the **Files of type** field.

Options are:

ASCII Forms (*.caf)	Files that were created and saved or were edited and saved as *.caf files in the in the CimForms Text Editor.
All Files (*.*)	Text files that were saved with a text extension other than .caf, e.g. .txt

4. Select a .caf or another text file that contains the appropriate ASCII form tokens.

5. Click Open.

The selected file opens in the CimForms Text Editor.

Step 3.2. Design the ASCII Form

Step 3.2. Design the ASCII Form

Step 3.2.1 (page 949)	Use pre-defined tokens.
Step 3.2.2 (page 952)	Display selected CimForms Text Editor tools.

Step 3.2.1. Use Pre-Defined Tokens

Step 3.2.1. Use Pre-Defined Tokens for ASCII Forms

Tokens in any of the following categories can be used in ASCII forms.

Option 3.2.1.1 (page 949)	General purpose tokens.
Option 3.2.1.2 (page 950)	Data tokens.
Option 3.2.1.3 (page 951)	Conditional tokens.

Option 3.2.1.1 General Purpose Tokens

General-purpose tokens have a constant value in ASCII forms.

General-purpose token types include:

Token Type	Use To
//	Create a comment Line. All text following a // token until the end of the line will be ignored.
TEXT	Enter hard coded text. Example <code>TEXT "Action Company"</code>
CONCHAR	Define the device specific control sequence. Example <code>CONCHAR "^MAH^^"</code>
FORMNAME	Name the form. No other data is associated with this token. The form is stored in a file with the same name.
DATETIME	Print the current date and time All current date and time formats supported by windows are available. Example <code>DATETIME "Today is %A, day %d of %B in the year %Y.\n"</code> The output will be Today is Tuesday, day 03 of May in the year 2003.
ASCIICH	Sends the non-printable Control characters that you cannot type in a text file to the printer. Example <code>ASCIICH 13</code> Note: 13 is a decimal value of the Carriage return for a Printronix printer.
SEQNUM	Print the sequence number on the ASCII form. This sequence number is generated on a per device basis.
User Defined Tokens	Add the control character tokens that are specific for your printer in the Control Character Token file. When the Control Character Token file is assigned to an ASCII form they will be used.

Example

Option 3.2.1.2. Data Tokens

Data tokens are dynamic tokens that are replaced by production data during runtime.

The 3 types of data tokens that are available for use in ASCII forms are:

1. Tracker attributes.
2. Tracker extended attributes.
3. Solve results.
 - Data token types.
 - Sub-string description.
 - Data token example

Data token types are:

Token Type	Description
PRTATTR	Tracker attribute data token. We were looking at two different token names, but in the product the user will have only one token name. Example <code>PRTATTR <VINNO></code> Where <code><VINNO></code> is the name of a Tracker attribute defined in the Tracker model.
PRTEXATTR	Tracker Extended attribute data token. Example <code>PRTEXATTR <SHORTVIN></code> Where <code><SHORTVIN></code> is the name of a Tracker extended attribute, which is defined in the Tracker model.
SOLVE	Solve data token representing the result of a value solve. Example <code>SOLVE <SolveName:SolveMethod></code> Example <code>SOLVE <\$GETVALA\$;getvalues></code> Where <code>\$GETVALA\$</code> is solve name and <code>getvalues</code> is a solve method.
POINTID	Point ID data token. Example <code>POINTID <\$PROJECT></code> Where <code>\$PROJECT</code> is the Point Id , which is a defined Point in the Workbench.

Sub-string description. The data tokens types can also be used as follows: `XXXX yyy , a , b` where:

Code	Description
XXXX	Any of the four data tokens <ul style="list-style-type: none"> • PRTATTR • PRTEXATTR • SOLVE • POINTID.
yyy	Any of the following. <ul style="list-style-type: none"> • Tracker Attribute name • Extended tracker attribute name • Query name • Point ID
a	(Optional) starting character If <code>a</code> is not given, the whole string is returned.

Code	Description
b	(Optional) count of characters to extract starting from a. If b is not given, then starting from a a <code>while</code> string is returned.

The user cannot add a List Solve to the ASCII forms. A Boolean solve can be used with conditional tokens.


Data token example

Option 3.2.1.3. Conditional Tokens

Conditional tokens enable the user to add if-then-else conditionals in ASCII forms.

The `IF` conditional token is a combination of tokens as shown below.

Conditional	Description
Basic	
<code>IF <condition></code>	The condition part of the IF statement must contain a Boolean solve (testItem) which returns TRUE or FALSE from the solve engine.
<code>THEN <Statement 1></code>	Must use a new line.
<code>ELSE <Statement 2></code>	(ELSE is optional) If used, ELSE and <Statement 2> must each use a new line.
	Must use a new line.
<code>ENDIF</code>	Must use a new line.
	Example <code>IF <condition> THEN<statement1> - (set of other general purpose tokens or data purpose tokens) ELSE <statement2> - (set of other general purpose tokens or data purpose tokens) 1> ENDIF</code>
Optional	
<code>AND <Condition N></code>	Must use a new line. Any number of Boolean solves may be used in the condition part.
<code>OR <Condition N></code>	Must use a new line. Any number of Boolean solves may be used in the condition part.
	<pre>Example IF <SOLVENAME1> OR <SOLVENAME2> THEN TEXT "RED" ELSE TEXT "BLACK" ENDIF</pre>


 **Note:** You can create nested conditional tokens; there is no limit on number of nested `if` statements.

Example: Basic Conditional

Step 3.2.2. Display Selected CimForms Text Editor Tools

1. Click View on the WYSIWYG Form Editor menu bar.
2. Select either of the following.
 - Toolbar, File and Edit menus
 - Status bar

Toolbar, File and Edit Menus

 **Note:** The [WYSIWYG Form Editor \(page 970\)](#) also has the standard menu (and toolbar) items that are available to the CimForms Text Editor, e.g. Print, Copy, Cut, Paste.

circle 18, 34, 14 [\(page 952\)](#)

circle 41, 34, 14 [\(page 952\)](#)

circle 66, 34, 14 [\(page 952\)](#)

circle 90, 34, 14 [\(page 952\)](#)

circle 118, 34, 14 [\(page 952\)](#)

circle 142, 34, 14 [\(page 953\)](#)

circle 167, 34, 14 [\(page 953\)](#)

circle 195, 34, 14 [\(page 953\)](#)

circle 219, 34, 14 [\(page 953\)](#)

circle 248, 34, 14 [\(page 953\)](#)

circle 273, 34, 14 [\(page 953\)](#)

circle 295, 34, 14 [\(page 953\)](#)

circle 319, 34, 14 [\(page 953\)](#)

circle 348, 34, 14 [\(page 953\)](#)

circle 371, 34, 14 [\(page 953\)](#)

circle 396, 34, 14 [\(page 953\)](#)

circle 417, 34, 14 [\(page 953\)](#)

	Button	Menu	Menu Selection	Opens
A	New file	File	New	
B	Open file		Open	
C	Save file		Save	
D	Print file		Print	
			Print Preview	
			Page Setup	Page Setup dialog box
			Print Setup	Print Setup dialog box.
E	Cut	Edit	Cut	

F	Copy		Copy	
G	Paste		Paste	
			Delete	
			Select All	
H	Undo		Undo	
I	Redo		Redo	
J	Find		Find	Find dialog box.
			Go to line number	Go To dialog box.
K	Find next		Find next	
L	Find previous		Find previous	
M	Replace		Replace	Replace dialog box
			Read only	
N	Add/delete bookmark(s)			
			Bookmarks>Toggle bookmark	
O	Next bookmark		Bookmarks>Next bookmark	
P	Previous bookmark		Bookmarks>Previous bookmark	
Q	Clear bookmark(s)		Bookmarks>Clear bookmark(s)	

 **Note:**

- Click File>Close to close a selected form.
- Click File>Exit to close the Editor.
- Select an object and Click Edit>Delete to delete it.
- Click Edit>Select All to select all objects in the form.

Status Bar

Check Status Bar to display the status bar.

Step 3.3. Save the ASCII Form

1. Do one of the following.

Save an existing file

Method 1

- a. Click File on the CimForms Text Editor menu bar.
- b. Select Save.

Method 2

Click the Save button on the CimForms Text Editor toolbar.

Method 3

Press Ctrl+S on the keyboard.

Result: The changes made to the existing file are saved. This is all that is required for this option.

Save a new file

Choose any of the three methods you use to save a new file.

Result: The Save As dialog box opens.

Save an existing file with a new name and/or extension

- a. Click File on the CimForms Text Editor menu bar.
- b. Select Save as.

Result: The Save As dialog box opens.

2. Continue to save the new file or save an existing file with a new name and/or extension as follows.
 - a. Open the <Project Name>\CIMForms\ASCIIForms\Working folder.
 - b. Enter a name for the file in the **File name** field.
 - c. Save the file with a .caf extension.

Note: You can save the file with another text file extension, e.g. .txt, and then save it as a .caf file when you add/publish it for the Broadcast project.

The .caf extension is required if you want to use it as a Broadcast ASCII Form file.

3. Click Save.

The form is saved; the name you entered displays on the form's title bar.

Step 3.4. Print the ASCII Form Configuration

You can print the form configuration in the CimForms Text Editor. This feature does not test printing data. It can aid you in determining if the form layout is clear and in a logical order.

- Preview the printed form configuration.
- Print the form configuration.

Preview the printed form configuration

1. Click File on the CimForms Text Editor file menu.
2. Select Print Preview.

The CimForms Text Editor Print Preview window opens.

Navigation tools are as follows.

A	Toolbar	
	Print	Print what you are viewing (configuration).
	Next Page	Go to the next page in the Preview window.
	Prev Page	Go to the previous page in the Preview window.
	One Page/Two Page	Toggle between a one and two page display if the ASCII configuration exceeds one page. Note: If you zoom in to enlarge the display, the One Page/Two Page button is unavailable.
	Zoom In	Make the text and graphics in the document larger.
	Zoom Out	Make the test and graphics in the document smaller.
	Close	Close the Print Preview window.
B	Status bar	
	Page	Page in the form that is displaying in Print Preview.
	Ln N, Col N	Position of cursor in the Print Preview window.

Print the form configuration

3. Do one of the following in the CimForms Text Editor.

Method 1

Click the Print button on the CimForms Text Editor toolbar.

Method 2

- a. Click File on the CimForms Text Editor menu bar.
- b. Select Print.

Method 3

Press Ctrl+P on the keyboard.

A Print dialog box opens when you use any method.

4. Make available selections.
5. Print the form configuration.

Step 4. Compile an ASCII Form

1. Right-click **ASCII Forms**.
2. Select Compile on the popup menu.

The ASCII Form Compile dialog box opens.

3. Click the Open button to the right of the **ASCII Form** field.

The Select the ASCII Form dialog box opens.

4. Open the <Project Name>\CIMForms\ASCIIForms\Working folder.
5. Select the file you want to compile.

Note: File types to select from are:

CimForms ASCII Form files (*.caf)	Files that were created and saved or were edited and saved as *.caf files in the in the CimForms Text Editor.
All Files (*.*)	All text files. Note: Text files that do not conform to the ASCII Form requirements will be compiled; the content will simply be reported as errors.

6. Click Open.

The path and file name display in the **ASCII Form** field in the ASCII Form Compile dialog box.

7. Select a Control Character Token file from the drop-down list of available token files.
8. Click OK.

Broadcast compiles the ASCII form.

If there are errors, Broadcast:	<ul style="list-style-type: none"> • Reports the errors. • Places error messages in a text file at \$ProjectRoot\CIMForms\ASCIIForms\Errors.txt.
---------------------------------	--

If there are no errors, Broadcast	Displays a message telling you that the compilation was successful.
-----------------------------------	---

Step 5. Add/Publish an ASCII Form

Step 5. Add/Publish an ASCII Form

1. Right-click ASCII Forms in the TrackerCfg_UI left pane.
2. Select Add/Publish on the popup menu.

The Add /Update ASCII Form dialog box opens.

3. Click the Open button to the right of the **ASCII Form** field.

The Select the ASCII Form dialog box opens.

4. Open the <Project Name>\CIMForms\ASCIIForms\Working folder.
5. Select a file to add/publish for Broadcast.

Note: File types to select from are:

CimForms ASCII Form files (*.caf)	Files that were created and saved or were edited and saved as *.caf files in the in the CimForms Text Editor.
All Files (*.*)	All text files. Note: Text files that have been successfully compiled can be selected.

6. Click Open.

The path and file name display in the **ASCII Form** field in the Add /Update ASCII Form dialog box.

The options are as follows.

rect 13, 33, 349, 55 ([page 957](#))

rect 10, 60, 354, 87 ([page 957](#))

rect 11, 93, 367, 151 ([page 957](#))

Option	Description
ASCII Form	Path and file name are automatically filled in when the file is selected in the Select the ASCII Form dialog box. Path and filename can also be typed in.
Control Character Token File	Will be associated with the ASCII form during runtime. The file will provide direction for printer output.
Save Form As	Check to be able to change the name of the form.

	Broadcast saves the file as a .caf file. You do not need to include the file extension.
--	---

7. Click OK.

An ASCII Form Properties dialog box opens.

8. The ASCII Form Properties dialog box provides the following information.

rect 8, 22, 240, 39 ([page 958](#))
 rect 10, 40, 239, 56 ([page 958](#))
 rect 9, 58, 369, 126 ([page 958](#))
 rect 11, 126, 229, 142 ([page 958](#))
 rect 11, 143, 177, 160 ([page 958](#))
 rect 9, 159, 306, 178 ([page 958](#))
 rect 9, 176, 237, 197 ([page 958](#))

Field	Description
Form Name	Name of the selected file or of the name entered in the Save As field.
Form Version	Count of the number of times the file has been published.
Form Description	Description that will help users know the content and intended use of the added file.
Control Character Token File	File associated with the ASCII form to provide direction for the printer.
Control Character Token File Version	Count of the number of times the associated Control Character Token file has been published.
Published Date and Time	Last time the file was published.
Published User	Windows logged in User ID

9. Click OK.

Result: When you add/publish an ASCII form the Broadcast:

- a. Adds the file to the list of published files in the TrackerCfg_UI right pane.
- a. Saves the file in the <Project Name>\CIMForms\ASCIIForms\prod folder.

10. You can update or delete a published form.

Option 5.1 (page 958)	Update a published ASCII form.
Option 5.2 (page 959)	Delete a published ASCII form.

Option 5.1. Update Published ASCII Forms

1. Expand CimForms in the TrackerCfg_UI.
2. Right-click ASCII Forms.
3. Select View from the Popup menu.

The published forms display in the right pane.

4. Right-click the form you want to update.
5. Select Update from the Popup menu.

An add / Update CimForms ASCII Forms dialog box displays the name of the file in the Working folder that was used to publish the selected file.

6. Do one of the following.

Use the current version of the file that was used before.

Click OK.

Use a different file

- a. Click the Popup menu button to the right of the **ASCII Form** field.
- b. Find the CimForms\ASCIIForms\Working folder.
- c. Select the file you want to use.

The selected file name and path will display in the **ASCII Form** field.

- a. Click OK.

An ASCII Form Properties dialog box displays.

7. Enter the latest information in the **Form Description** field.
8. Click OK.

Broadcast does the following:

- Updates the selected file in the CIMForms\ASCIIForms\prod folder.
- Creates backup copies of the updated form in the CIMForms\ASCIIForms\old folder.

Option 5.2. Delete a Published ASCII Form

1. (If the CIMPLICITY project is running) click the Stop Project button on the Workbench toolbar.

2. Open the TrackerCfg_UI.
3. Expand the CimForms folder.
4. Right-click ASCII forms in the TrackerCfg_UI left pane.
5. Select View from the Popup menu.
6. Right-click the form you want to delete in the TrackerCfg_UI right pane.
7. Select Delete from the Popup menu.

A message asks you to confirm deletion.

8. Click Yes.

A message asks you if you want to delete all backup files.

9. Click:

Yes	To delete the published and back up files.
No	To delete only the published file.

Step 6. Test Print an ASCII Form

1. Click the Start Project button on the Workbench toolbar.
2. Open the TrackerCfg_UI.
3. Expand the CimForms folder in the TrackerCfg_UI left pane.
4. Right-click ASCII Forms.
5. Select Test from the Popup menu.


An ASCII Form Test dialog box opens.

6. Fill in the fields as follows.

Field	Description
ASCII Form	Form in the CimForms\WYSIWYGForms\Working folder that will be tested.
Item ID	Item whose data the form will print.
Device To test	Printer that will print the form.

7. Click OK.

The form will print the data for the selected item in the configured data source fields.

 **Note:** Messages will alert you to possible problems; if the device is not configured correctly the form will not print.

Sample Forms for ASCII File

Sample Forms for ASCII File

- Sample Control Character Token file.
- Sample ASCII form.

Sample Control Character Token File

ASCIICH NUL	0
ASCIICH SOH	1
ASCIICH STX	2
ASCIICH ETX	3
ASCIICH EOT	4
ASCIICH ENQ	5
ASCIICH ACK	6
ASCIICH BEL	7
ASCIICH BS	8
ASCIICH HT	9
ASCIICH LF	10
ASCIICH VT	11
ASCIICH FF	12
ASCIICH CR	13
ASCIICH SO	14
ASCIICH SI	15
ASCIICH DLE	16
ASCIICH DC1	17
ASCIICH DC2	18
ASCIICH DC3	19
ASCIICH DC4	20
ASCIICH NAK	21
ASCIICH SYN	22
ASCIICH ETB	23
ASCIICH CAN	24
ASCIICH EM	25
ASCIICH SUB	26
ASCIICH ESC	27
ASCIICH FS	28
ASCIICH GS	29
ASCIICH RS	30
ASCIICH US	31
ASCIICH SPACE32	

```

ASCIIICH DQUOTE      34
ASCIIICH SQUOTE      39
ASCIIICH      DEL      127
SAMPLE      "SAMPLE Company"
ADDRESS1     "New York"
ADDRESS2     "USA"

```

Sample ASCII Form

```

TEXT      "~NORMAL"
CR
TEXT      "~CREATE;SAMPLE;390"
CR
TEXT      "HDUP;2;37"
CR
TEXT      "BOX"
CR
TEXT      "2;3.5;9;30;35"
CR
TEXT      "STOP"
CR
TEXT      "HORZ"
CR
TEXT      "1;14.5;9;35"
CR
TEXT      "1;19.5;9;35"
CR
TEXT      "1;24.5;9;35"
CR
TEXT      "STOP"
CR
TEXT      "CORNER"
CR
TEXT      "2;4;11;9;33;1.2;2"
CR
TEXT      "2;9.6;11;13.6;33;1.2;2"
CR
TEXT      "STOP"
CR
TEXT      "ALPHA"
CR
TEXT      "4.8;12;0;0;*FROM:*"
CR
TEXT      "6.3;12;2;2;*"
FORD
TEXT      "*"
CR
TEXT      "C15;7.3;16;0;0;*"
ADDRESS1
TEXT      "*"
CR
TEXT      "C15;8.1;16;0;0;*"

```

```
ADDRESS2
TEXT " * "
CR
TEXT "10;12;0;0;*COLORs*"
CR
TEXT "14.8;11;0;0;*DDMMYYYY*"
CR
TEXT "19.8;11;0;0;*HHMMSS*"
CR
TEXT "24.8;11;0;0;*WOY*"
CR
TEXT "STOP"
CR
TEXT "HDUP;OFF"
CR
TEXT "BARCODE"
CR
TEXT "C3/9;H7;BF1;8;DARK;15.1;10.4"
CR
TEXT "PDF;O"
CR
TEXT "STOP"
CR
TEXT "BARCODE"
CR
TEXT "C3/9;H7;BF2;8;DARK;20;10.4"
CR
TEXT "PDF;O"
CR
TEXT "STOP"
CR
TEXT "BARCODE"
CR
TEXT "C3/9;H7;BF3;8;DARK;25.5;10.4"
CR
TEXT "PDF;O"
CR
TEXT "STOP"
CR
TEXT "ALPHA"
CR
TEXT "AF1;20;10.8;12;0;0"
CR
TEXT "AF2;20;11.8;12;0;0"
CR
TEXT "AF3;21;12.8;12;0;0"
CR
TEXT "STOP"
CR
TEXT "END"
CR
TEXT "~EXECUTE;SAMPLE"
CR
```

```

TEXT      "~AF1 ; * "
PRTATTR   COLOR1
TEXT      " * "
CR
TEXT      "~AF2 ; * "
PRTATTR   COLOR2
TEXT      " * "
CR
TEXT      "~AF3 ; * "
PRTATTR   COLOR3
TEXT      " * "
CR
TEXT      "~BF1 ; * "
DATETIME  "%d%m%Y"
TEXT      " * "
CR
TEXT      "~BF2 ; * "
DATETIME  "%H%M%S"
TEXT      " * "
CR
TEXT      "~BF3 ; * "
DATETIME  "%U"
TEXT      " * "
CR
TEXT      "~NORMAL "
CR

```

WYSIWYG Forms

WYSIWYG Form Configuration

Step 1 (page 965)	Open a WYSIWYG Form Editor.
Step 2 (page 965)	Open a WYSIWYG form.
Step 3 (page 967)	Save the WYSIWYG form.
Step 4 (page 970)	Work with the WYSIWYG Form Editor tools.
Step 5 (page 979)	Configure text objects for the WYSIWYG form.
Step 6 (page 996)	Configure Broadcast Bar Code labels.

Step 7 (page 1047)	Insert graphic objects for the WYSIWYG form.
Step 8 (page 1062)	Organize objects to create an easy-to-read form.
Step 9 (page 1068)	Compile a WYSIWYG form.
Step 10 (page 1068)	Test print the WYSIWYG form.
Step 11 (page 1069)	Add/publish a WYSIWYG form.

Step 1. Open the WYSIWYG Form Editor

1. Expand the CimForms folder in the TrackerCfg_UI left pane.
2. Right-click **WYSIWYG Forms**.
3. Select WYSIWYG Editor on the popup menu.

The WYSIWYG Form Editor opens.

Step 2. Open a WYSISYG Form

Step 2. Open a WYSISYG Form

Option 2.1 (page 965)	Open a new WYSIWYG form.
Option 2.2 (page 966)	Open an existing WYSIWYG form.

Option 2.1. Open a New WYSIWYG Form

1. Expand the CimForms folder in the TrackerCfg_UI left pane.
2. Right-click **WYSIWYG Forms**.
3. Select Text Editor on the popup menu.

The WYSIWYG Form Editor opens.

4. Do one of the following.

Method 1

- a. Click File on the WYSIWYG Form Editor menu bar.
- b. Select New.

Method 2

Press Ctrl+N on the keyboard.

A New Form dialog box opens.

5. Enter specifications for the new form as follows.

rect 16, 159, 278, 183 [\(page 966\)](#)

rect 4, 195, 93, 279 [\(page 966\)](#)

rect 100, 200, 285, 277 [\(page 966\)](#)

rect 224, 283, 287, 307 [\(page 966\)](#)

Option	Description	
Size	Size of the form that will be printed. Choices depend on your printer's capability. Common choices are: Letter A4 Legal	
Orientation	Click either	
	Portrait	The long side is top to bottom.
	Landscape	The wide side is top to bottom.
Margins	Margins from the form edge that display on the form during configuration and are adhered to during printing.	
Printer	Button opens a New Form dialog box for selecting and configuring the form's printer if it will be different from the default printer.	
	Default	Computer's default printer.

6. Click OK to close the New Form dialog box.

The new form displays in the WYSIWYG Form Editor window. The form's size, orientation and margins reflect your choices.

Option 2.2. Open an Existing WYSIWYG Form

1. Do one of the following in the WYSIWYG Editor.

Method 1

- a. Click File on the WYSIWYG Text Editor menu bar.
- b. Select Open.

Method 2

Click the Open button on the WYSIWYG Text Editor toolbar.


Method 3

Press Ctrl+O on the keyboard.

An Open dialog box opens.

2. Open the <Project Name>\CIMForms\WYSIWYGForms\Working folder.
3. Select a .cwf file.
4. Click Open.

The selected file opens in the WYSIWYG Editor.


 **Note:** Recently saved files can also be selected on the File menu.

Step 3. Save the WYSIWYG Form

Step 3. Save the WYSIWYG Form

Step 3.1 <i>(page 967)</i>	Confirm Form Properties (New Form)
Step 3.2 <i>(page 969)</i>	Save the WYSIWYG Form

Step 3.1. Confirm Form Properties (New Form)

 **Note:** You cannot change the form's page layout after you create it. It is recommended that you double-check the form's properties before you save it. If you want to change them you can easily create a new form and enter your revised specifications in the New Form dialog box.

It is also a good idea to enter summary information for future reference.

1. Click File on the WYSIWYG Form Editor's menu bar.
2. Select Form Properties...

A Form Properties dialog box opens.

3. Enter and review information on each of the tabs.

- Summary Info
- Layout
- Print options


Summary Info tab

Summary info provides users with the form's administrative details. as follows.

Data	Field Access	Description
Author	Read/write	Person, company or division responsible for creating the form. Entry depends on your company's policy.
Created Date	Read only	Date the form was created.
Modified Date	Read only	Date of the last time the form was modified.
Last saved User ID	Read only	Last Windows login ID of the person who saved the form.
Printer Name or Type	Read/write	Name or type of printer that should be selected for this form.
Revision number	Read only	Number of the last revision.
Description	Read/write	Reference for users and developers describing what the form is and what it is for.
		Maximum Length
		1024 characters

Layout tab

The read-only layout tab displays the choices you made in the New Form dialog box. If you want to change these choices close this form without saving it and create a new form that will reflect your changes.

 **Note:** If you want to change the paper and/or margins, create a new form and copy your configuration to that form.

Print Options

Options on the Print Options tab are as follows.

Option	Description
--------	-------------

Print Upside Down	Checked	Prints the form upside down.
	Clear	Prints the form right-side up.
Missing Data Characters	Missing characters that need to be printed for the form's objects where data cannot be fetched.	
	Default	???
Copies	Read-only	The number of copies that will be printed of each form. The number is selected in the BCO function blocks: Broadcast Form, Send Form to Device, Send WYSIWYG Form, Send WYSIWYG Form, Send WYSIWYG Form By Variable.
		The history file is the only copy of the form that will display a number other than 1, the default value for number of copies. This number is recorded only from entries in either function block. A user can change the number of copies when resending (page 1101) or redirecting (page 1102) selected forms through the Broadcast Queue Manager. However, the temporary number will not display on the Print Options tab in the forms' history files. The number of copies displays for forms as follows.
		Default: 1
		New form: 1
	*	History form: Number of copies printed of the selected form; the file of the printed form is found in the printed forms' History (page 1112) folder. The history file is the only copy of the form that will display a number other than 1.

Step 3.2. Save the WYSIWYG Form

1. Do one of the following.

Save an existing file

Method 1

- a. Click File on the WYSIWYG Editor menu bar.
- b. Select Save or Save as.

Method 2

Click the Save button on the WYSIWYG Editor toolbar.

Method 3

Press Ctrl+S on the keyboard.

Result: The changes made to the existing file are saved. This is all that is required for this option.

Save a new file

Choose any of the three methods you use to save a new file.

Result: The Save As dialog box opens.

Save an existing file with a new name and/or extension

- a. Click File on the WYSIWYG Editor menu bar.
- b. Select Save as.

Result: The Save As dialog box opens.

2. Continue to save the new file or save an existing file with a new name and/or extension as follows.

- a. Open the <Project Name>\CIMForms\WYSIWYGForms\Working folder.
- b. Enter a name for the file in the **File name** field.
- c. Save the file with a .cwf extension.

3. Click Save.

The form is saved; the name you entered displays on the form's title bar.

Step 4. Work with the WYSIWYG Form Editor Tools

Step 4. Work with the WYSIWYG Form Editor Tools

The WYSIWYG Form Editor provides you with several tools that can be used for the form and/or a variety of options.

These tools include:

Option 4.1 (page 971)	Selected WYSIWYG Form Editor view tools.
Option 4.2 (page 973)	WYSIWYG Form Editor Grid
Option 4.3 (page 975)	WYSIWYG Form Editor color tool.
Option 4.4 (page 975)	WYSIWYG Form Editor object defaults.
Option 4.5 (page 978)	Use the WYSIWYG Form Editor auto-scroll feature.
Option 4.6 (page 978)	Duplicate and arrange WYSIWYG forms.

Option 4.1. Selected WYSIWYG Form Editor View Tools

1. Click View on the WYSIWYG Form Editor menu bar.
2. Select any of the following.
 - Toolbars
 - Status bar
 - Grid Visible
 - Zoom (screen magnification)

Toolbars

Check any toolbar listed on the extended menu that you want to display.

The toolbars and buttons are as follows (click the menu to go to the toolbar description).

rect 203, 45, 292, 65 ([page 971](#))

rect 203, 65, 293, 83 ([page 971](#))

rect 204, 84, 293, 103 ([page 972](#))

rect 205, 102, 291, 125 ([page 972](#))

rect 204, 125, 290, 143 ([page 972](#))

	Toolbar		Description
	Standard		
		A	New form
		B	Open
		C	Save
		D	Print
		E	Print Preview
		F	Test Print
		G	Form Properties
Up (page 971)		H	Help
	Edit		
		A	Cut
		B	Copy
		C	Paste
		D	Delete
		E	Properties

		F	Undo
Up (page 971)		G	Redo
	View		
		A	Grid Visible/Not Visible
		B	Zoom In
		C	Percent Zoom
Up (page 971)		D	Zoom out
	Draw		
		A	Select object
		B	Line
		C	Rectangle
		D	Text
		E	Bar code
Up (page 971)		F	Image tool (Insert image file)
	Action		
		A	Align left
		B	Align top
		C	Align right
		D	Align bottom
		E	Group
		F	Ungroup
		G	Bring to front
		H	Send to back
		I	Same width
		J	Same height
		K	Same width and height

Status bar

Check Status Bar to display the status bar.

The status bar displays the:

- Percent zoom.
- NUM if the Num Lock key is on.

Grid

Check Grid Visible to display the grid.

Note: Press Ctrl+G to toggle the grid on and off.

A	Grid Visible checked
B	Grid Visible cleared

Zoom

Check the percent magnification in which you want the edit form to display

Option 4.2. WYSIWYG Form Editor Grid

You can use a configurable grid to help place and align objects on your WYSIWYG form.

- Display or clear the grid.
- Configure the grid.

Display or clear the grid

1. Click View on the WYSIWYG Form Editor menu bar.
2. Check Grid Visible to display the grid.

Note: Press Ctrl+G to toggle the grid on and off.

A	Grid Visible checked
B	Grid Visible cleared

Configure the grid

3. Click Tools>Options on the WYSIWYG Form Editor toolbar.

The Options dialog box opens.


4. Select the Grid tab.

Configuration options are as follows.

rect 38, 48, 297, 127 ([page 974](#))

rect 35, 127, 296, 189 ([page 974](#))

rect 36, 190, 295, 254 ([page 974](#))

	Field	Description	
	Grid Style	Dot Grid	Displays the grid as dots.
		Line Grid	Displays the grid as lines.
		Draw Center Line	Displays horizontal and vertical lines in the center of the form.
Up (page 974)	Grid Size	H Spacing	Spacing between horizontal dots or lines.
		Range	10-400
		V Spacing	Spacing between vertical dots or lines.
		Range	10-400
		Examples	
			Dots 10 X 10
			Lines 50 X 50
			Lines 250 X 10
Up (page 974)	Snap to Grid	Snap Position to Grid	When you release the left mouse button or use the arrow keys to move an object, the object snaps to the nearest horizontal/vertical grid lines/dots.
		Snap Size to Grid	When you release the left mouse button after increasing or decreasing the object size, the object edges snap to the nearest horizontal/vertical grid lines/dots.
		Example	
			Item before snapped.
			Item after snapped (position and size).
		 Note: Toggle the Alt key to disable/enable Snap to Grid as follows:	
		Press Alt when:	Result
		Snap to Grid is checked	Disables the functionality. Objects do not snap to the grid.

		Snap to Grid is clear	Enables the functionality. Objects do snap to the grid.
--	--	-----------------------	---

Option 4.3. WYSIWYG Editor Color Tool

1. Click Tools>Colors on the WYSIWYG Form Editor toolbar.

An advanced Color Palette opens.

2. Select a color to find it's .numeric value.

Option 4.4. Work with Pre-Configured Styles on the WYSIWYG Form Editor

Option 4.4. Work with Pre-Configured Styles on the WYSIWYG Form Editor

The WYSIWYG Form Editor enables you to work with pre-defined styles and templates.

Option 4.4.1 (page 975)	Use Default Configurations for WYSIWYG Form Objects
Option 4.4.2 (page 977)	WYSIWYG Form Editor Template

Option 4.4.1. Use Default Configurations for WYSIWYG Form Objects

You can do the following with default WYSIWYG form object configuration:

- Create default configurations.
- Apply default configurations.
- Edit default configurations.

Create default configuration

1. Configure any of the following objects.
 - Text
 - Line
 - Rectangle
 - Bar code
2. Right click each configured object.
3. Select Save as Default Properties on the popup menu.

Result: The following properties associated with the selected object become the default properties.

Object	Default Properties
Text	Border, font
Line	Line
Rectangle	Border/Fill
Bar code	Bar code/Static text for caption

Apply default configurations

Place an object on the WYSIWYG form.

Result: The object's properties are the specified defaults.

Edit default configuration

Do one of the following.

Method 1

4. Configure an object on the WYSIWYG form.
5. Re-open the object's popup.
6. Re-select Save as Default Properties.

Method 2

7. Click Tools>Object Defaults><object> on the WYSIWYG Form Editor menu bar.

An <Object> Defaults dialog box opens for the selected object as follows.

Object	<Object> Defaults Dialog Box	Tab
Line		Line (page 1061)
Rectangle		Border (page 1057)
Text		Border (page 994) Font (page 995)
Bar code		Barcode (page 1007)


Option 4.4.2. Use a WYSIWYG Form Editor Template

You can save and then use any form as a template.

- Save a template.
- Use a template.

Save a template

1. Create default objects on a form that you will save as a template.
2. Save the form.

 **Note:** You cannot edit the template, so it is important to save the form to make it easy to modify the template.

3. Click Tools>Options on the WYSIWYG Form Editor menu bar.

The options dialog box opens.

4. Select the Template tab.

5. Click Save.

A Save As dialog box opens.

6. Enter the template name in the WYSIWYG form's Working directory.

7. Click Save.

The path name and file display in the File Path field.

8. Click OK.

Result: The file can now be used as a template when you create or edit forms.

Use a template

9. Click Tools>Options on the WYSIWYG Form Editor menu bar of your active form.

The options dialog box opens.

10. Select the Template tab.

11. Click Load.

An Open dialog box opens.

12. Find and select the template you want to use.

Note: The template is a .cwt file.

Option 4.5. Use the WYSIWYG Form Editor Auto-Scroll Feature


The WYSIWYG Graphic Editor will auto-scroll if you drag an object off of the visible area.

For example:

A	An item is selected.
B	A wire frame image displays where you are dragging the selected item.
C	The WYSIWYG Editor scrolls to make the form area visible as it follows the item's moving location.

Option 4.6. Duplicate and Arrange WYSIWYG Forms

- Duplicate the active WYSIWYG form.
- Cascade open WYSIWYG forms.
- Tile open WYSIWYG forms.
- Arrange Minimized WYSIWYG forms.

 **Note:** This functionality is also available in the [CimForms Text Editor \(page 938\)](#).

Duplicate the active WYSIWYG form.

Click Window>New on the WYSIWYG form menu bar.

Result: A new window opens with a duplicate of the active form.

The new form is:

- Numbered in an increment of 1 from the original form.
- Has the same name as the original.

Cascade open WYSIWYG forms

Click Window>Cascade on the WYSIWYG form menu bar.

Result: Open windows are diagonally stacked on top of each other with only the title bar displaying.

Tile open WYSIWYG forms.

Click Window>Tile on the WYSIWYG form menu bar.

Result: Open windows are stacked vertically.

Arrange Minimized WYSIWYG forms

Click Window>Arrange Icons on the WYSIWYG form menu bar.

Result: Minimized windows line up on the bottom of the WYSIWYG Form Editor. The minimized forms will wrap to stay within the visible section of the editor window.

Step 5. Configure Text Objects for the WYSIWYG Form

Step 5. Configure Text Objects for the WYSIWYG Form

Step 5.1 (page 979)	Place a text object on the WYSIWYG form.
Step 5.2 (page 980)	Open the Text Object Properties dialog box.
Step 5.3 (page 980)	Select a data source for the text object.
Step 5.4 (page 992)	Format the text object.

Step 5.1. Place a Text Object on the WYSIWYG Form

1. Do one of the following.

Method 1

- a. Click Draw on the WYSIWYG Text Editor menu bar.
- b. Select Text.

Method 2

Click the Text button on the WYSIWYG Text Editor toolbar.

Method 3

- a. Right-click the WYSIWYG screen.

b. Select Text.

2. Drag your mouse on the form where you want to place the text.

A text box is placed where you drag.

i Tip: : You can cut and paste or copy and paste a text object from one WYSIWYG form into another (as described for an [image object \(page 1049\)](#)). The source text configuration is copied to the target.

Step 5.2. Open the Text Object Properties Dialog Box

1. Select the text box that you just placed in the WYSIWYG form.

2. Do one of the following.

Method 1

Double-click the text box.

Method 2

a. Click Edit on the WYSIWYG Text Editor menu bar.

b. Select Properties.

Method 3

Click the Properties button on the WYSIWYG Text Editor toolbar.

Method 4

Press Alt+Enter on the keyboard.

The Text Object Properties dialog box opens when you use any of the methods.

Step 5.3. Select a Data Source for the Text Object

Step 5.3. Select a Data Source for the Text Object

Option 5.3.1 (page 981)	Enter static text as the text object data source.
Option 5.3.2 (page 981)	Select a system option as the text object data source.

Option 5.3.3 (page 984)	Select an external option as the text object data source.
Option 5.3.4 (page 987)	Use a script to define the text object data source.

Option 5.3.1. Enter Static Text as the Text Object Data Source

Static text displays as the default text for the form.

You can select this option alone or in addition to any of the other data source options. You can also use it to provide headings and descriptions for dynamic data.

1. Select the Data Source tab in the Text Object Properties dialog box.
2. Click Static.
3. Enter the text in the **Text** box that you want to display on the form.

If Static is the only data source selected, the text you enter will display on the printed form.

4. Click Apply.
5. (Optional) Revise the font, border and position specifications to accommodate the text requirements.

Note: This can be done during any stage of the form design.

6. Click OK.

Text displays on the form in the currently specified format.

Option 5.3.2. Select a System Option as the Text Object Data Source

1. Select the Data Source tab in the Text Object Properties dialog box.
2. Click System.
3. The options are as follows.

rect 245, 66, 307, 81 ([page 982](#))
 rect 245, 81, 320, 95 ([page 982](#))
 rect 245, 93, 368, 110 ([page 983](#))

Option	Description
--------	-------------

Form Name	File name	
Date & Time	Inserts the date and time recorded on Broadcast server. Format options are:	
	Selection	Example
	mm/dd	07/11
	mm/dd/yy	07/11/03
	dd/mm/yy	11/07/03
	dd/mm/yyyy	11/07/2003
	dd-Month	11-July
	dd-Month-yy	11-July-03
	dd-Month-yyyy	11-July-2003
	Month-yy	July-03
	Month-dd-yyyy	July-11-2003
	Month dd	July 11
	Month, yyyy	July, 2003
	dd-Month (Short Month Name)	11-Jul
	dd-Month-yy (Short Month Name)	11-Jul-03
	dd-Month-yyyy (Short Month Name)	11-Jul-2003
	Month-yy (Short Month Name)	Jul-03
	Month dd, yyyy (Short Month Name)	Jul 11, 2003
	Month dd (Short Month Name)	Jul 11
	Month, yyyy (Short Month Name)	Jul, 2003
	Day, Month dd, yyyy	Friday, July 11, 2003
	hh:mm AM/PM	11:13 AM
	hh:mm:ss AM/PM	11:13:37 AM
	HH:mm	11:14
	hh:mm:ss AM/PM	11:14:14 AM
	HH:mm:ss	11:14:55
	mm/dd/yyyy HH:mm	07/11/2003 11:15
	mm/dd/yyyy HH:mm AM/PM	07/11/2003 11:15:46 AM
	Day, Month dd, yyyy HH:mm:ss	Friday, July 11, 2003 11:16:14

		Day, Month dd, yyyy hh:mm:ss AM/PM	Friday, July 11, 2003 11:16:46 AM
	Format	Represents	
	%a	Abbreviated weekday name	
	%A	Full weekday name	
	%b	Abbreviated month name	
	%B	Full month name	
	%c	Date and time representation appropriate for locale	
	%d	Day of month as decimal number (01 - 31)	
	%H	Hour in 24-hour format (00 - 23)	
	%I	Hour in 12-hour format (01 - 12)	
	%j	Day of year as decimal number (001 - 366)	
	%m	Month as decimal number (01 - 12)	
	%M	Minute as decimal number (00 - 59)	
	%p	Current locale's A.M./P.M. indicator for 12-hour clock	
	%S	Second as decimal number (00 - 59)	
	%U	Week of year as decimal number, with Sunday as first day of week (00 - 53)	
	%w	Weekday as decimal number (0 - 6; Sunday is 0)	
	%W	Week of year as decimal number, with Monday as first day of week (00 - 53)	
	%x	Date representation for current locale	
	%X	Time representation for current locale	
	%y	Year without century, as decimal number (00 - 99)	
	%Y	Year with century, as decimal number	
	%z, %Z	Time-zone name or abbreviation; no characters if time zone is unknown	
	%%	Percent sign	
Form Sequence Number		Sequence number of the broadcast; the number is set during runtime. It is useful to insure, for example, that printouts are in the correct order.	

4. Click Static.

5. Enter a brief description of your selection in the text box.

6. Click Apply.
7. (Optional) Revise the font, border and position specifications to accommodate the text requirements.

Note: This can be done during any stage of the form design.

8. Click OK.

The static text entry displays in the editable form. When the form is broadcast, the dynamic selection will replace this text.


Option 5.3.3. Select an External Option as the Text Object Data Source

Option 5.3.3. Select an External Option as the Text Object Data Source

1. Select the Data Source tab in the Text Object Properties dialog box.
2. Click External.
3. Select an external source in the **Type** field.

Option 5.3.3.1 (page 984)	Use Solves for external Broadcast data.
Option 5.3.3.2 (page 986)	Use Tracker attributes for external Broadcast data.
Option 5.3.3.3 (page 987)	Use extended attributes for external Broadcast data.

Option 5.3.3.1. Use Solves for External Broadcast Data

 **Important:** You can open Solve's Expression Browser through the Text Object Properties dialog box. Make sure your project is running if you want to take advantage of this feature.

1. Click External on the Data Source tab in the Text Object Properties dialog box.
2. Select Solve in the **Type** field.
3. Use the Data Source features to select a Solve expression.

Feature	Description
Named Solve / Expression	The selected solve expression that will be dynamically broadcast during runtime. Broadcast uses the active version for the selected expression.

Expression Browser access	You have access to the full functionality of the browser, e.g. browsing, creating new expressions, editing existing expression.
	Opens the Expression Browser. To add an existing expression: Double-click an expression in the Expression Browser. Result: The expression is entered in the Named Solve / Expression field.
New	Opens a New Expression dialog box.
	<p>To add a new expression:</p> <ol style="list-style-type: none"> Enter a unique name in the Name field. Click OK. <p>The Expression Editor opens for the new expression.</p> <ol style="list-style-type: none"> Write a new expression. Click Apply & Activate when you are finished. Click Cancel or the Close button in the Expression Editor. <p>Result: The new expression displays in the Named Solve / Expression field.</p>
Edit	Provides the ability to edit a selected expression.
	If: An expression is in the Named Solve / Expression field.
	Then: The Expression Editor opens with the expression available for editing. To add an edited expression: <ol style="list-style-type: none"> Edit the expression. Click Apply & Activate. Click Cancel or the Close button in the Expression Editor. <p>Result: The form will use the edited version of the expression when it is used by Broadcast. Note: Broadcast will use the active expression. If you choose not to activate the edited version, it will not be used.</p>
	If: No expression is in the Named Solve / Expression field.
	Then: Edit is inactive.
	If: The expression is not in the Solve database.
	Then: A message tells you that the expression does not exist.
Browse	Opens the Expression Browser

- Enter a unique name in the Name field.
- Click OK.
The Expression Editor opens for the new expression.
- Write a new expression.
- Click Apply & Activate when you are finished.
- Click Cancel or the Close button in the Expression Editor.
- Edit the expression.
- Click Apply & Activate.

11. Click Cancel or the Close button in the Expression Editor.

12. Click Static on the Data Source tab.

Note: You can enter static text in addition to selecting system or external data sources. This enables you to describe the selected source so the source can be easily identified.


13. Enter text that describes the source in the **Text** box.

14. (Optional) Revise the font, border and position specifications to accommodate the text requirements.

Note: This can be done during any stage to the form design.

15. Click OK.

The text that was entered as static text displays on the form. The selected Solve expression results will replace this text when the form is broadcast.

 **Important:** A Solve name/expression configured in the form that has multiple return values will result in a Data Fetch error in Broadcast, during runtime.

Option 5.3.3.2. Use Tracker Attributes for External Broadcast Data

1. Click External on the Data Source tab in the Text Object Properties dialog box.

2. Select Tracker in the **Type** field.

3. Enter a Tracker attribute in the **Attribute Name** field, e.g. an attribute listed in the Attribute Maintenance dialog box

4. Click Static on the Data Source tab.

Note: You can enter static text in addition to selecting system or external data sources. This enables you to describe the selected source so the source can be easily identified.

5. Enter text that describes the source in the **Text** box.

6. (Optional) Revise the font, border and position specifications to accommodate the text requirements.

Note: This can be done during any stage to the form design.

7. Click OK.

The text that was entered as static text displays on the form. The value of the selected attribute will replace this text when the form is broadcast.

Option 5.3.3.3. Use Extended Attributes as External Broadcast Data

1. Click External on the Data Source tab in the Text Object Properties dialog box.
2. Select Ext Tracker in the **Type** field.
3. Enter an extended Tracker attribute in the **Attribute Name** field, e.g. an attribute listed on the Attributes tab in a PRT_UI Item dialog box.
4. Click Static on the Data Source tab.

Note: You can enter static text in addition to selecting system or external data sources. This enables you to describe the selected source so the source can be easily identified.

5. Enter text that describes the source in the **Text** box.
6. (Optional) Revise the font, border and position specifications to accommodate the text requirements.

Note: This can be done during any stage of the form design.

7. Click OK.

The text that was entered as static text displays on the form. The value of the selected extended attribute will replace this text when the form is broadcast.

Option 5.3.4. Use a Script for a Text Object Data Source

Option 5.3.4. Use a Script for a Text Object Data Source

1. Click Script on the Data Source tab in the Text Object Properties dialog box.
2. Click Edit.

A WYSIWYG Text Editor opens.

3. Enter the script that you want to apply to the object.

 **guide: Guidelines for the script**

- The Compilation and Execution of the Object script starts and ends in function `Main`,

- `Sub Main` functions are mandatory.
- All the references to the object inside the script use `MyObject`.

Review:

Text object script properties.
Text object script constants.
Text object script date and time formats.
Text object form level script functions and return types.

Note: Menu options and toolbar buttons provide the same functionality as the same items in the CIMPPLICITY Program Editor.

4. Do one of the following to compile the script.
 - Click the Compile button .
 - Click File>Compile on the WYSIWYG Form Editor menu bar.

The script is compiled. Broadcast reports errors or informs you that the script compiled successfully.

5. Close the WYSIWYG Form Editor.

A message asks you if you want to save the script to the object.

6. Click Yes.

The script displays as read-only text in the **Script** box.

Note: You can also click Compile on the Data Source tab to compile the script.

7. Click Static.

8. Enter a brief description of your selection in the text box.

9. Click Apply.

10. (Optional) Revise the font, border and position specifications to accommodate the text requirements.

Note: This can be done during any stage of the form design.

11. Click OK.

The static text entry displays in the editable form. When the form is broadcast, the dynamic selection will replace this text.

Sample Script

```
Sub Main
  Dim strValue as String
  strValue = Form.ItemID
  MyObject.TextValue = strValue
End Sub
```

Text Object Script Properties

Get and Set properties are:

Property Name	Data Type	Description
TextValue	String	The text value that will be used by the object to show the text.
LineColor	Long	Bounding rectangle outline color.
FillColor	Long	Bounding rectangle fill color, this color is the background color of the text value.
FillRectangle	Bool	Fill the text object rectangle or not, if a set is done for FillColor property then this property is set true.
BackgroundColor	Long	Bounding rectangle outline background color, this color is visible only when appropriate line style is selected
LineWidth	Short	Bounding rectangle outline width (0 – 50)
LineStyle	Short	Bounding rectangle outline style (0 – 5)
Bold	Bool	If set to TRUE, the text will be printed in BOLD.
FontName	String	The font to be used can be set using this property.
Italic	Bool	Using the font in Italic format for the text value
Strikeout	Bool	Using the font in strikeout format for the text value
Underline	Bool	Using the font in underline format for the text value
Wraptext	Bool	Wrapping the text at word level is the text value is not fitting in the bounding rectangle
FontSize	Short	Using the font size for the text value (1 – 254)
Justify	Byte	Text value justify inside the bounding rectangle (0 – 8)
TextColor	Long	Text Font Color
FillRectangle	Bool	Fills the background rectangle
ReverseFont	None	Swaps the <code>TextColor</code> and <code>BackgroundColor</code> and sets the <code>FillRectangle</code> as True

Text Object Script Constants

Constants that can be used in the scripts include:

- Text justification.
- Predefined colors.
- Predefined line styles.

Text Justification

TopLeft

TopCenter

TopRight

CenterLeft

CenterCenter

CenterRight

BottomLeft

BottomCenter

BottomRight

Predefined Colors

Black

Red

Blue

Green

Yellow

Fuchsia

Aqua

White

Predefined Line Style

Solid

Dash

Dot

DashDot

DashDotDot

Null

Text Object Script Date and Time Formats

Format	Represents
%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name
%c	Date and time representation appropriate for locale
%d	Day of month as decimal number (01 - 31)
%H	Hour in 24-hour format (00 - 23)
%I	Hour in 12-hour format (01 - 12)
%j	Day of year as decimal number (001 - 366)
%m	Month as decimal number (01 - 12)
%M	Minute as decimal number (00 - 59)
%p	Current locale's A.M./P.M. indicator for 12-hour clock
%S	Second as decimal number (00 - 59)
%U	Week of year as decimal number, with Sunday as first day of week (00 - 53)
%w	Weekday as decimal number (0 - 6; Sunday is 0)
%W	Week of year as decimal number, with Monday as first day of week (00 - 53)
%x	Date representation for current locale
%X	Time representation for current locale
%y	Year without century, as decimal number (00 - 99)
%Y	Year with century, as decimal number

%z, %Z	Time-zone name or abbreviation; no characters if time zone is unknown
%%	Percent sign

Text Object Form Level Script Functions and Return Types

The following functions will return the runtime form level properties so the user can use them in the script.

All the properties:

- Are read only.
- Can be accessed using the syntax:

Form.XXXX where XXXX can be one of the values given below:

Function Name	Return Type	Description
ProjectName	String	Project name where this form is broadcast.
ItemID	String	Item ID for the broadcast.
SequenceNo	String	Sequence Number of the broadcast.
Name	String	WYSIWYG Form name.

Following are functions that can be used to access:

- Tracker attributes
- Extended Tracker attributes
- Solve value

Function	Where
Form.GetTrackerAttr(YYYY)	YYYY is a string passed as a tracker or extended tracker attribute.
Form.GetSolveValue(YYYY)	YYYY is a string passed as a Solve name or expression.

A function that gets the Long value of Color in using RGB is:

Function	Where
RGBColor(Rvalue , Gvalue , Bvalue)	Rvalue is the Red value Gvalue is the Green value Bvalue is the Blue value

Step 5.4. Format a Text Object

Step 5.4. Format a Text Object

Option 5.4.1 (page 993)	Position text on the WYSIWYG form.
Option 5.4.2 (page 994)	Configure the text object border.
Option 5.4.3 (page 995)	Select the text object font.

Option 5.4.1. Position Text on the WYSIWYG Form

Methods to change the size and position of the text box on the WYSIWYG form include:

- Use the mouse to change the text box location.
- Use the Text Object Properties dialog box.

Use the mouse to change the text box location

1	Click and hold down the right-mouse button anywhere in the text box.
	The cursor changes to cross hairs when the text box is ready to be moved.
2	Hold down the right-mouse button as you drag the cursor on the screen.
	A dotted line frame displays the destination of the box as you drag the cursor.
3	Release the right-mouse button when the box is in the correct location.
	The text box moves to the selected location.

Use the Text Object Properties dialog box

1. Select the Position tab in the Text Editor Properties dialog box.

The field entries reflect the current position of the text box.

2. Edit any of the positions as follows.

rect 51, 83, 199, 120 ([page 993](#))

rect 49, 124, 202, 160 ([page 994](#))

rect 49, 163, 199, 205 ([page 994](#))

rect 50, 208, 200, 247 ([page 994](#))

Field	Co-ordinate/Value	Description
Left	X co-ordinate of the starting position	First clicked the mouse after selecting the text object.

Top	Y co-ordinate of the starting position	First clicked the mouse after selecting the text object.
Width	Delta X value	Difference between the starting position X value and the ending position X value of the rectangle, This defines the width of the text box object.
Height	Delta Y value	Difference between the starting position Y value and the ending position Y value of the rectangle. This defines the height of the text box object.

[Up \(page 993\)](#)

3. Do one of the following.
 - a. Click Apply to apply your entries and continue to configure other text features.
 - b. Click OK to close the Text Object Properties dialog box and view your changes.

Option 5.4.2. Configure the Text Object Border

1. Select the Border tab in the Text Object Properties dialog box.
2. The text object border options are as follows.

rect 19, 68, 236, 93 [\(page 994\)](#)
 rect 21, 97, 243, 128 [\(page 994\)](#)
 rect 24, 132, 242, 161 [\(page 994\)](#)
 rect 21, 164, 247, 189 [\(page 995\)](#)
 rect 23, 193, 244, 223 [\(page 995\)](#)

	Field	Description	
A (page 995)	Line Color	For a line that is:	
		Solid	Color of the line
		Dash/Dot	Primary line color
			Selections include:
		Basic colors	Select from the field's drop-down list.
		Custom colors	When Custom is selected a Color palette opens. The palette can be expanded to include custom selections.
B (page 995)	Fill Color	Background area inside the border.	
		Selections are the same as for the line color.	
C (page 995)	Background Color	For a line that is	

		Solid	This the background color does not display.
		Dash	Secondary line color.
		Selections are the same as for the line color.	
D (page 995)	Line Width	The higher the number, the thicker the line	
E (page 995)	Line Style	Selections include	
		Solid	
		Dashes	
		Dashes/ Dots	
		None	The entry for no line is blank

3. Do one of the following.

- a. Click Apply to apply your entries and continue to configure other text features.
- b. Click OK to close the Text Object Properties dialog box and view your changes.

Option 5.4.3. Select the Text Object Font

1. Select the Font tab in the Text Object Properties dialog box.

2. Select the font as follows.

rect 15, 49, 335, 136 [\(page 995\)](#)


rect 343, 70, 407, 136 [\(page 995\)](#)

rect 20, 141, 93, 207 [\(page 996\)](#)

rect 22, 212, 117, 250 [\(page 996\)](#)

rect 344, 139, 407, 247 [\(page 996\)](#)

Field	Description
Font	Basic selections for the text object are:
Font	UNICODE fonts, e.g. Arial UNICODE, can be used for single and multi-byte languages.
Font Style	Standard bold, italic and bold italic selections
Size	Standard font sizes
Text Layout	Circle that is clicked determines the text position inside its text box.

	Examples	
Effects	Strikeout	
	Underline	
	Wrap text	Note: The text starts at the justify location and wraps from there. It is recommended that you justify the text from top left To help insure that it fits in the box.
Color	Choices include basic and custom colors.	
Rotate	Angle of text in the text box.	
	Examples	
	<p> Important: Text rotation will work only for Vector Fonts (True Type and Open Type fonts). Text rotation may not work as expected for fonts like Raster and bitmap based fonts. To determine a font type:</p> <ol style="list-style-type: none"> Open the Windows Control Panel. Double-click Fonts to open the Fonts window. Double-click the font you plan to use. <p>Result: The font type displays in the heading. Note: The icon to the left of the font also identifies the font type.</p>	

3. Do one of the following.
 - a. Click Apply to apply your entries and continue to configure other text features.
 - b. Click OK to close the Text Object Properties dialog box and view your changes.

Step 6. Configure Broadcast Bar Code Labels

Step 6. Configure Broadcast Bar Code Labels

You can add bar codes from several groups to the WYSIWYG. The bar code characters represent the data that can come from system or external data.

Step 6.1 (page 997)	Place a bar code object on the WYSIWYG form.
Step 6.2 (page 997)	Open the Barcode Object Properties dialog box.
Step 6.3 (page 998)	Select a data source for the bar code object.
Step 6.4 (page 1007)	Configure the bar code symbology.

Step 6.5 (page 1046)	Position bar codes on WYSIWYG forms.
---	--------------------------------------

Step 6.1. Place a Bar Code Object on the WYSIWYG Form

1. Do one of the following.

Method 1

- a. Click Draw on the WYSIWYG Text Editor menu bar.
- b. Select Barcode.

Method 2


Click the Bar Code button on the WYSIWYG Text Editor toolbar.

Method 3

- a. Right-click the WYSIWYG screen.
- b. Select Barcode.

2. Drag your mouse on the form where you want to place the bar code.

A bar code is placed where you drag.

 Tip : You can cut and paste or copy and paste a bar code from one WYSIWYG form into another (as described for an [image object \(page 1049\)](#)). The source bar code configuration is copied to the target.

Step 6.2. Open the Barcode Object Properties Dialog Box

1. Select the bar code that you just placed in the WYSIWYG form.
2. Do one of the following.

Method 1

Double-click the bar code.

Method 2

- a. Click Edit on the WYSIWYG Text Editor menu bar.
- b. Select Properties.

Method 3

Click the Properties button on the WYSIWYG Text Editor toolbar.

Method 4

Press Alt+Enter on the keyboard.

The Barcode Object Properties dialog box opens when you use any of the methods.

Step 6.3. Select a Data Source for the Bar Code Object

Step 6.3. Select a Data source for the Bar Code Object

Option 6.3.1 (page 998)	Enter static text as the bar code object data source.
Option 6.3.2 (page 998)	Select a system option as the bar code object data source.
Option 6.3.3 (page 1000)	Select an external option as the bar code object data source.
Option 6.3.4 (page 1002)	Use a script to define the bar code object data source.

Option 6.3.1. Enter Static Text as the Bar Code Object Data Source

The bar code characters can be configured

You can select this option alone or in addition to any of the other data source options. You can also use it to provide headings and descriptions for dynamic data.

1. Select the Data Source tab in the Barcode Object Properties dialog box.
2. Click Static.
3. Enter the text in the **Text** box that you want the bar code symbols to reflect.

If Static is the only data source selected, the bar code will display the associated symbols on the printed form.

Option 6.3.2. Select a System Option to Define the Bar Code

1. Select the Data Source tab in the Bar Code Object Properties dialog box.
2. Click System.

3. The options are as follows.

rect 245, 67, 395, 80 ([page 999](#))rect 247, 81, 392, 93 ([page 999](#))rect 244, 93, 390, 108 ([page 1000](#))

Option	Description	
Form Name	File name	
Date & Time	Inserts the date and time recorded on Broadcast server. Format options are:	
	Format	Example
	mm/dd	07/11
	mm/dd/yy	07/11/03
	dd/mm/yy	11/07/03
	dd/mm/yyyy	11/07/2003
	dd-Month	11-July
	dd-Month-yy	11-July-03
	dd-Month-yyyy	11-July-2003
	Month-yy	July-03
	Month-dd-yyyy	July-11-2003
	Month dd	July 11
	Month, yyyy	July, 2003
	dd-Month (Short Month Name)	11-Jul
	dd-Month-yy (Short Month Name)	11-Jul-03
	dd-Month-yyyy (Short Month Name)	11-Jul-2003
	Month-yy (Short Month Name)	Jul-03
	Month dd, yyyy (Short Month Name)	Jul 11, 2003
	Month dd (Short Month Name)	Jul 11
	Month, yyyy (Short Month Name)	Jul, 2003
	Day, Month dd, yyyy	Friday, July 11, 2003
	hh:mm AM/PM	11:13 AM
	hh:mm:ss AM/PM	11:13:37 AM
	HH:mm	11:14
	hh:mm:ss AM/PM	11:14:14 AM
	HH:mm:ss	11:14:55

	mm/dd/yyyy HH:mm	07/11/2003 11:15
	mm/dd/yyyy HH:mm AM/PM	07/11/2003 11:15:46 AM
	Day, Month dd, yyyy HH:mm:ss	Friday, July 11, 2003 11:16:14
	Day, Month dd, yyyy hh:mm:ss AM/PM	Friday, July 11, 2003 11:16:46 AM
Form Sequence Number	Sequence number of the broadcast; the number is set during runtime. It is useful to insure, for example, that printouts are in the correct order.	


Option 6.3.3. Select an External Option for the Bar Code Object Data Source

Option 6.3.3. Select an External Option for the Bar Code Object Data Source

1. Select the Data Source tab in the Barcode Object Properties dialog box.
2. Click External.
3. Select an external source in the **Type** field.

Option 6.3.3.1 (page 1000)	Use Solves to define the bar code.
Option 6.3.3.2 (page 1002)	Use Tracker attributes to define the bar code.
Option 6.3.3.3 (page 1002)	Use extended attributes to define the bar code.


Option 6.3.3.1. Use Solves to Provide Data for the Bar Code

 **Important:** You can open Solve's Expression Browser through the Barcode Object Properties dialog box. Make sure your project is running if you want to take advantage of this feature.

1. Click External on the Data Source tab in the Text Object Properties dialog box.
2. Select Solve in the **Type** field.
3. Use the Data Source features to select a Solve expression.

Feature	Descriptions
Named Solve / Expression	The selected solve expression that will be dynamically broadcast during runtime. Broadcast uses the active version for the selected expression.
Expression Browser access	You have access to the full functionality of the browser, e.g. browsing, creating new expressions, editing existing expression.
	Opens the Expression Browser. To add an existing expression: Double-click an expression in the Expression Browser. Result: The expression is entered in the Named Solve / Expression field.

New	Opens a New Expression dialog box.
	<p>To add a new expression:</p> <ol style="list-style-type: none"> Enter a unique name in the Name field. Click OK. <p>The Expression Editor opens for the new expression.</p> <ol style="list-style-type: none"> Write a new expression. Click Apply & Activate when you are finished. Click Cancel or the Close button in the Expression Editor. <p>Result: The new expression displays in the Named Solve / Expression field.</p>
Edit	Provides the ability to edit a selected expression.
If:	An expression is in the Named Solve / Expression field.
Then:	<p>The Expression Editor opens with the expression available for editing. To add an edited expression:</p> <ol style="list-style-type: none"> Edit the expression. Click Apply & Activate. Click Cancel or the Close button in the Expression Editor. <p>Result: The form will use the edited version of the expression when it is used by Broadcast. Note: Broadcast will use the active expression. If you choose not to activate the edited version, it will not be used.</p>
If:	No expression is in the Named Solve / Expression field.
Then:	Edit is inactive.
If:	The expression is not in the Solve database.
Then:	A message tells you that the expression does not exist.
Browse	Opens the Expression Browser

 **Important:** A Solve name/expression configured in the form that has multiple return values will result in a Data Fetch error in Broadcast, during runtime.

- Enter a unique name in the Name field.
- Click OK.
The Expression Editor opens for the new expression.
- Write a new expression.
- Click Apply & Activate when you are finished.
- Click Cancel or the Close button in the Expression Editor.
- Edit the expression.
- Click Apply & Activate.
- Click Cancel or the Close button in the Expression Editor.

The form will use the edited version of the expression when it is used by Broadcast.

Option 6.3.3.2. Use Tracker Attributes to Provide Data for the Bar Code

1. Click External on the Data Source tab in the Barcode Object Properties dialog box.
2. Select Tracker in the **Type** field.
3. Enter a Tracker attribute in the **Attribute Name** field, e.g. an attribute listed in the Attribute Maintenance dialog box.
4. (Optional) Change the default static text.

Note: You can enter static text in addition to selecting system or external data sources.

The default entry is Barcode.

- a. Click Static on the Data Source tab.
- b. Enter the text that you want the default bar code characters to represent. .

Option 6.3.3.3. Use Extended Attributes to Provide Data for the Bar Code

1. Click External on the Data Source tab in the Text Object Properties dialog box.
2. Select Ext Tracker in the **Type** field.
3. Enter an extended Tracker attribute in the **Attribute Name** field, e.g. an attribute listed on the Attributes tab in a PRT_UI Item dialog box.
4. (Optional) Change the default static text.

Note: You can enter static text in addition to selecting system or external data sources.

The default entry is Barcode.

- a. Click Static on the Data Source tab.
- b. Enter the text that you want the default bar code characters to represent.

Option 6.3.4. Use a Script to Provide Data for the Bar Code

Option 6.3.4. Use a Script to Provide Data for the Bar Code

- Configure script.
- Sample script 1.
- Sample script 2.

Configure script

1. Click Script on the Data Source tab in the Barcode Object Properties dialog box.
2. Click Edit.

A WYSIWYG Text Editor opens.

3. Enter the script that you want to apply to the object.

guide: Guidelines for the script

- The Compilation and Execution of the Object script starts and ends in function `Main`,
- `Sub Main` functions are mandatory.
- All the references to the object inside the script use `MyObject`.

Review:

Text object script properties.
Text object script constants.
Text object script date and time formats.
Text object form level script functions and return types.

Note: Menu options and toolbar buttons provide the same functionality as the same items in the CIMPPLICITY Program Editor.

4. Do one of the following to compile the script.
 - Click the Compile button .
 - Click File>Compile on the WYSIWYG Form Editor menu bar.

The script is compiled. Broadcast reports errors or informs you that the script compiled successfully.

5. Close the WYSIWYG Form Editor.

A message asks you if you want to save the script to the object.

6. Click Yes.

The script displays as read-only text in the **Script** box.

Note: You can also click Compile on the Data Source tab to compile the script.

7. Click Static.
8. Enter a brief description of your selection in the text box.
9. Click Apply.
10. (Optional) Revise the font, border and position specifications to accommodate the text requirements.

Note: This can be done during any stage of the form design.

11. Click OK.

The static text entry displays in the editable form. When the form is broadcast, the dynamic selection will replace this text.

Sample Script

```
Sub Main
  Dim strValue as String
  strValue = Form.ItemID
  MyObject.BarcodeValue = strValue
End Sub
```

Sample Script

```
Sub Main
  Dim strValue as String
  strValue = PointGet("ITEM ID")
  MyObject.TextValue = strValue
```

Bar Code Object Script Properties

Get and Set properties are:

Property Name	Data Type	Description
BarcodeValue	String	The value that needs to be printed as a barcode.
Rotation	Short	Rotation of the barcode (0 – 3) 0 - 0 ⁰ 1 – 90 ⁰ 2 – 180 ⁰ 3 – 270 ⁰
Ratio	Float	Barcode Ratio (2.0 to 3.0) if the barcode type supports.
Checksum	Bool	Enable check sum for the barcode if the barcode type supports.
BarColor	Long	Barcode bar color
BackgroundColor	Long	Barcode background color
CaptionBold	Bool	Barcode Caption style Bold

CaptionItalic	Bool	Barcode Caption style Italic
CaptionUnderline	Bool	Barcode Caption style Underline
CaptionStrikeout	Bool	Barcode Caption style Strikeout
CaptionFontName	String	Barcode Caption Font Name
CaptionPointSize	Short	Barcode Caption size (1 - 254)
CaptionLocation	Short	Barcode Caption Location (0 – 6)
CaptionColor	Long	Barcode Caption Color
MaxicodeMode	Short	Maxicode barcode Mode (2 – 6)
MaxicodePostalCode	String	Maxicode barcode Postal Code
MaxicodeCountryCode	String	Maxicode barcode Country Code
MaxicodeClassOfService	String	Maxicode barcode Class of Service
PDF417Columns	Short	PDF417 barcode Columns (1 – 30)
PDF417Rows	Short	PDF417 barcode Rows (3 – 90)
PDF417ErrorLevel	Short	PDF417 barcode Error Level (-1 – 8)
PDF417Truncate	Bool	PDF417 barcode truncate or not

Bar Code Objects Script Constants

Constants that can be used for the bar code include:

- Bar code rotation.
- Bar code caption location.
- Predefined colors.

Bar code Rotation

Rotate0

Rotate90

Rotate180

Rotate270

Bar code Caption Location

None

BelowLeft

BelowCenter

BelowRight

AboveLeft

AboveCenter

AboveRight

Predefined Colors

Black

Red

Blue

Green

Yellow

Fushcia

Aqua

White

Bar Code Form Level Script Functions and Return Types

The following functions will return the runtime form level properties so the user can use them in the script.

All the properties:

- Are read only.
- Can be accessed using the syntax:

Form.XXXX where XXXX can be one of the values given below:

Function Name	Return Type	Description
ProjectName	String	Project name where this form is broadcast.
ItemID	String	Item ID for the broadcast.
SequenceNo	String	Sequence Number of the broadcast.
Name	String	WYSIWYG Form name.

Following are functions that can be used to access:

- Tracker attributes
- Extended Tracker attributes
- Solve value

Function	Where
<code>Form.GetTrackerAttr(YYYY)</code>	YYYY is a string passed as a tracker or extended tracker attribute.
<code>Form.GetSolveValue(YYYY)</code>	YYYY is a string passed as a Solve name or expression.

A function that gets the Long value of Color in using RGB is:

Function	Where
<code>RGBColor(Rvalue , Gvalue , Bvalue)</code>	Rvalue is the Red value Gvalue is the Green value Bvalue is the Blue value

Step 6.4. Configure the Bar Code Symbology

Step 6.4. Configure the Bar Code Symbology

Option 6.4.1 (page 1008)	Configure a Code 3 of 9 Full ASCII bar code.
Option 6.4.2 (page 1010)	Configure an Interleaved 2 of 5 bar code.
Option 6.4.3 (page 1012)	Configure a Codebar bar code.
Option 6.4.4 (page 1015)	Configure an MSI Plessey bar code.
Option 6.4.5 (page 1017)	Configure a Code 93 bar code.
Option 6.4.6 (page 1020)	Configure a UPCA bar code.
Option 6.4.7 (page 1021)	Configure a UPC-E (zero suppressed) bar code.
Option 6.4.8 (page 1022)	Configure a PostNet bar code.
Option 6.4.9 (page 1024)	Configure an EAN/JAN-8 bar code.
Option 6.4.10 (page 1025)	Configure an EAN/JAN-13 bar code.
Option 6.4.11 (page 1027)	Configure a Code 128 bar code.

Option 6.4.12 (page 1028)	Configure a Code 3 of 9 Standard bar code.
Option 6.4.13 (page 1031)	Configure a PDF417 bar code.
Option 6.4.14 (page 1033)	Configure a MaxiCode bar code.
Option 6.4.15 (page 1034)	Configure a DataMatrix bar code.
Option 6.4.16 (page 1035)	Configure a LogMars bar code.

Option 6.4.1. Configure Code 3 of 9 Full ASCII Bar Code

Code 3 of 9 principal feature is to encode messages using the [full alphanumeric character \(page 1038\)](#) set.

1. Select the Barcode tab in the Barcode Object Properties dialog box.
2. Select Code 3 of 9 Full ASCII in the **Symbology** field.

Note: Code 3 of 9 Full ASCII is a one-dimensional bar code type.

3. Fill in the fields as follows.

	Option	Description	
1 (page 1008)	Bar Code		
	Width	Bar code width from left to right in points	
	Rotation	Options going counter-clockwise are:	
		0o	No rotation
		90o	Left-Vertical
		180o	Upside down
		270o	Right-vertical
	Bar Color	Color of the bar characters.	
	Background Color	Color of the area behind the bars that is background.	
	Ratio	Range	2.0-3.0
		Default	2.0

	Checksum	Checked	Any checksum that is specific to any implementation has to be done outside the scope of broadcast, and should be part of the data made available to broadcast.
		Default	Clear
2 (page 1008)	Caption		
	Style	Available features for the caption font are: <ul style="list-style-type: none"> • Bold • Italic • Underline • Strike Thru 	
	Font Name	Selected font for the caption. This font can be a standard font (different from the bar code font) so it can be easily read.	
	Point Size	Size in which the caption font will display. The caption font displays within the space allocated for the bar. Therefore, the larger the font the more space it uses and the less space there is to display the bar. You can increase the space for the bar or decrease the font size if the bar is smaller than you want.	
	Location	Caption font position static to the bar. Options are.	
		<ul style="list-style-type: none"> • None • Below Left • Below Center • Below Right • Above Left • Above Center • Above Right 	
	Color	Color of the caption font.	
* (page 1008)	Colors can be customized through the Color palette.		
	Select Custom in the Color field to display the palette.		

4. Select the Data Source tab.

5. Click Static.

6. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

Requirements	Description
Valid characters	Six special alphanumeric characters.
	Capital letters A through D
	All numeric digits
Length	Variable

Options for Code 39 symbology include

Option	Description
Full ASCII	Standard Code 39 contains only 43 characters as follows. 0-9 A-Z \$ / % + Code 39 can be extended to an 128 character symbology (full ASCII) by combining one of the special characters (\$, /, %, +) with a letter (A-Z) to form the characters that are not present in the standard Code 39 symbology. Example In standard Code 39 a lowercase a cannot be represented. In Code 39 Full ASCII a lowercase a is represented as +A.
Check Digit	A modulo 43 check character can be used to enhance data security for Code 39 symbols. The last digit of the symbol is assumed to be the check digit; it is compared to a calculated check digit to verify the symbol.
Append	It is sometimes advantageous to break up long messages into multiple, shorter symbols. If the first character of a Code 39 symbol is a space (ASCII 32), then the scanned symbol is appended to a storage buffer. This operation continues for all successive Code 39 symbols with a leading space being added to previously stored ones. When a Code 39 symbol that does not include a leading space is scanned: <ul style="list-style-type: none"> a. The symbol appended to the buffer b. The entire buffer is transmitted c. The buffer is cleared.

7. Click Apply.

8. Click OK.

The bar code displays your static text specifications.

Option 6.4.2. Configure an Interleaved 2 of 5 Bar Code

Interleaved 2 of 5:

- Is typically used in industrial and master carton labeling.
- Is a variable length, even numbered, numeric bar code.

1. Select the Barcode tab in the Barcode Object Properties dialog box.
2. Select Interleaved 2 of 5 in the **Symbology** field.

Note: Interleaved 2 of 5 is a one-dimensional bar code type.

3. Fill in the fields as follows.

rect 9, 58, 265, 263 [\(page 1011\)](#)

rect 265, 58, 490, 274 [\(page 1011\)](#)

rect 21, 270, 146, 366 [\(page 1012\)](#)

rect 56, 29, 118, 56 [\(page 1012\)](#)

	Option	Description	
1 (page 1010)	Bar Code		
	Width	Bar code width from left to right in points	
	Rotation	Options going counter-clockwise are:	
		0o	No rotation
		90o	Left-Vertical
		180o	Upside down
		270o	Right-vertical
	Bar Color	Color of the bar characters.	
	Background Color	Color of the area behind the bars that is background.	
	Ratio	Range	2.0-3.0
		Default	2.0
	Checksum	Checked	Any checksum that is specific to any implementation has to be done outside the scope of broadcast, and should be part of the data made available to broadcast.
		Default	Clear
2 (page 1010)	Caption		
	Style	Available features for the caption font are: <ul style="list-style-type: none"> • Bold • Italic • Underline • Strike Thru 	
	Font Name	Selected font for the caption. This font can be a standard font (different from the bar code font) so it can be easily read.	
	Point Size	Size in which the caption font will display. The caption font displays within the space allocated for the bar. Therefore, the larger the font the more space it uses and the less space there is to display the bar. You can increase the space for the bar or decrease the font size if the bar is smaller than you want.	
	Location	Caption font position static to the bar. Options are.	
		<ul style="list-style-type: none"> • None • Below Left • Below Center • Below Right • Above Left • Above Center • Above Right 	
	Color	Color of the caption font.	

* (page 1010)	Colors can be customized through the Color palette.
	Select Custom in the Color field to display the palette.

4. Select the Data Source tab.
5. Click Static.
6. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

Requirements	Description
Valid characters	Numbers

The symbology uses

- Bars to represent the first character
- Interleaved (white) spaces to represent the second character.

Each character has:

- Two wide elements
- Three narrow elements.

Options for Interleaved 2 of 5 bar code include the following.

Option	Description
Check Digit	A modulo 10 check character can be used to enhance data security for Interleaved 2 of 5 symbols. When this option is selected, the last digit of the symbol is assumed to be the check digit, and it is compared to a calculated check digit to verify the symbol.

7. Click Apply.
8. Click OK.

The bar code displays your default specifications.

Note: The bar characters may change when they are broadcast to represent your other data source selections.

Option 6.4.3. Configure a Codabar Bar Code

Option 6.4.3. Configure a Codabar Bar Code

Codabar is a variable length symbology capable of encoding 16 characters within any length message.

1. Select the Barcode tab in the Barcode Object Properties dialog box.
2. Select Codabar in the **Symbology** field.

Note: Codabar is a one-dimensional bar code type.

3. Fill in the fields as follows.

rect 11, 48, 260, 258 [\(page 1013\)](#)

rect 264, 50, 488, 264 [\(page 1013\)](#)

rect 22, 270, 146, 363 [\(page 1014\)](#)

rect 53, 21, 119, 50 [\(page 1015\)](#)

	Option	Description	
1 (page 1013)	Bar Code		
	Width	Bar code width from left to right in points	
	Rotation	Options going counter-clockwise are:	
		0o	No rotation
		90o	Left-Vertical
		180o	Upside down
		270o	Right-vertical
	Bar Color	Color of the bar characters.	
	Background Color	Color of the area behind the bars that is background.	
	Ratio	Range	2.0-3.0
		Default	2.0
	Checksum	Checked	Any checksum that is specific to any implementation has to be done outside the scope of broadcast, and should be part of the data made available to broadcast.
		Default	Clear
2 (page 1013)	Caption		

	Style	Available features for the caption font are: <ul style="list-style-type: none"> • Bold • Italic • Underline • Strike Thru
	Font Name	Selected font for the caption. This font can be a standard font (different from the bar code font) so it can be easily read.
	Point Size	Size in which the caption font will display. The caption font displays within the space allocated for the bar. Therefore, the larger the font the more space it uses and the less space there is to display the bar. You can increase the space for the bar or decrease the font size if the bar is smaller than you want.
	Location	Caption font position static to the bar. Options are. <ul style="list-style-type: none"> • None • Below Left • Below Center • Below Right • Above Left • Above Center • Above Right
	Color	Color of the caption font.
	* (page 1013)	Colors can be customized through the Color palette.
		Select Custom in the Color field to display the palette.

4. Select the Data Source tab.

5. Click Static.

6. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

Requirements	Description
Valid characters	Six special alphanumeric characters.
	Capital letters A through D
	All numeric digits

Options for Codabar symbology include:

Option	Description
Check Digit	A modulo 16 check character can be used to enhance data security for Codabar symbols. The last digit of the symbol is <ul style="list-style-type: none"> • Assumed to be the check digit • Compared to a calculated check digit to verify the symbol.

Append	A D stop character is used to indicate that the data from the symbol should be concatenated with data from an adjacent symbol with a D start character. Append can concatenate any number of Codabar symbols in a row up to the maximum message length. The D characters are omitted.
Traditional Names	The traditional names ascribed different start and stop characters. Start characters are a b c d Stop characters are t n * e Standard start/stop characters are A B C D
Special Characters	When calling the DLLs, use the following keyboard characters in the input string to insert the Special Characters: A (Alt+065), B (Alt+066), C (Alt+067), and D (Alt+068).

7. Click Apply.

8. Click OK.

The bar code displays your default specifications.

Option 6.4.3.1. Options for Codabar Symbology

Options for Codabar symbology include:

Option	Description
Check Digit	A modulo 16 check character can be used to enhance data security for Codabar symbols. When this option is selected, the last digit of the symbol is assumed to be the check digit, and it is compared to a calculated check digit to verify the symbol.
Append	When this option is selected, a "D" stop character is used to indicate that the data from the symbol should be concatenated with data from an adjacent symbol with a "D" start character. This option can concatenate any number of Codabar symbols in a row up to the maximum message length. The "D" characters are omitted.
Traditional Names	The traditional names ascribed different start and stop characters. The start characters are "a", "b", "c", and "d". The stop characters are "t", "n", "*", and "e". The standard start/stop characters are "A", "B", "C", and "D".
Special Characters	When calling the DLLs, use the following keyboard characters in the input string to insert the Special Characters: A (Alt+065), B (Alt+066), C (Alt+067), and D (Alt+068).

Option 6.4.4. Configure an MSI Plessey Bar Code

MSI Plessey is primarily used in marking retail shelves.

MSI Plessey is a variable length numeric symbology. Each character consists of four bars with intervening spaces for each encoded digit, one or two symbol check digits, and a reverse start code.

1. Select the Barcode tab in the Barcode Object Properties dialog box.
2. Select MSI Plessey in the **Symbology** field.

Note: MSI Plessey is a one-dimensional bar code type.

3. Fill in the fields as follows.

rect 8, 56, 261, 262 ([page 1016](#))

rect 263, 57, 491, 262 ([page 1016](#))

rect 18, 273, 141, 363 ([page 1017](#))

rect 50, 21, 117, 53 ([page 1017](#))

Option	Description	
1 (page 1016)	Bar Code	
Width	Bar code width from left to right in points	
Rotation	Options going counter-clockwise are:	
	0o	No rotation
	90o	Left-Vertical
	180o	Upside down
	270o	Right-vertical
Bar Color	Color of the bar characters.	
Background Color	Color of the area behind the bars that is background.	
Ratio	Range	2.0-3.0
	Default	2.0
Checksum	Checked	Any checksum that is specific to any implementation has to be done outside the scope of broadcast, and should be part of the data made available to broadcast.
	Default	Clear
2 (page 1016)	Caption	
Style	Available features for the caption font are: <ul style="list-style-type: none"> • Bold • Italic • Underline • Strike Thru 	
Font Name	Selected font for the caption. This font can be a standard font (different from the bar code font) so it can be easily read.	
Point Size	Size in which the caption font will display. The caption font displays within the space allocated for the bar. Therefore, the larger the font the more space it uses and the less space there is to display the bar. You can increase the space for the bar or decrease the font size if the bar is smaller than you want.	

	Location	Caption font position static to the bar. Options are.
		<ul style="list-style-type: none"> • None • Below Left • Below Center • Below Right • Above Left • Above Center • Above Right
	Color	Color of the caption font.
* (page 1016)		All colors in the bar code can be customized through the Color palette.
		Select Custom in the Color field to display the palette.

4. Select the Data Source tab.
5. Click Static.
6. Make a valid entry in the **Text** box.

Valid characters	Numbers
Maximum length	15 characters

Each character consists of the following.

- Four bars with intervening spaces for each encoded digit,
- One or two symbol check digits,
- A reverse start code.

Options for MSI Plessey bar code include the following.

Option	Description
Second Check Digit	A modulo 10 check digit calculation is always performed on the data string. A second modulo 10 check digit is optional and is used to perform a check on the entire string including the first check digit.

7. Click Apply.
8. Click OK.

The bar code displays your default specifications.

Option 6.4.5. Configure a Code 93 Bar Code

Code 93 is similar to [Code 39 \(page 1028\)](#) but encodes more characters per inch.

Code 93 encodes the full 128 [ASCII character set \(page 1042\)](#) and is encoded similarly to the [extended Code 39 \(page 1008\)](#).

1. Select the Barcode tab in the Barcode Object Properties dialog box.
2. Select Code 93 in the **Symbology** field.

Note: Code 93 is a one-dimensional bar code type.

3. Fill in the fields as follows.

	Option	Description
1 (page 1018)	Bar Code	
	Width	Bar code width from left to right in points
	Rotation	Options going counter-clockwise are:
		0o No rotation
		90o Left-Vertical
		180o Upside down
		270o Right-vertical
	Bar Color	Color of the bar characters.
	Background Color	Color of the area behind the bars that is background.
2 (page 1018)	Caption	
	Style	Available features for the caption font are: <ul style="list-style-type: none"> • Bold • Italic • Underline • Strike Thru
	Font Name	Selected font for the caption. This font can be a standard font (different from the bar code font) so it can be easily read.
	Point Size	Size in which the caption font will display. The caption font displays within the space allocated for the bar. Therefore, the larger the font the more space it uses and the less space there is to display the bar. You can increase the space for the bar or decrease the font size if the bar is smaller than you want.
	Location	Caption font position static to the bar. Options are.

		<ul style="list-style-type: none"> • None • Below Left • Below Center • Below Right • Above Left • Above Center • Above Right
	Color	Color of the caption font.
* (page 1018)	Colors can be customized through the Color palette.	
	Select Custom in the Color field to display the palette.	

4. Select the Data Source tab.

5. Click Static.

6. Make a valid entry in the **Text** box.

Valid characters	Full 128 ASCII character set (page 1042)
------------------	--

Code 93 encodes as follows.

- Nine modules are arranged into 3 bars with adjacent spaces.
- Two of the characters are check characters.

Options for the Code 93 symbology include the following.

Option	Description
Append	It is sometimes advantageous to break up long messages into multiple, shorter symbols. If the first character of a Code 93 symbol is a space (ASCII 32), then the scanned symbol is appended to a storage buffer. This operation continues for all successive Code 93 symbols with a leading space being added to previously stored ones. When a Code 93 symbol that does not include a leading space is scanned: <ul style="list-style-type: none"> a. It is appended to the buffer. b. The entire buffer is transmitted. c. The buffer is cleared.

7. It is appended to the buffer.

8. The entire buffer is transmitted.

9. The buffer is cleared.

10. Click Apply.

11. Click OK.

The bar code displays your default specifications.

Option 6.4.6. Configure a UPC-A Bar Code

UPC-A (Universal Product Code-A) is the most common UPC bar code for retail product labeling and is seen in most grocery stores across the United States.

1. Select the Barcode tab in the Barcode Object Properties dialog box.
2. Select UPC-A in the **Symbology** field.

Note: UPC-A is a one-dimensional bar code type.

3. Fill in the fields as follows.

rect 19, 58, 266, 256 ([page 1020](#))

rect 18, 274, 144, 365 ([page 1020](#))

rect 57, 27, 119, 55 ([page 1021](#))

Option	Description
Bar Code	
Width	Bar code width from left to right in points
Rotation	Options going counter-clockwise are:
	0o No rotation
	90o Left-Vertical
	180o Upside down
	270o Right-vertical
Bar Color	Color of the bar characters.
Background Color	Color of the area behind the bars that is background.
*	Colors can be customized through the Color palette.
	Select Custom in the Color field to display the palette.

4. Select the Data Source tab.
5. Click Static.
6. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

Requirements	Description
Valid characters	Numeric only 12 digit number

12 digit code	First six digits	Are assigned from the Uniform Code Council (UCC) in Dayton, Ohio
	Next five digits	Are assigned by the manufacturer
	Final digit	Is a modulo 10 check digit
Height for the bar code	Nominal	One inch
	Reduced size	80% of the nominal size

7. Click Apply.

8. Click OK.

The bar code displays your default specifications.

Option 6.4.7. Configure a UPC-E (zero suppressed) Bar Code

UPC-E (Universal Product Code-E) is:

- Used for marking small packages including magazines and paperback books.
- A fixed length
- A compressed six digit code

UPC-E symbols are UPC-A symbols that have been zero suppressed (i.e. consecutive zeros are not included in the symbol).

1. Select the Barcode tab in the Barcode Object Properties dialog box.
2. Select UPC-E in the **Symbology** field.

Note: UPC-E is a one-dimensional bar code type.

3. Fill in the fields as follows.

Option	Description
Bar Code	
Width	Bar code width from left to right in points
Rotation	Options going counter-clockwise are:
	0o No rotation
	90o Left-Vertical
	180o Upside down

		270o	Right-vertical
	Bar Color	Color of the bar characters.	
	Background Color	Color of the area behind the bars that is background.	
*	Colors can be customized through the Color palette.		
	Select Custom in the Color field to display the palette.		

4. Select the Data Source tab.
5. Click Static.
6. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

Requirements	Description	
Valid characters	Numeric only 12 digit number	
12 digit code	First six digits	Are assigned from the Uniform Code Council (UCC) in Dayton, Ohio
	Next five digits	Are assigned by the manufacturer
	Final digit	Is a modulo 10 check digit
Height for the bar code	Nominal	One inch
	Reduced size	80% of the nominal size

7. Click Apply.
8. Click OK.

The bar code displays your default specifications.

Option 6.4.8. Configure a PostNet Bar Code

Postal Numeric Encoding Technique is used to encode ZIP code information on letter mail.

PostNet utilizes redundant information within a compact bar code format to provide error detection capability and a significant degree of error correction capability. The check digit is always enabled.

1. Select the Barcode tab in the Barcode Object Properties dialog box.
2. Select PostNet in the **Symbology** field.

Note: PostNet is a one-dimensional bar code type.

3. Fill in the fields as follows.

rect 8, 52, 263, 266 ([page 1023](#))

rect 267, 54, 490, 272 ([page 1023](#))

rect 22, 272, 145, 360 ([page 1023](#))

rect 58, 23, 119, 50 ([page 1024](#))

	Option	Description
1	Bar Code	
	Width	Bar code width from left to right in points
	Rotation	Options going counter-clockwise are:
		0o No rotation
		90o Left-Vertical
		180o Upside down
		270o Right-vertical
	Bar Color	Color of the bar characters.
	Background Color	Color of the area behind the bars that is background.
2	Caption	
	Style	Available features for the caption font are: <ul style="list-style-type: none"> • Bold • Italic • Underline • Strike Thru
	Font Name	Selected font for the caption. This font can be a standard font (different from the bar code font) so it can be easily read.
	Point Size	Size in which the caption font will display. The caption font displays within the space allocated for the bar. Therefore, the larger the font the more space it uses and the less space there is to display the bar. You can increase the space for the bar or decrease the font size if the bar is smaller than you want.
	Location	Caption font position static to the bar. Options are. <ul style="list-style-type: none"> • None • Below Left • Below Center • Below Right • Above Left • Above Center • Above Right
	Color	Color of the caption font.
*	Colors can be customized through the Color palette.	
	Select Custom in the Color field to display the palette.	

4. Select the Data Source tab.

5. Click Static.

6. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

Requirements	Description
Valid characters	Any of the following zip code formats
	Five numbers
	Nine (zip+4)
	Nine+DP digits

7. Click Apply.

8. Click OK.

The bar code displays your default specifications.

Option 6.4.9. Configure an EAN/JAN-8 Bar Code

EAN/JAN-8 (European Article Numbering/Japanese Article Numbering) is

- Used when a bar code label must be affixed to a small package.
- Essentially a shortened version of [EAN-13 \(page 1025\)](#) code, similar to the [UPC-E \(page 1021\)](#) code.

1. Select the Barcode tab in the Barcode Object Properties dialog box.

2. Select EAN/JAN-8 in the **Symbology** field.

Note: EAN/JAN-8 is a one-dimensional bar code type.

3. Fill in the fields as follows.

Option	Description
Bar Code	
Width	Bar code width from left to right in points
	Minimum 80
Rotation	Options going counter-clockwise are:

		0o	No rotation
		90o	Left-Vertical
		180o	Upside down
		270o	Right-vertical
	Bar Color	Color of the bar characters.	
	Background Color	Color of the area behind the bars that is background.	
*	Colors can be customized through the Color palette.		
	Select Custom in the Color field to display the palette.		

4. Select the Data Source tab.
5. Click Static.
6. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

Requirements	Description	
Valid characters	Numeric only 8 digit number	
8 digit code	First two digits	Country code
	Next five digits	Data digits
	Final digit	Check sum
Height for the bar code	Nominal	One inch
	Reduced size	80% of the nominal size

7. Click Apply.
8. Click OK.

The bar code displays your default specifications.

Option 6.4.10. Configure an EAN/JAN-13 Bar Code

EAN/JAN-13 (European Article Numbering/Japanese Article Numbering) is:

- Used worldwide in bar coding retail goods.
- Fixed length and is similar to the [UPC-A \(page 1020\)](#) symbology, but encodes a 13th digit.

1. Select the Barcode tab in the Barcode Object Properties dialog box.

2. Select EAN/JAN-13 in the **Symbology** field.

Note: EAN/JAN-13 is a one-dimensional bar code type.

3. Fill in the fields as follows.

Option	Description
Bar Code	
Width	Bar code width from left to right in points
Rotation	Options going counter-clockwise are:
	0o No rotation
	90o Left-Vertical
	180o Upside down
	270o Right-vertical
Bar Color	Color of the bar characters.
Background Color	Color of the area behind the bars that is background.
*	Colors can be customized through the Color palette.
	Select Custom in the Color field to display the palette.

4. Select the Data Source tab.

5. Click Static.

6. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

Requirements	Description
Valid characters	0-9
Length	Options for number of digits are:
	12
	14 12 + 2 digit supplement
	17 12 + 5 digit supplement
Height for the bar code	Nominal One inch
	Reduced 80% of the nominal size size

7. Click Apply.

8. Click OK.

The bar code displays your default specifications.

Option 6.4.11. Configure a Code 128 Bar Code

Code 128 is the most easily read code with the highest message integrity due to several separate message check routines.

1. Select the Barcode tab in the Barcode Object Properties dialog box.
2. Select Code 128 in the **Symbology** field.

Note: Code 128 is a one-dimensional bar code type.

3. Fill in the fields as follows.

rect 8, 51, 258, 251 ([page 1027](#))

rect 262, 54, 488, 268 ([page 1027](#))

rect 21, 271, 142, 359 ([page 1028](#))

rect 56, 25, 119, 53 ([page 1028](#))

Option	Description
1	Bar Code
Width	Bar code width from left to right in points
Rotation	Options going counter-clockwise are:
	0o No rotation
	90o Left-Vertical
	180o Upside down
	270o Right-vertical
Bar Color	Color of the bar characters.
Background Color	Color of the area behind the bars that is background.
2	Caption
Style	Available features for the caption font are: <ul style="list-style-type: none"> • Bold • Italic • Underline • Strike Thru

Font Name	Selected font for the caption. This font can be a standard font (different from the bar code font) so it can be easily read.
Point Size	Size in which the caption font will display. The caption font displays within the space allocated for the bar. Therefore, the larger the font the more space it uses and the less space there is to display the bar. You can increase the space for the bar or decrease the font size if the bar is smaller than you want.
Location	Caption font position static to the bar. Options are.
	<ul style="list-style-type: none"> • None • Below Left • Below Center • Below Right • Above Left • Above Center • Above Right
Color	Color of the caption font.
*	Colors can be customized through the Color palette.
	Select Custom in the Color field to display the palette.

4. Select the Data Source tab.
5. Click Static.
6. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

Valid characters	Full 128 ASCII character set (page 1042)
Length	Variable

7. Click Apply.
8. Click OK.

The bar code displays your default specifications.

Option 6.4.12. Configure a Code 3 of 9 Standard Bar Code

Code 39 is variable length and is the most frequently used symbology in industrial bar code systems today.

The principal feature is to encode messages using the [alphanumeric character \(page 1038\)](#) set.

1. Select the Barcode tab in the Barcode Object Properties dialog box.

2. Select Code 3 of 9 Standard in the **Symbology** field.

Note: Code 3 of 9 Standard is a one-dimensional bar code type.

3. Fill in the fields as follows.

rect 8, 50, 264, 263 ([page 1029](#))

rect 265, 52, 489, 266 ([page 1029](#))

rect 19, 274, 146, 367 ([page 1030](#))

rect 54, 23, 120, 51 ([page 1031](#))

Option	Description	
1	Bar Code	
	Width	Bar code width from left to right in points
	Rotation	Options going counter-clockwise are:
		0o No rotation
		90o Left-Vertical
		180o Upside down
		270o Right-vertical
	Bar Color	Color of the bar characters.
	Background Color	Color of the area behind the bars that is background.
	Ratio	Range 2.0-3.0
		Default 2.0
	Checksum	Checked Any checksum that is specific to any implementation has to be done outside the scope of broadcast, and should be part of the data made available to broadcast.
		Default Clear
2	Caption	
	Style	Available features for the caption font are: <ul style="list-style-type: none"> • Bold • Italic • Underline • Strike Thru
	Font Name	Selected font for the caption. This font can be a standard font (different from the bar code font) so it can be easily read.
	Point Size	Size in which the caption font will display. The caption font displays within the space allocated for the bar. Therefore, the larger the font the more space it uses and the less space there is to display the bar. You can increase the space for the bar or decrease the font size if the bar is smaller than you want.
	Location	Caption font position static to the bar. Options are.

		<ul style="list-style-type: none"> • None • Below Left • Below Center • Below Right • Above Left • Above Center • Above Right
	Color	Color of the caption font.
*	Colors can be customized through the Color palette.	
	Select Custom in the Color field to display the palette.	

4. Select the Data Source tab.

5. Click Static.

6. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

Requirements	Description
Valid characters	Six special alphanumeric characters.
	Capital letters A through D
	All numeric digits
Length	Variable

Options for Code 39 symbology include

Option	Description
Full ASCII	Standard Code 39 contains only 43 characters as follows. 0-9 A-Z \$ / % + Code 39 can be extended to an 128 character symbology (full ASCII) by combining one of the special characters (\$, /, %, +) with a letter (A-Z) to form the characters that are not present in the standard Code 39 symbology. Example In standard Code 39 a lowercase a cannot be represented. In Code 39 Full ASCII a lowercase a is represented as +A.
Check Digit	A modulo 43 check character can be used to enhance data security for Code 39 symbols. The last digit of the symbol is assumed to be the check digit; it is compared to a calculated check digit to verify the symbol.
Append	It is sometimes advantageous to break up long messages into multiple, shorter symbols. If the first character of a Code 39 symbol is a space (ASCII 32), then the scanned symbol is appended to a storage buffer. This operation continues for all successive Code 39 symbols with a leading space being added to previously stored ones. When a Code 39 symbol that does not include a leading space is scanned: <ul style="list-style-type: none"> a. The symbol appended to the buffer b. The entire buffer is transmitted c. The buffer is cleared.

7. Click Apply.

8. Click OK.

The bar code displays your static text specifications.

Option 6.4.13. Configure a PDF417 Bar Code

PDF417 (Portable Data File) is:

- Used in a wide variety of applications, including logistics & transportation, retailing, healthcare, government, identification, and manufacturing.
- A stacked-bar symbol that can encode full ASCII, numeric or binary data and uses sophisticated error correction algorithms to keep intact the

Large amounts of text and data can be stored securely and inexpensively when using the PDF417 symbology

1. Select the Barcode tab in the Barcode Object Properties dialog box.
2. Select PDF417 in the **Symbology** field.

Note: PDF417 is a two-dimensional bar code type, providing the capability to attach data files to physical objects.

3. Fill in the fields as follows.

rect 8, 52, 260, 261 ([page 1031](#))

rect 263, 51, 486, 266 ([page 1032](#))

rect 20, 270, 144, 360 ([page 1032](#))

rect 54, 21, 116, 52 ([page 1033](#))

	Option	Description	
1 (page 1031)	Bar Code		
	Width	Bar code width from left to right in points	
	Rotation	Options going counter-clockwise are:	
		0o	No rotation
		90o	Left-Vertical
		180o	Upside down
		270o	Right-vertical

	Bar Color	Color of the bar characters.		
	Background Color	Color of the area behind the bars that is background.		
2 (page 1031)	PDF417Info			
	Columns	Valid Range	1 through 30	
		Note: If zero, then the number of columns and rows is calculated to best fit a 1 to 3 column to row ratio.		
	Rows	Valid Range	3 through 90	
		If:	The number of rows is zero:	
		Then:	The number of rows is calculated given the number of columns	
		If:	The number of columns is zero,	
		Then:	The number of columns is not used	
	Error Levels	Valid Range	-1 to 8	
		level 0	Provides only error detection,	
		level 8	Provides the maximum error detection and correction capacity.	
		Note: If the error level is equal to minus one the error correction level is set to the recommended level for the given number of data codewords in the symbol, as shown in the following table.		
		Number of Data Codewords		Error Correction Level
		1 - 40		2
		41 - 160		3
		161 - 320		4
		321 - 863		5
	Truncate	If TRUE, omit the right row indicators and the stop pattern for the PDF417 symbol. Important: Truncation reduces the non-data overhead for the symbol, but also reduces the decode performance and the ability of the symbol to withstand degradation. This value should be TRUE only in statically clean environments where label damage is unlikely.		
* (page 1031)	Colors can be customized through the Color palette.			
	Select Custom in the Color field to display the palette.			

4. Select the Data Source tab.

5. Click Static.

6. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

PDF417 is variable length and consists of 4 bars and 4 spaces in a 17 module structure. Each PDF417 symbol consists of 3 to 90 stacked rows surrounded by a quiet zone on all four sides.

Each row consists of a leading quiet zone, start pattern, left row indicator character, one to thirty data characters, right row indicator character, stop pattern, and trailing quiet zone.

PDF417 can accommodate up to 340 characters per square inch with a maximum data capacity of 1850 text characters.

7. Click Apply.

8. Click OK.

The bar code displays your static text specifications.

Option 6.4.14. Configure a MaxiCode Bar Code

1. Select the Barcode tab in the Barcode Object Properties dialog box.

2. Select MaxiCode in the **Symbology** field.

Note: MaxiCode is a two-dimensional bar code type, providing the capability to attach data files to physical objects.

3. Fill in the fields as follows.

rect 10, 54, 262, 264 ([page 1033](#))

rect 264, 55, 490, 268 ([page 1034](#))

rect 21, 273, 148, 364 ([page 1034](#))

rect 48, 25, 119, 53 ([page 1034](#))

	Option	Description	
1	Bar Code		
	Width	Bar code width from left to right in points	
	Rotation	Options going counter-clockwise are:	
		0o	No rotation
		90o	Left-Vertical
		180o	Upside down
		270o	Right-vertical

	Bar Color	Color of the bar characters.
	Background Color	Color of the area behind the bars that is background.
2	MaxiCode Info	
	Mode	
	Postal Code	
	Country Code	
	Class of Service	
*	Colors can be customized through the Color palette.	
	Select Custom in the Color field to display the palette.	

4. Select the Data Source tab.

5. Click Static.

6. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

7. Click Apply.

8. Click OK.

The bar code displays your default specifications.

Option 6.4.15. Configure a DataMatrix Bar Code

DataMatrix is a:

- Two dimensional matrix symbology, which is made up of square modules arranged with a perimeter finder pattern.
- Two-dimensional bar code type.

rect 21, 275, 141, 362 ([page 1035](#))

	Option	Description	
	Bar Code		
	Width	Bar code width from left to right in points	
	Rotation	Options going counter-clockwise are:	
		0o	No rotation

		90o	Left-Vertical
		180o	Upside down
		270o	Right-vertical
	Bar Color	Color of the bar characters.	
	Background Color	Color of the area behind the bars that is background.	
* (page 1034)	Colors can be customized through the Color palette.		
	Select Custom in the Color field to display the palette.		

1. Select the Data Source tab.
2. Click Static.
3. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

Requirements	Description	
Valid characters	Full ASCII	
	Numeric	0-9
	Binary data	

DataMatrix uses sophisticated error correction algorithms to insure data integrity.

4. Click Apply.
5. Click OK.

The bar code displays your default specifications.

Option 6.4.16. Configure a LogMars Bar Code

LOGMARS (Logistics Applications of Automated Marking and Reading Symbols) is a:

- Special application of Code 39 used by the U.S. Department of Defense and is governed by Military Standard MIL-STD-1189B.
- One-dimensional bar code type.

rect 19, 272, 141, 359 ([page 1037](#))rect 9, 53, 259, 257 ([page 1036](#))rect 261, 53, 494, 269 ([page 1036](#))

	Option	Description	
1 (page 1035)	Bar Code		
	Width	Bar code width from left to right in points	
	Rotation	Options going counter-clockwise are:	
		0o	No rotation
		90o	Left-Vertical
		180o	Upside down
		270o	Right-vertical
	Bar Color	Color of the bar characters.	
	Background Color	Color of the area behind the bars that is background.	
	Ratio	Range	2.0-3.0
		Default	2.0
	Checksum	Checked	Any checksum that is specific to any implementation has to be done outside the scope of broadcast, and should be part of the data made available to broadcast.
		Default	Clear
2 (page 1035)	Caption		
	Style	Available features for the caption font are: <ul style="list-style-type: none"> • Bold • Italic • Underline • Strike Thru 	
	Font Name	Selected font for the caption. This font can be a standard font (different from the bar code font) so it can be easily read.	
	Point Size	Size in which the caption font will display. The caption font displays within the space allocated for the bar. Therefore, the larger the font the more space it uses and the less space there is to display the bar. You can increase the space for the bar or decrease the font size if the bar is smaller than you want.	
	Location	Caption font position static to the bar. Options are.	

		<ul style="list-style-type: none"> • None • Below Left • Below Center • Below Right • Above Left • Above Center • Above Right
	Color	Color of the caption font.
* (page 1035)	Colors can be customized through the Color palette.	
	Select Custom in the Color field to display the palette.	

1. Select the Data Source tab.
2. Click Static.
3. Make a valid entry in the **Text** box.

Note: Also make sure that values from other data sources adhere to these requirements.

Requirements	Description
Valid characters	Six special alphanumeric characters.
	Capital letters A through D
	All numeric digits
Length	Variable

The Modulus 43 check digit, optional with Code 39, is defined and recommended in the specification.

Option	Description
Check Digit	<p>A modulo 43 check character can be used to enhance data security for Code 39 symbols. The last digit of the symbol is:</p> <ul style="list-style-type: none"> • Assumed to be the check digit. • Compared to a calculated check digit to verify the symbol.


Bar Code Technical Reference

Bar Code Technical Reference

ASCII Table for Code 39
ASCII Table

ASCII Table for Code 39

The table below lists the keystrokes required when generating a Code 39 symbology character.

 **Note:** Most of the ASCII characters listed below have a one-to-one relationship with the bar code character and can be typed directly from the keyboard.

All of the unprintable characters require an Alt+<Key>.

ASCII		
Value	Character	Keyboard
0	NUL	Alt+000
1	SOH	Alt+001
2	STX	Alt+002
3	ETX	Alt+003
4	EOT	Alt+004
5	ENQ	Alt+005
6	ACK	Alt+006
7	BEL	Alt+007
8	BS	Alt+008
9	HT	Alt+009
10	LF	Alt+010
11	VT	Alt+011
12	FF	Alt+012
13	CR	Alt+013
14	SO	Alt+014
15	SI	Alt+015
16	DLE	Alt+016
17	DC1	Alt+017
18	DC2	Alt+018
19	DC3	Alt+019
20	DC4	Alt+020
21	NAK	Alt+021
22	SYN	Alt+022
23	ETB	Alt+023

24	CAN	Alt+024
25	EM	Alt+025
26	SUB	Alt+026
27	ESC	Alt+027
28	FS	Alt+028
29	GS	Alt+029
30	RS	Alt+030
31	US	Alt+031
32	SPACE	Alt+032 (ASCII 32 Reserved)
33	!	!
34	"	"
35	#	#
36	\$	\$ (Alt+0178 to create Standard Code 39 "\$")
37	%	% (Alt+0179 to create Standard Code 39 "%")
38	&	&
39	`	'
40	((
41))
42	*	START/STOP (Alt+0165 to create "**")
43	+	+ (Alt+0176 to create Standard Code 39 "+")
44	,	,
45	-	-
46	.	.
47	/	/ (Alt+0177 to create Standard Code 39 "/")
48	0	0
49	1	1
50	2	2
51	3	3
52	4	4
53	5	5
54	6	6
55	7	7

56	8	8
57	9	9
58	:	:
59	;	;
60	<	<
61	=	=
62	>	>
63	?	?
64	@	@
65	A	A
66	B	B
67	C	C
68	D	D
69	E	E
70	F	F
71	G	G
72	H	H
73	I	I
74	J	J
75	K	K
76	L	L
77	M	M
78	N	N
79	O	O
80	P	P
81	Q	Q
82	R	R
83	S	S
84	T	T
85	U	U
86	V	V
87	W	W

88	X	X
89	Y	Y
90	Z	Z
91	[[
92	\	\
93]]
94	^	^
95	_	SPACE (Alt+0161 to create "_")
96	'	START (Alt+0162 to create "'")
97	a	a
98	b	b
99	c	c
100	d	d
101	e	e
102	f	f
103	g	g
104	h	h
105	i	i
106	j	j
107	k	k
108	l	l
109	m	m
110	n	n
111	o	o
112	p	p
113	q	q
114	r	r
115	s	s
116	t	t
117	u	u
118	v	v
119	w	w

120	x	x
121	y	y
122	z	z
123	{	{
124		
125	}	}
126	~	STOP (Alt+0163 to create "~")
127	DEL	Alt+0166 (ASCII 127 Reserved)

ASCII Table

The entire code set for the American Standard Code for Information Interchange (ASCII) is listed below along with the keystrokes required to enter the ASCII Character.

ASCII		
Value	Character	Keyboard
0	NUL	Alt+000
1	SOH	Alt+001
2	STX	Alt+002
3	ETX	Alt+003
4	EOT	Alt+004
5	ENQ	Alt+005
6	ACK	Alt+006
7	BEL	Alt+007
8	BS	Alt+008
9	HT	Alt+009
10	LF	Alt+010
11	VT	Alt+011
12	FF	Alt+012
13	CR	Alt+013
14	SO	Alt+014
15	SI	Alt+015
16	DLE	Alt+016
17	DC1	Alt+017

18	DC2	Alt+018
19	DC3	Alt+019
20	DC4	Alt+020
21	NAK	Alt+021
22	SYN	Alt+022
23	ETB	Alt+023
24	CAN	Alt+024
25	EM	Alt+025
26	SUB	Alt+026
27	ESC	Alt+027
28	FS	Alt+028
29	GS	Alt+029
30	RS	Alt+030
31	US	Alt+031
32	SPACE	Alt+032
33	!	!
34	"	"
35	#	#
36	\$	\$
37	%	%
38	&	&
39	`	'
40	((
41))
42	*	*
43	+	+
44	,	,
45	-	-
46	.	.
47	/	/
48	0	0
49	1	1

50	2	2
51	3	3
52	4	4
53	5	5
54	6	6
55	7	7
56	8	8
57	9	9
58	:	:
59	;	;
60	<	<
61	=	=
62	>	>
63	?	?
64	@	@
65	A	A
66	B	B
67	C	C
68	D	D
69	E	E
70	F	F
71	G	G
72	H	H
73	I	I
74	J	J
75	K	K
76	L	L
77	M	M
78	N	N
79	O	O
80	P	P
81	Q	Q

82	R	R
83	S	S
84	T	T
85	U	U
86	V	V
87	W	W
88	X	X
89	Y	Y
90	Z	Z
91	[[
92	\	\
93]]
94	^	^
95	_	_
96	'	'
97	a	a
98	b	b
99	c	c
100	d	d
101	e	e
102	f	f
103	g	g
104	h	h
105	i	i
106	j	j
107	k	k
108	l	l
109	m	m
110	n	n
111	o	o
112	p	p
113	q	q

114	r	r
115	s	s
116	t	t
117	u	u
118	v	v
119	w	w
120	x	x
121	y	y
122	z	z
123	{	{
124		
125	}	}
126	~	~
127	DEL	Alt+127

Step 6.5. Position Bar Codes on WYSIWYG Forms

Methods to change the size and position of the text box on the WYSIWYG form include:

- Use the mouse to change the text box location.
- Use the Barcode Object Properties dialog box.

Use the mouse to change the bar code location

1	Click and hold down the right-mouse button anywhere in the text box.
	The cursor changes to cross hairs when the text box is ready to be moved.
2	Hold down the right-mouse button as you drag the cursor on the screen.
	A dotted line frame displays the destination of the box as you drag the cursor.
3	Release the right-mouse button when the box is in the correct location.
	The text box moves to the selected location.

Use the Barcode Object Properties dialog box

1. Select the Position tab in the Barcode Object Properties dialog box.

The field entries reflect the current position of the bar code.

2. Edit any of the positions as follows.

rect 80, 78, 335, 131 ([page 1046](#))

rect 80, 140, 335, 178 ([page 1047](#))

rect 85, 181, 339, 219 ([page 1047](#))

rect 87, 222, 341, 257 ([page 1047](#))

Field	Co-ordinate/Value	Description
Left	X co-ordinate of the starting position	First clicked the mouse after selecting the bar code object.
Top	Y co-ordinate of the starting position	First clicked the mouse after selecting the bar code object.
Width	Delta X value	Difference between the starting position X value and the ending position X value of the rectangle, This defines the width of the bar code object.
Height	Delta Y value	Difference between the starting position Y value and the ending position Y value of the rectangle. This defines the height of the bar code object.

[Up \(page 1047\)](#)

3. Do one of the following.

- a. Click Apply to apply your entries and continue to configure other bar code features.
- b. Click OK to close the Barcode Object Properties dialog box and view your changes.

Bar Code Object Symbology Properties

Select the specific topic describing a bar code object's symbology configuration.

Configure the Bar Code Symbology.
Use Default Configurations for WYSIWYG Form.

Step 7. Insert Graphic Objects for the WYSIWYG Form

Step 7. Insert Graphic Objects on the WYSIWYG Form

Option 7.1 (page 1048)	Add an image object to the WYSIWYG form.
Option 7.2 (page 1054)	Add a rectangle to the WYSIWYG form.
Option 7.3 (page 1058)	Add a line to the WYSIWYG form

*Option 7.1. Use an Image Object on the WYSIWYG Form**Option 7.1. Use an Image Object on the WYSIWYG Form*

Option 7.1.1 (page 1048)	Place an image on the WYSIWYG form.
Option 7.1.2 (page 1050)	Change the image size and position.

*Option 7.1.1. Place an Image on the WYSIWYG Form**Option 7.1.1. Place an Image on the WYSIWYG Form*

Option 7.1.1.1 (page 1048)	Insert an image object on the WYSIWYG form
Option 7.1.1.2 (page 1049)	Copy an image object to the WYSIWYG form.

Option 7.1.1.1. Insert an Image Object on the WYSIWYG Form

1. Do one of the following.

Method 1

- a. Click Draw on the WYSIWYG Text Editor menu bar.
- b. Select Image.

Method 2

Click the Image button on the WYSIWYG Text Editor toolbar.

Method 3

- a. Right-click the WYSIWYG screen.
- b. Select Image.

An Image cursor displays when you use any method.

2. Specify the image location as follows.

- a. Point the cursor where you want the image to be inserted.
- b. Drag the cursor horizontal and vertically as you hold down the right-mouse button.

A rectangle displays based on where you dragged the cursor.

3. Click the left-mouse button.

A Load Image dialog box opens.

4. Find the folder that has the image.
5. Select the image that will be inserted.

Accepted formats are:

- .bmp
- .jpeg
- .jpg
- .gif

The image opens in the image rectangle.

Option 7.1.1.2. Copy an Image Object from one WYSIWYG Form to Another

1. Display the form that has the image to be copied.
2. Select the image to be copied.
3. Do one of the following.

Copy the image

Method 1

Click the Copy button on the WYSIWYG Editor toolbar.

Method 2

- a. Click Edit on the WYSIWYG Editor menu bar.
- b. Select Copy.

Method 3

Press Ctrl+C on the keyboard.

Cut the image

Method 1

Click the Cut button on the WYSIWYG Editor toolbar.

Method 2

- a. Click /edit on the WYSIWYG Editor menu bar.
- b. Select Cut.

Method 3

Press Ctrl+X on the keyboard.

4. Display the form to which the image will be pasted.

5. Do one of the following.

Method 1

Click the Paste button on the WYSIWYG Editor toolbar.

Method 2

- a. Click Edit on the WYSIWYG Editor menu bar.
- b. Select Paste.

Method 3

Press Ctrl+P on the keyboard.

The image and its configured size is copied from the source to the target form. You can change the size and position the same way you would for an inserted graphic.

Option 7.1.2. Change an Image Size and Position on the WYSIWYG Form

Option 7.1.2. Change an Image Size and Position on the WYSIWYG Form

Use either of the following to change the size and position of a selected image.

Option 7.1.2.1 (page 1050)	Change an image size and position through the image properties dialog box.
Option 7.1.2.2 (page 1053)	Change an image size and position using the mouse.

Option 7.1.2.1. Change an Image Size and Position through the Image Properties Dialog Box

When you insert the image on the form it may look distorted. This is because the rectangle in which you inserted the image may not have the same aspect ratio.

- Correct image display.
- Examples.

Correct image display

The Object Properties dialog box enables you to easily correct the image display.

1. Do one of the following to open the Image Object Properties dialog box.

Method 1

Click the Object Properties button on the WYSIWYG toolbar.

Method 2

- a. Click Edit on the WYSIWYG Editor menu bar.
- b. Select Properties.

Method 3

- a. Right-click the image.
- b. Select Properties from the Popup menu.

Method 4

Press Alt+Enter on the keyboard.

Method 5

Double-click the image.

The Image Object Properties dialog box opens when you use any method.

2. Select the Image tab.
3. Options on the image tab are as follows.

Options	Stretch to Fit	Keep Aspect Ratio	Description
	Checked	Checked	Image fits in rectangle in proportion to the aspect ratio of the original graphic
	Checked	Cleared	Image height and width can be changed manually.
	Cleared	Disabled	Image can display in is its actual size
Statistics	Width	Actual image width.	
	Height	Actual image height	

4. Click Apply when you have made your selections.

The image display changes according to your selections.

5. Select the Position tab.

The fields display the size and position the rectangle was in when you opened the Image properties dialog box.

Position and dimension selections are as follows.

Field	Description
Left	X co-ordinate of the left side of the image.
Top	Y co-ordinate of the top of the image.
Width	Width of the image. Difference between the starting position X value and the ending position X-value.
Height	Height of the image. Difference between the starting position Y value and the ending position Y value.

Note: If you checked Keep aspect ratio on the Image tab, Broadcast will:

- Change the height to the dimension you enter in the Height field.
- Change the width to keep the aspect ratio of the original position.

The width that Broadcast makes the image will display in the Width field when you close and re-open the Image Properties dialog box.

6. Click Apply when you make your changes.

7. Click OK to close the Image Properties dialog box.

The image displays in the size and position you selected.

Examples

Example 1
Example 2
Example 3

Example 1

Image is the actual size.

Selections are:

Tab	Feature	Selection
Image	Stretch to Fit	Cleared
	Keep Aspect Ratio	Disabled

[Up \(page 1052\)](#)

Example 2

Image is small, but in proportion to the original.

Selections are:

Tab	Feature	Selection
Image	Stretch to Fit	Checked
	Keep Aspect Ratio	Checked
Position	Height	e.g. 75

[Up \(page 1052\)](#)

Example 3

Image proportion and size are manually selected.

Selections are:

Tab	Feature	Selection
Image	Stretch to Fit	Checked
	Keep Aspect Ratio	Cleared
Position	Height	e.g. 93
	Width	e.g. 200

[Up \(page 1052\)](#)

Option 7.1.2.2. Change an Image Size and Position using the Mouse

- Image position changed using a mouse
- Image size changed using a mouse

Image position changed using a mouse

1. Click the right-mouse button down as you select area of the image that is not a handle ().

The cursor changes to a cross.

2. Hold the right-mouse button down as you drag the image in any direction.

A rectangle outline shows you the position the rectangle will move to when you release the right-mouse button.

3. Release the right-mouse button when the image is in the position you want.

Result: The image moves to the selected position.

Image size changed using a mouse

4. Click the right-mouse button down as you select a handle in the image.

Handles to select are as follows.

Handle	Changes the image:
Left/right	Width
Top/bottom	Height
Corner	Width and height

The cursor changes to a cross-hair.

5. Drag the handle to increase or decrease the selected dimension.

A rectangle outline shows you how the size will change when you release the right-mouse button.

Example 1

.

Example 2

6. Release the right-mouse button when the rectangle is the size you want.

The rectangle displays the size you selected.

! **Important:** If Keep Aspect Ratio is checked in the Image Properties dialog box, when you release the mouse button, the width of the image will resize in proportion to the aspect ratio of the original graphic static to the height you select.

Option 7.2. Create a Rectangle Object

Option 7.2. Create a Rectangle Object

Option 7.2.1 (page 1055)	Place a rectangle on the WYSIWYG form.
Option 7.2.2 (page 1055)	Change a rectangle's size and position on the WYSIWYG form.
Option 7.2.3 (page 1057)	Format a rectangle's border and fill.

Option 7.2.1. Place a Rectangle on the WYSIWYG Form

1. Do one of the following.

Method 1

Click the Rectangle button on the WYSIWYG toolbar.

Method 2

- a. Click Draw on the WYSIWYG menu bar.
- b. Select Rectangle.

Method 3

- a. Right-click the WYSIWYG screen.
- b. Select Rectangle from the Popup menu.

Your cursor turns into a cross-hair with a small rectangle.

2. Click down the right-mouse button where you want the rectangle to start.
3. Drag the cursor vertically and horizontally to increase the size of the rectangle.
4. Release the right-mouse button when the rectangle is the size you want.

The rectangle displays on the screen in the selected location and size.

i Tip : You can cut and paste or copy and paste a rectangle from one WYSIWYG form into another (as described for an [image object \(page 1049\)](#)). The source rectangle configuration is copied to the target.

Option 7.2.2. Change a Rectangle's Size and Position on the WYSIWYG Form

Use either of the following to change a selected rectangle's size and position.

- Use the Object Properties dialog box.

- Use the mouse and cursor.

Object Properties dialog box

1. Do one of the following to open the Rectangle Object Properties dialog box.

Method 1

Click the Object Properties button on the WYSIWYG toolbar.

Method 2

- a. Click Edit on the WYSIWYG Editor menu bar.
- b. Select Properties.

Method 3

- a. Right-click the rectangle.
- b. Select Properties from the Popup menu.

Method 4

Press Alt+Enter on the keyboard.

Method 5

Double-click the rectangle.

The Rectangle Object Properties dialog box opens when you use any method.

2. Select the Position tab.

The fields display the current size and position of the rectangle.

Position and dimension selections are as follows.

Field	Description
Left	X co-ordinate of the left side of the rectangle.
Top	Y co-ordinate of the top of the rectangle.
Width	Width of the rectangle. Difference between the starting position X value and the ending position X-value.
Height	Height of the rectangle. Difference between the starting position Y value and the ending position Y value.

3. Click Apply after you have made your changes.
4. Click OK.

Result: The rectangle displays in the size and position you specified.

Mouse and cursor

5. Change the rectangle's position as follows.

- a. Click the right-mouse button down as you select any area of the rectangle that is not a handle ().

The cursor changes to a cross.

- a. Hold the right-mouse button down as you drag the rectangle in any direction.

A rectangle outline shows you the position the rectangle will move to when you release the right-mouse button.

- a. Release the right-mouse button when the rectangle is in the position you want.

Result: The rectangle moves to the selected position.

6. Change the rectangle's size as follows.

- a. Click the right-mouse button down as you select a handle in the rectangle.

Handles to select are as follows.

Handle	Changes the rectangle's:
Left/right	Width
Top/bottom	Height
Corner	Width and height

The cursor changes to a cross-hair.

- a. Drag the handle to increase or decrease the selected dimension.

A rectangle outline shows you how the size will change when you release the right-mouse button.

1. 2..

- a. Release the right-mouse button when the rectangle is the size you want.

The rectangle displays the size you selected.


Option 7.2.3. Format a Rectangle's Border and Fill

1. Open a selected rectangle's Rectangle Object Properties dialog box.

2. Select the Border tab.

3. Select colors and fill as follows.

Field	Description
Line Color	Rectangle's primary border color. Line color: <ul style="list-style-type: none"> • Displays if you select a solid line. • Is the primary color if you select a dash or dotted line.
Fill Color	Color in the rectangle's area
Background color	Rectangle's secondary border color if a dash or dotted line is selected.
Line width	Width in points.
Line Style	Selections include: <ul style="list-style-type: none"> • Solid line • Variety of dash and dot styles • Blank entry for no line

 **Note:** Colors for all color fields include:

- Basic selections
- Custom color that opens a color palette in which you can define the color you want.

4. Click Apply to apply your changes.

5. Click OK.

The rectangle is formatted with your selections.

Option 7.3. Create a Line on the WYSIWYG Form

Option 7.3. Create a Line on the WYSIWYG Form

Option 7.3.1 (page 1058)	Place a line on the WYSIWYG form.
Option 7.3.2 (page 1059)	Change the line's size and position.
Option 7.3.3 (page 1061)	Format the line color and style.

Option 7.3.1. Place a Line on the WYSIWYG Form

1. Do one of the following.

Method 1

Click the Line button on the WYSIWYG toolbar.

Method 2

- a. Click Draw on the WYSIWYG menu bar.
- b. Select Line.


Method 3

- a. Right-click the WYSIWYG screen.
- b. Select Line from the Popup menu.

Your cursor turns into a cross-hair with a small rectangle.

2. Click down the right-mouse button where you want the line to start.
3. Drag the cursor vertically or horizontally to increase the size of the line.
4. Release the right-mouse button when the line is the size you want.

The line displays on the screen in the selected location and size.

 **Tip:** : You can cut and paste or copy and paste a line from one WYSIWYG form into another (as described for an [image object \(page 1049\)](#)). The source line configuration is copied to the target.

Option 7.3.2. Change a Line's Position and Length on the WYSIWYG Form

Use either of the following to change a selected line's size and position.

- Object Properties dialog box.
- Mouse and cursor.

Object Properties dialog box

1. Do one of the following to open the Line Object Properties dialog box.

Method 1

Click the Object Properties button on the WYSIWYG toolbar.

Method 2

- a. Click Edit on the WYSIWYG Editor menu bar.
- b. Select Properties.

Method 3

- a. Right-click the line.
- b. Select Properties from the Popup menu.

Method 4

Press Alt+Enter on the keyboard.

Method 5

Double-click the line.

The Line Object Properties dialog box opens when you use any method.

2. Select the Position tab.

The fields display the current size and position of the rectangle.

3. Change the dimensions as follows.

Field	Line Angle	Description
Left	Horizontal or angled	X co-ordinate of the left point of the line
	Vertical	Left side of the line width
Top	Horizontal	Y co-ordinate of the top point of the line width
	Vertical or angled	Y co-ordinate of the top point of the line
Width		Length of the line Difference between the starting position X value and the ending position X-value.
Height		Height of the line. Difference between the starting position Y value and the ending position Y value.

4. Click Apply after you have made your changes.

5. Click OK.

Result: The line displays in the size and position you specified.

Mouse and cursor

6. Change the line's position as follows.

- a. Click the right-mouse button down as you select any edge of the rectangle that is not a handle ().

The cursor changes to a cross.

- a. Hold the right-mouse button down as you drag the line in any direction.

A line outline shows you the position the line will move to when you release the right-mouse button.

- a. Release the right-mouse button when the line is in the position you want.

Result: The line moves to the selected position.

7. Change the line's length as follows.

- a. Click the right-mouse button down as you select a handle at either end of the line.

The cursor changes to a cross-hair.

- a. Drag the handle to increase or decrease the selected length.

A line outline shows you how the length will change when you release the right-mouse button.


- a. Release the right-mouse button when the length you want.

The line displays the length you selected.

Option 7.3.3. Format a Line's Color and Style

1. Open a selected line's Line Object Properties dialog box.
2. Select the Line tab.
3. Select colors and style as follows.

Field	Description	
Line Color	Line's primary color. Line color: <ul style="list-style-type: none"> • Displays if you select a solid line. • Is the primary color if you select a dash or dotted line. 	
	Default	Black
Background color	Rectangle's secondary border color if a dash or dotted line is selected.	
	Default	White
Line Width	Width in points.	
	Default	1
Line Style	Selections include: <ul style="list-style-type: none"> • Solid line • Variety of dash and dot styles 	
	Default	Solid

 **Note:** Colors for all color fields include:

- Basic selections
- Custom color that opens a color palette in which you can define the color you want.

4. Click Apply to apply your changes.

5. Click OK.

The line is formatted with your selections.

Step 8. Organize Objects to Create an Easy-to-Read Form

Step 8. Organize Objects to Create an Easy-to-Read Form

Option 8.1 (page 1062)	Move an object to the front or back of another object.
Option 8.2 (page 1063)	Enter option name or description here.
Option 8.3 (page 1065)	Make object sizes the same.
Option 8.4 (page 1066)	Group/Ungroup objects.

Option 8.1. Move an Object to the Front or Back of Another Object

- Move an object to the back of other objects.
- Move an object to the front of other objects

Move an object to the back of other objects

1. Select an object that is in front of one or more objects.
2. Do one of the following.

Method 1

Click the Send to Back button on the WYSIWYG toolbar.

Method 2

- a. Click Layout on the WYSIWYG menu bar.

b. Select Order>Send to back.

Result: The selected object moves to the back.

Move an object to the front of other objects

3. Select an object that is in back of one or more objects.

4. Do one of the following.

Method 1

Click the Bring to Front button on the WYSIWYG toolbar.

Method 2

a. Click Layout on the WYSIWYG menu bar.

b. Select Order>Bring to Front.

The selected object moves to the front.

Option 8.2. Align Objects on the WYSIWYG Form

- Configure alignment.
- Examples

Configure alignment

1. Select the objects you want to align.

2. Do one of the following.

Method 1

Click the appropriate alignment button on the WYSIWYG Action toolbar.

A	Align left
B	Align top
C	Align right
D	Align bottom

Method 2

a. Click Layout on the WYSIWYG menu bar.

b. Select Align><Align option> from the menus.

Method 3

- a. Right-click the selected objects.
- b. Select Align><Align option> from the popup menus.

The objects align according to your selection, static to the object that is the furthest out in the selected position.

Examples

Example 1
Example 2

Example 1

Align left or right

Before

Align left

*	Truck was furthest to the left. Objects aligned static to the truck.
**	Object not selected.

Align Right

*	Item ID was furthest to the right. Objects aligned static to the Item ID.
---	---

[Up \(page 1064\)](#)

Example 2

Align top or bottom

Before

Align top

*	Truck was at the top. Objects aligned static to the truck.
---	--

Align bottom

*	Date & Time was at the bottom. Objects aligned static to Date & Time.
---	---

[Up \(page 1064\)](#)

Option 8.3. Make Object Sizes the Same

1. Select the objects that should have the same dimensions.

Note: The dimensions of object selected last will be the dimensions applied to other selected objects.

A	Object selected first.
B	Object selected second.
C	Object selected last.
*	Truck has Keep Aspect Ratio (page 1051) checked.
	The truck's horizontal only size will not change using this option.

2. Do one of the following.

Method 1

Click the appropriate size button on the WYSIWYG Action toolbar.

A	Same width
B	Same height
C	Same width and height

Method 2

- a. Click Layout on the WYSIWYG menu bar.
- b. Select Size><Size option> from the menus.

Method 3

- a. Right-click the selected objects.
- b. Select Size><Size option> from the popup menus.

The objects are resized according to your selection.

Examples

- Example 1
- Example 2
- Example 3

Example 1

Same width

*	Truck has Keep Aspect Ratio (page 1051) checked.
	Since the aspect ratio is based on the image height, the truck's size did not change using this option.

[Up \(page 1065\)](#)

Example 2

Same height

*	Truck has Keep Aspect Ratio (page 1051) checked.
	Since the aspect ratio is based on the image height, the truck's size changed in proportion to the actual image.

[Up \(page 1065\)](#)

Example 3

Same height and width

*	Truck has Keep Aspect Ratio (page 1051) checked.
	Since the aspect ratio is based on the image height, the truck's size changed in proportion to the actual image; the width does not conform to the selected width.

[Up \(page 1065\)](#)

Option 8.4. Group/Ungroup Objects

- Group objects.
- Ungroup objects.

Group objects

1. Select the objects you want to group.

Note: If objects are in front of or in back of each other, the objects you select last will be in the front when they are grouped.

A	Object selected first.
B	Object selected second.

C	Object selected last.
---	-----------------------

2. Do one of the following.

Method 1

Click the Group button on the WYSIWYG Layout toolbar.

Method 2

- a. Click Layout on the WYSIWYG menu bar.
- b. Select Group.

Method 3

- a. Right-click the selected objects.
- b. Select Group on the popup menu.

circle 116, 36, 23 [\(page 1066\)](#)

circle 46, 64, 25 [\(page 1066\)](#)

circle 289, 41, 27 [\(page 1067\)](#)

A message displays telling you that grouping the objects will disable the Aspect Ratio function.

3. Click Yes to group the objects.

Result: The objects are grouped when you use any method. You can change the size or position of the group the same way you would with a single object. The aspect ratio function is disabled.

Ungroup objects

4. Select the group you want to ungroup.

5. Do one of the following.

Method 1

Click the Ungroup button on the WYSIWYG Layout toolbar.

Method 2

- a. Click Layout on the WYSIWYG menu bar.
- b. Select Group.

Method 3

- a. Right-click the group.
- b. Select Ungroup on the popup menu.

The objects in the group are separated when use any method.

Step 9. Compile a WYSIWYG Form

1. Right-click **WYSIWYG Forms**.
2. Select Compile on the popup menu.

The WYSIWYG Form Compile dialog box opens.

3. Click the Open button to the right of the **WYSIWYG Form** field.

The Select the WYSIWYG Form dialog box opens.

4. Open the <Project Name>\CIMForms\WYSIWYGForms\Working folder.
5. Select the file you want to compile.

The file type is

CimForms WYSIWYG Form files (*.cwf)	Files that were created and saved or were edited and saved as *.cwf files in the in the WYSIWYG Text Editor.
-------------------------------------	--

6. Click Open.

The path and file name display in the **WYSIWYG Form** field in the WYSIWYG Form Compile dialog box.

7. Click OK.

Broadcast compiles the WYSIWYG form and reports any errors that it finds.

If there are no errors a message displays telling you that the compilation was successful.

Step 10. Test Print the WYSIWYG Form

1. Click the Start Project button on the Workbench toolbar.
2. Open the TrackerCfg_UI.
3. Expand the CimForms folder in the TrackerCfg_UI left pane.
4. Right-click WYSIWYG Forms.
5. Do one of the following.

- a. Open the WYSIWYG Form Test dialog box from the Popup menu.

Select Test from the Popup menu.

- a. Open the WYSIWYG Form Test dialog box in the WYSIWYG Editor.

Select File>Test Print on the WYSIWYG Editor toolbar.


A WYSIWYG Form Test dialog box opens when you use either method.

6. Fill in the fields as follows.

Field	Description
WYSIWYG Form	Form in the CimForms\WYSIWYGForms\Working folder that will be tested.
Item ID	Item whose data the form will print.
Device To test	Printer that will print the form.

7. Click OK.

The form will print the data for the selected item in the configured data source fields.

 **Note:** Messages will alert you to possible problems; if the device is not configured correctly the form will not print.

Step 11. Add/Publish a WYSIWYG Form

Step 11. Add/Publish a WYSIWYG Form

1. Right-click WYSIWYG Forms in the TrackerCfg_UI left pane.
2. Select Add/Publish on the popup menu.

The Add /Update WYSIWYG Form dialog box opens.

3. Click the Open button to the right of the **WYSIWYG Form** field.

The Select the WYSIWYG Form dialog box opens.

4. Open the <Project Name>\CIMForms\WYSIWYGForms\Working folder.

5. Select the file you want to compile.

The file type is

CimForms WYSIWYG Form files (*.cwf)	Files that were created and saved or were edited and saved as *.cwf files in the WYSIWYG Text Editor.
-------------------------------------	---

6. Click Open.

The Add /Update CimForms WYSIWYG Forms dialog box displays the path and file you selected.

The options are as follows.

Option	Description
WYSIWYG Form	Path and file name are automatically filled in when the file is selected in the Select the WYSIWYG Form dialog box. Path and filename can also be typed in.
Save Form As	Check to be able to change the name of the form.
	Broadcast saves the file as a .cwf file. You do not need to include the file extension.

7. Click OK.

An WYSIWYG Form Properties dialog box opens.

8. The ASCII Form Properties dialog box provides the following information.

rect 8, 24, 434, 47 ([page 1070](#))

rect 8, 47, 433, 74 ([page 1070](#))

rect 10, 71, 454, 151 ([page 1070](#))

rect 14, 151, 454, 177 ([page 1070](#))

rect 10, 177, 455, 197 ([page 1070](#))

Field	Description	
Form Name	Name of the selected file or of the name entered in the Save As field.	Read only
Form Version	Count of the number of times the file has been published.	Read only
Form Description	Description that will help users know the content and intended use of the added file.	Read/Write
Published Date and Time	Last time the file was published.	Read only
Published User	Windows logged in User ID	Read only

9. Click OK.

Result: When you add/publish a WYSIWYG form the Broadcast:

- a. Adds the file to the list of published files in the TrackerCfg_UI right pane.
- a. Saves the file in the <Project Name>\CIMForms\WYSIWYGForms\prod folder.

10. You can update or delete a published form.

Option 11.1 (page 1071)	Update a published WYSIWYG form.
Option 11.2 (page 1072)	Delete a published WYSIWYG form.

Option 11.1. Update Published WYSIWYG Forms

1. Expand CimForms in the TrackerCfg_UI.
2. Right-click WYSIWYG Forms.
3. Select View from the Popup menu.

The published forms display in the right pane.

4. Right-click the form you want to update.
5. Select Update from the Popup menu.

An add / Update CimForms WYSIWYG Forms dialog box displays the name of the file in the Working folder that was used to publish the selected file.

6. Do one of the following.

Use the current version of the file that was used before.

Click OK.

Use a different file

- a. Click the Popup menu button to the right of the **WYSIWYG Form** field.
- b. Find the CimForms\WYSIWYGForms\Working folder.
- c. Select the file you want to use.

The selected file name and path will display in the **WYSIWYG Form** field.

- a. Click OK.

A WYSIWYG Form Properties dialog box displays.

7. Enter the latest information in the **Form Description** field.
8. Click OK.

Broadcast does the following:

- Updates the selected file in the CIMForms\WYSIWYGForms\prod folder.
- Creates backup copies of the updated form in the CIMForms\WYSIWYGForms\old folder.

Option 11.2. Delete a Published WYSIWYG Form

1. (If the CIMPLICITY project is running) click the Stop Project button on the Workbench toolbar.
2. Open the TrackerCfg_UI.
3. Expand the CimForms folder.
4. Right-click WYSIWYG forms in the TrackerCfg_UI left pane.
5. Select View from the Popup menu.
6. Right-click the form you want to delete in the TrackerCfg_UI right pane.
7. Select Delete from the Popup menu.

A message asks you to confirm deletion.

8. Click Yes.

A message asks you if you want to delete all backup files.

9. Click:

Yes	To delete the published and back up files.
No	To delete only the published file.

WYSIWYG Properties Dialog Boxes

Review properties for:

Text object properties.
Bar code properties.
Change an image size and position.
Change a rectangle's size and position.
Change a line's position and length.

Border Properties for WYSIWYG Form Objects

Click the graphic to select the specific topic describing an object's border configuration.

Option 5.4.2. Configure the Text Object Border (page 994)	Text Object Border
Option 7.2.3. Format a Rectangle's Border and Fill (page 1057)	Rectangle's Border and Fill
Option 4.4.1. Use Default Configurations for WYSIWYG Form Objects (page 975)	Default Configurations for WYSIWYG Form Objects

Line Color and Style Configuration for WYSIWYG Form Objects

Select the specific topic describing an object's border configuration.

Option 7.3.3. Format a Line's Color and Style (page 1061)	Line's Color and Style
Option 4.4.1. Use Default Configurations for WYSIWYG Form Objects (page 975)	Default Configurations for WYSIWYG line Objects

Font Configuration for WYSIWYG Form Objects

Click the graphic to select the specific topic describing an object's border configuration.

Option 5.4.3. Select the Text Object Font (page 995)	Text Object Font
Option 4.4.1. Use Default Configurations for WYSIWYG Form Objects (page 975)	Default Configurations for WYSIWYG Form Objects

Select Documentation For

Select documentation to:

Create a Control Character Token file.
Select a data source for a text object.
Select a data source for a bar code object..

Broadcast Device Groups*Broadcast Device Groups*

Broadcast enables you to create device groups in which you can select primary and secondary devices printers for Broadcast use. You can then use these groups when you configure BCO function blocks or through the Broadcast Queue Monitor.

Advantages of using device groups include:

- The printer configuration dialog enables you to associate available resources in a CIMPPLICITY project with a given printer.
- Because the printers in the groups are conveniently listed in the Workbench right pane, you can make sure that the selections adhere to the [valid primary/secondary \(page 1076\)](#) combinations.

Steps to create a device group include:

Step 1 (page 1074)	Create a new Broadcast Device Group
Step 2 (page 1075)	Configure the Device Group

You can create as many device groups as you need as long as they adhere to the valid combinations. You can also modify or delete them, as needed.

The following occurs if the primary device is not available for broadcast and a secondary device is selected.

1. The status bit of the Primary device is set to indicate that it is not available for broadcast.
2. Broadcast is sent to the secondary.
3. All subsequent broadcasts are directed to the secondary device until one of the following occurs.
 - If the secondary device also fails Broadcast:
 - Checks if the primary is online again.
 - Resets the status bit back to available.
 - Sends the job to the primary.

Note: If the primary is still down, Broadcast raises an alarm.

- The user manually switches back to the primary.

Step 1. Create a new Broadcast Device Group

1. Select **Equipment** folder>**Broadcast Device Groups** in the Workbench left pane.
2. Do one of the following.

A	Click File>New>Object on the Workbench menu bar.	
B	Click the New Object button on the Workbench toolbar.	
C	In the Workbench left pane:	
	Either	Or

	Double click Broadcast Device Groups .	a. Right-click Broadcast Device Groups . b. Select New on the Popup menu.
D	a. In the Workbench right pane. a. Right-click any Broadcast device group. b. Select New on the Popup menu.	
E	Press Ctrl+N on the keyboard.	

A New Device Group dialog box opens.

3. Right-click **Broadcast Device Groups**.
4. Select New on the Popup menu.
5. Right-click any Broadcast device group.
6. Select New on the Popup menu.
7. Enter a name to identify the device group in the **Name** field.
8. Click OK.

A Device Group dialog box opens.

Step 2. Configure the Device Group

- Device Group dialog box fields.
- Valid printer combinations.

Device Group dialog box fields

Make selections in the Device Group dialog box as follows.

Field	Description
Primary Device	<ul style="list-style-type: none"> • Primary device to which Broadcasts sends the data. The device can be a printer or online print driver (e.g. Acrobat distiller). • Selections must adhere to valid combinations (page 1076). • Do either of the following. <ul style="list-style-type: none"> • Select a printer that is listed in the list box. • Enter a valid printer. <p>Note: The Primary Device field accepts an unlimited number of characters.</p>

Secondary Device	<ul style="list-style-type: none"> • (Optional) Backup for the primary device. • Do either of the following. • Select a printer that is listed in the list box. • Enter a valid printer. <p>Note: The Secondary Device field accepts an unlimited number of characters.</p>	
Re-queue count	Jobs to be re-queued when a primary printer fails.	
Report	Report that will be regenerated for auto-queue jobs. The three selections are:	
	No report.	No report will be generated.
	Common Report	A common report will be generated for all the re-queued and re-directed jobs.
	Report per job	A report is generated for each job that is re-queued and redirected.
Resource ID	A physical or conceptual unit to which the printer group should be assigned.	

Valid printer combinations

- Valid two printer configuration

The printers can trade places, primary vs. secondary.

Printer Group Name	Primary Printer	Secondary Printer
PG1	A	B
PG2	B	A

- Valid multiple printer configurations

The same printer can be a secondary printers in more than one group; the primary printers are different.

Printer Group Name	Primary Printer	Secondary Printer
PG1	A	B
PG2	C	B

One printer can be a primary for one printer, a secondary for another printer.

- If printer **A** in PG1 fails the jobs will be directed to **B**.
- If printer **B** fails in PG1 the job will not be directed to **C**.

Printer Group Name	Primary Printer	Secondary Printer
PG1	A	B

PG2	B	C
-----	----------	---

- Illegal printer configuration

The same printer cannot be a primary printer in two different groups.

Printer Group Name	Primary Printer	Secondary Printer
PG1	A	B
PG2	A	C

Broadcast Function Blocks

About Broadcast Function Blocks

RCO provides you with Broadcast (BCO) function blocks to direct printing of the published forms.

- Function blocks
- Example: Broadcast Form Site configuration

Function Blocks

Blocks include:

- Get ASCII Form
- Get ASCII Form By Variable
- Get WYSIWYG Form
- Get WYSIWYG Form By Variable
- Send ASCII Form
- Send ASCII Form BY Variable
- Send Form to Device
- Send WYSIWYG Form
- Send WYSIWYG Form By Variable
- Set Point with ASCII Data
- Set Point with ASCII Data Based on Region

Example: Broadcast Form Site Configuration

Following is an example for configuring a Broadcast Form site.

Step 1 (page 1078)	Create a Broadcast Form site.
Step 2 (page 1078)	Create a new trigger sequence.

Step 3 (page 1079)	Create a new decision.
Step 4 (page 1081)	Test the broadcast form.

Step 1. Create a Broadcast Form Site

1. Right-click the top folder in the TrackerCfg_UI.
2. Select the following on the popup menu.
 - a. Select Lock
 - b. Select New>Folder to create a new folder.

A new folder is created.

3. Name the folder, e.g. BCO_SITE.
4. Create a new site (routing control object) as follows.
 - a. Right-click **BCO_SITE**.
 - b. Select New>Routing Control Object.

A new site is created.

5. Configure the following fields.

	Field	Description
A	Name	Name of the new Routing Control Logic.
B	Resource ID	Resource ID used by the Routing Control Logic.
C	RLM Point	The RLM point must be:
		<ul style="list-style-type: none"> • The same name as your Routing Control Object • A virtual point • A Text point.

Step 2. Create a New Trigger Sequence

1. Create a new trigger sequence as follows.
 - a. Right-click Triggers.
 - b. Select New Trigger Sequence from the popup menu.
2. Name the sequence, e.g. START_PRINTQ.
3. Click the New button on the **Trigger Points** box toolbar.

4. Trigger Details dialog box opens.
5. Click the Popup button to the right of the **Trigger Point** field.
6. Select New on the popup menu.

A New Point dialog box opens.

7. Create a new point as follows.
 - a. Name the point, e.g. START_PRINTW.
 - b. Make the point Virtual, Boolean.
8. Click OK.

The Point Properties dialog box opens.

9. Enter a description on the General tab, e.g. Trigger point printing WYSIWYG form CO.
10. Click OK.

The point displays in the **Trigger Point** field in the Trigger Details dialog box.

11. Check On Update.
12. Click OK.

The trigger displays in the Trigger Points box.

Step 3. Create a New Decision

1. Right-click Decisions routing control object folder, e.g. under BCO_RCO1.
2. Select New Decision from the Popup menu.

Configure the Decision pane as follows.

	Field	Description
A	Decision Name	Name of the new decision.
B	Source	Region that is the source of the order item.
C	Destination	Region that is the destination of the order item.

3. Right-click the Popup menu button to the right of the **Output Logic Module** field.
4. Select New on the Popup menu.

An Output Logic Wizard opens.

5. Enter a name, e.g. Send W forms.
6. Enter a description, e.g. Send WYSIWYG Forms to printer.
7. Create a Get WYSIWYG Form function block as follows.
 - a. Click the New button on the **Function Blocks** box toolbar.

The Select a Function Block browser opens.

- a. Expand the BCO folder.
- b. Select Get WYSIWYG Form.
- a. Click OK.

A Get WYSIWYG Form dialog box opens.

- a. Fill in the fields as follows.

Field	Description
WYSIWYG Form Name	The name of the WYSIWYG form that is merged with the data that has been fetched, e.g. WYSIWYGForm
Region ID	ID of the region where the item is located, e.g. \$OM_MASTR_BLND
Region Location	Location of the item in the selected region.
Item Class	Class of the item, e.g. ORDER. Note: If Item Class is blank all items in the selected Region, Region Location and of the selected Item Type will be broadcast.
Item Type	Type of the item, e.g. PC ORDER. Note: If Item Type is blank all items in the selected Region, Region Location and of the selected item class will be broadcast.

- a. Click OK.

Get WYSIWYG Form displays in the Function Blocks list.

8. Create a Send WYSIWYG Form function block as follows.
 - a. Click the New button on the **Function Blocks** box toolbar.

The Select a Function Block browser opens.

- a. Expand the BCO folder.
- b. Select Send Form to Device.
- a. Click OK.

A Send Form to Device dialog box opens.

- a. Fill in the fields as follows.

Note: Complete details are available in the Send Form to Device function block description.

Field	Description
Device Group	Device group (page 1073) to which Broadcast sends the data. The device group includes a primary and, optionally, a secondary computer.
Priority	The order in which the form should be broadcast static to other forms in the queue. Note: If the user does not specify any job priority the default priority will be used to put the job in the device queue. Default priority is by broadcast type. Accepted values: 0-99 Highest priority: 99
Status Point Id	The status point must: <ul style="list-style-type: none"> • Be a STRING_20 point array • Have the number of elements greater than 2 times the maximum number of items that may be broadcast at one time. A status point get the statuses of broadcast; the point can be used in CIMPLICITY Screens for monitoring purposes. Statuses that are updated to this status point are: <ul style="list-style-type: none"> • Item Id Broadcasted • Success/Failure Of Broadcast (1/0) Where: 1 = Success 2 = Failure

a. Click OK.

Get WYSIWYG Form and Send WYSIWYG Form display in the Function Blocks list.

- a. Add any other RCO function blocks that you want to include in the OLM.
- b. Click Compile.
- c. Click OK to close the Output Logic Wizard.

Step 4. Create a Routing Logic Module

1. Right-click the BCO's Routing Control option logic in the TrackerCfg_UI left pane.
2. Select New Logic Module on the popup menu.
3. Name the logic module.
4. (Simplest configuration) Check Default Script.

Note: As needed, you can use any function block or script that you can use for other function blocks

5. Click Compile.

A default routing logic module is created. The BCO block is configured.

Broadcast Queue Manager

Broadcast Queue Manager IIS Server

About the Broadcast Queue Manager

The Broadcast Queue Manager UI is a Broadcast-specific Web interface to the Windows Queue Manager. Broadcast Queue Manager pages provide a broadcast specific [Broadcast Queue Manager User Interface \(page 1084\)](#) (UI) to the Windows Queue Manager.

! **Important:** If you are configuring a Broadcast Queue Manager server, you must configure the IIS Server as described in this documentation.

Review:

- Broadcast Queue Manager User Interface.
- Broadcast Queue Manager job file locations.

Step 1. Installing the IIS Server

1. Open Server Manager. You can click the icon on the bottom toolbar or you can click the icon to open Server Manager from the Start page. Server Manager opens to the Dashboard page by default.
2. Click **Local Server** from the left panel.
3. Click **Manage** in the upper-right corner, then select **Add Roles and Features**.
4. In the Add Roles and Features Wizard:
5. Review the Before you Begin page and ensure the identified prerequisites are completed, then click **Next**.
6. In the Select installation type dialog, ensure that **Role-based or feature-based installation** is selected, then click **Next**.
7. Select the destination server from the **Server Pool** section, then click **Next**.
8. In the Select server roles dialog, under **Roles**, scroll to and select **Web Server (IIS)**, then click **Next**.
 - e. Ensure the **Include management tools** check box is selected, then click **Add Features**.
 - f. When the Select server roles dialog reappears, click **Next**.
 - g. From the left panel in the Select role services dialog, select **Role Services** under **Web Server Role (IIS)**, then under **Role services** on the right, scroll to and select **Application Development > ASP**.

When you select the **ASP** check box, the following dialog opens immediately:

- h. Ensure the **Include management tools** check box is selected, then click **Add Features**.
- i. When the Select role services dialog reappears, click **Next**.
- j. To confirm your selections and begin the installation, click **Install** in the Confirm Installation selections dialog.

The Installation progress page appears with a progress status bar. When the installation is complete, a success message appears below the bar.

- k. Click **Close**.

Step 2. Configuring Application Pools

Complete the steps below to create one or more application pools on the IIS Server.

1. Click the icon on the bottom toolbar, then click **Control Panel** on the Start page.
2. Click **Administrative Tools**, then select **Internet Information Services (IIS) Manager**. If you receive the following pop-up message, click **Cancel**.
3. When the Internet Information Services (IIS) Manager page appears, right-click **Application Pools** in the left panel and select **Add Application Pool**.
4. In the Add Application Pool pop-up, in the **Name** field, enter a name for your App Pool. In this example, **BQM App Pool** is used.
5. From the **.NET CLR version** drop-down list, select **No Managed Code**.
6. From the **Managed pipeline mode** drop-down list, select **Classic**.
7. Ensure the **Start application pool** immediately check box is selected.
8. Click **OK**. Your App Pool appears in the **Application Pools** list.
9. Right-click on your App Pool name (**BQM App Pool** in this example) and select **Advanced Settings**.
10. For **Enable 32-Bit Applications**, select **True** from the drop-down list, then click **OK**.
11. Continue to Adding a Virtual Directory.

Step 3. Adding a Virtual Directory

1. From the left panel expand **Sites**, then right-click **Default Web Site** and select **Add Virtual Directory**.
2. In the Add Virtual Directory dialog, enter **bqm** in the **Alias** field, and then navigate and select the following path to display the path in the **Physical Path** field:

```
C:\Program Files (x86)\Proficy\Proficy
  SIMPLICITY\Web Pages\bqm
```

3. Click **Connect as** to configure credentials for users with access to the **bqm** folder.
4. Click **Set** to open the Set Credentials dialog, then enter the specified user information and Click **OK**.
5. (Optional) When the Add Virtual Directory dialog reappears, click **Test Settings**.

NOTE: If the user name is not recognized, or if you enter an incorrect password, you receive an error message.

6. Select **Default Web Site** from the left panel, then click **Basic Settings** under **Actions** on the far right.
7. In the Edit Site dialog, for **Application Pool**, click **Select** and choose the BQM application pool (**BQMAppPool** in this example).
8. (Optional) Click **Test Settings** to validation user access.
9. Click **OK**.
10. Open a browser and enter the URL for the BQM login page (for example, <http://localhost/bqm/main/login.asp>).

The BQM login page should open in the browser.

11. Enter the **User ID** and **Password** from Step 4, then select a project and click **Login**.


The Broadcast Queue Manager user interface should open in your browser.

Broadcast Queue Manager User Interface

Broadcast Queue Manager User Interface

Step 1 (page 1085)	Log in to the Broadcast Queue Manager UI
------------------------------------	--

Step 2 (page 1085)	Configure job fields.
Step 3 (page 1087)	Work with devices through the Broadcast Queue Manager UI.
Step 4 (page 1096)	Work with jobs through the Broadcast Queue Manager UI.
Step 5 (page 1111)	Send an Adhoc Broadcast through the Broadcast Queue Manager UI.

 **Important:** You must trigger a Broadcast site in your project at least once in order to open the Broadcast Queue Manager. This will alert the Broadcast Queue Manager that the project actively uses Broadcast.

Step 1. Log in to the Broadcast Queue Manager UI


1. Open Internet Explorer.
2. Enter the URL for the server that hosts Broadcast Queue Manager.
3. Enter the **User Id** and **Password** assigned by your system administrator.
4. Select the project that you want to monitor in the **Select Project** drop down list.

The list includes all of the Broadcast projects that are currently running on your network.

5. Click Login.

The BQM View Devices page opens.

Step 2. Configure Job Fields

 **Note:** This step is required the first time you use the Broadcast Queue Manager. After that, you will only need to open it to modify the list of job fields.

You can categorize jobs to help users identify groups of jobs that are printed by the Broadcast Queue Manager. You enter names based on your project's Tracker attributes and extended attribute names.

1. Click Home Page on the Broadcast Queue Manager menu bar.
2. Select Configure Fields.

A Configure Job Fields page displays.

3. Enter a name in the **New Field** field.
4. Click Add Field.

A message asks you if you are sure you want to configure.

5. Click OK.

The field name is added to the genFld.cfg file that is in the Program Files\Proficy\Proficy CIMPLICITY directory.

Configure Job Fields features are as follows.



rect 65, 302, 136, 323 [\(page 1086\)](#)

rect 155, 29, 192, 226 [\(page 1086\)](#)

rect 11, 276, 193, 297 [\(page 1086\)](#)

rect 20, 27, 156, 224 [\(page 1086\)](#)

rect 44, 228, 157, 249 [\(page 1086\)](#)

Option	Description
Field name (1, 2, 3...)	Names for the job categories. Names can be Tracker attributes or extended attributes. When you do a search to find specific forms, you can enter criteria for one of these fields.  Note: Field names cannot have spaces. Example A Tracker attribute is COLOR You enter COLOR as a field in the Configure Job Fields box. You can then search for printed, historic and/or archived forms that contain selected colors, e.g. all forms in which COLOR = BLUE.
Delete	Check to delete the corresponding field when you click Modify/Delete.
Modify/ Delete	Click to do the following. <ul style="list-style-type: none"> • Update the system with changes made to a Field Name field. • Delete fields that have the Delete check box checked.
New Field	Enter a Tracker attribute or extended attribute in the field that is not already on the list.  Important: Names cannot have spaces.
Add Field	Click to add the entered name to the: <ul style="list-style-type: none"> • List in the Broadcast Queue Manager • GenFld.cfg file in the project's RCO directory.

[Up \(page 1086\)](#)

*Step 3. Work with Devices through the Broadcast Queue Manager**Step 3. Work with Devices through the Broadcast Queue Manager*

Step 3.1 (page 1087)	View devices configured on the IIS server.
Step 3.2 (page 1088)	Configure devices through the Broadcast Queue Manager UI
Step 3.3 (page 1091)	Reset devices.
Step 3.4 (page 1092)	Redirect a device.
Step 3.5 (page 1093)	Work with grouped devices.
Step 3.6 (page 1095)	Pause/Resume Devices

Step 3.1. View Broadcast Device Groups

View Devices page can be used to view the status of all the Devices that are primary devices selected in the [Broadcast Device Groups \(page 1073\)](#) .

1. Do one of the following to open the View Devices web page.
 - Log on to the Broadcast Queue Manager Web site.
 - Click Devices>Device Groups on the Broadcast Queue Manager toolbar.

The View Device Groups Web page opens when you use either method.

Information on the View Device Groups Web page is as follows.

rect 9, 218, 146, 304 [\(page 1087\)](#)
 rect 243, 152, 463, 174 [\(page 1088\)](#)
 rect 345, 216, 438, 302 [\(page 1088\)](#)
 rect 438, 218, 607, 298 [\(page 1088\)](#)
 rect 604, 220, 703, 300 [\(page 1088\)](#)
 rect 9, 218, 146, 304 [\(page 1087\)](#)

Selection	Procedure
Device Group	Device groups that have been defined in the Workbench Broadcast Device Groups (page 1073) .

Primary Device	Primary device selected for the Broadcast device group.
Status	Status of primary device. Includes resumed, no toner, user interruption, out-of-memory, etc.
Secondary Device	Secondary device selected for the Broadcast device group.
Status	Status of secondary device.

2. Reset the device group sequence numbers as follows.

a. Select the device group(s) you want to reset.

Nonconsecutive devices	Hold down the Ctrl key while you select the devices.
Consecutive devices	<ul style="list-style-type: none"> a. Hold down the Shift key. b. Select the first and last device in the group.

a. Click Reset Sequence Number.

A message asks to confirm that you want the printer(s)' sequence number to be reset..

a. Click OK.

A web page displays the printers whose sequence number has just been reset.

a. Click VIEW DEVICE GROUPS.

The View Devices web page re-opens.

Step 3.2. Configure Devices through the Broadcast Queue Manager UI

Step 3.2. Configure Device Groups through the Broadcast Queue Manager UI

1. Click Devices on the Broadcast Queue Manager UI menu bar.

2. Select Configure Devices Groups.

A Configure Devices Web page opens.

Data displayed on the web page includes the following.

Note: The first time you open the page the values that have not been set will be blank.

rect 13, 217, 141, 358 [\(page 1088\)](#)

rect 144, 213, 307, 363 [\(page 1089\)](#)

rect 305, 215, 473, 359 [\(page 1089\)](#)

rect 474, 214, 593, 359 [\(page 1089\)](#)

Data	Descriptions
Device Group	Devices that have been configured in a device group. Double-click any device group to open a Set Configuration Web page and change the displayed settings.

Active Queue Limit	Low	Broadcast raises an alarm when the number of jobs in the queue exceed the low limit.
	High	Broadcast raises an alarm when the number of jobs in the queue exceed the high limit.
History Queue Limit	Low	Broadcast raises an alarm when the number of jobs in the queue exceed the low limit.
	High	Broadcast raises an alarm when the number of jobs in the queue exceed the high limit.
TimeOut (sec.)		Length of time (seconds) that a job will stay in the queue before an alarm is generated.
Port		Port to which the printer is connected.

Configuration options on the Configure devices web page are as follows.

Option 3.2.1 (page 1089)	Change the configuration for a selected printer group
Option 3.2.2 (page 1090)	Clear the configuration for all printers.
Option 3.2.3 (page 1091)	Set the configuration for all printers.

Option 3.2.1. Change the Configuration for a Selected Device Group

1. Double-click a printer name on the Configure Device Groups page.

The Set Configuration page opens displaying the current settings for the selected device group.

2. Change any of the settings as follows.

rect 6, 110, 370, 146 [\(page 1090\)](#)

rect 5, 191, 354, 227 [\(page 1090\)](#)

rect 6, 36, 222, 72 [\(page 1089\)](#)

rect 3, 239, 220, 275 [\(page 1090\)](#)

rect 3, 271, 219, 307 [\(page 1090\)](#)

rect 2, 303, 222, 339 [\(page 1090\)](#)

	Field	Description
	Device Group	Device group that was selected either: <ul style="list-style-type: none"> • On the Configure Device Group Web page. • From the drop-down list of device groups

Up (page 1089)	Active Queue Limit	Low...Jobs	A level where the active queue is starting to have too many jobs. Broadcast will raise an alarm when this number is reached to alert an operator that the queue should be watched in case the number gets higher and jobs need to be, for example, redirected to another printer, archived or deleted immediately.
Up (page 1089)		High...Jobs	The highest or close to the highest number of jobs that should be in the active queue. Broadcast will raise an alarm when this number is reached to alert an operator that the queue needs attention immediately, for example jobs should be redirected to another printer, archived or deleted immediately.
	History Queue Limit	Low...Jobs	A level where the history queue is starting to have too many jobs. Broadcast will raise an alarm when this number is reached to alert an operator that the queue should be watched in case the number gets higher and jobs need to be, for example, archived or deleted immediately.
		High...Jobs	The highest or close to the highest number of jobs that should be in the history queue. Broadcast will raise an alarm when this number is reached to alert an operator that the queue needs attention immediately, for example jobs should be, archived or deleted immediately.
Up (page 1089)	Timeout...seconds		Length of time (seconds) that a job will stay in the queue before an alarm is generated.
	Primary Device Port		Port through which the primary device will print. Note: If you change the port in the Broadcast Queue Manager, it will be change in the printer's Properties dialog box.
	Secondary Device Port		Port through which the secondary device will print.

3. Click Update Configuration.

A message displays asking for confirmation that the changes should be made.

4. Click OK.

Broadcast Queue Manager updates the configuration.

A View Device Configuration page displays the changes that were made.

5. Click View Device Configuration.

The View Device Groups page displays the changed values for the selected printer.

Option 3.2.2. Clear the Configuration for all Device Groups

1. Click Clear Configuration on the Configure Device Groups page.

A message asks you to confirm that you want to clear the configuration of all the devices.

2. Click OK.

Broadcast clears all of the configuration.

Option 3.2.3. Set the Configuration for all Printers

1. Click Set Configuration on the Configure Device Groups page.

A Set Configuration For all Devices page with empty fields opens.

2. Fill in each field to set the limits and time out for every device available to the IIS server.

Note: You can change the settings for any individual printer at any time.

3. Click Update Configuration.

Broadcast configures every device with the new universal entries.

Step 3.3. Reset Devices

A device needs to be reset after it has been paused for Broadcast Queue Manager to send it forms.

Broadcast Queue Manager bypasses the device after:

- The device went down and the Broadcast Queue Manager automatically changed its status to Paused then back to Resumed.
- You manually changed the device to Paused, even if you changed the status back to Resumed.

1. Click Devices>Reset Devices on the Broadcast Queue Manager menu bar.

A Reset Device page opens.

2. Select the Group to reset.

If printers in a group need to be reset that group will be listed in the **Group** field.

3. Check either:

Both Devices	Resets both devices so they will print forms according to the group specifications; the primary device will once again have priority when jobs need to be printed.
Secondary Device	Resets the secondary device only.

4. Click Reset Device.

A message asks you if you are sure you want to reset the device.

5. Click OK.

Broadcast Queue Manager reports what device group was reset.

6. Do one of the following.

- Click View Device Groups to open the View Device Groups page.
- Click View Devices to open the View Devices page.
- Click Reset Device to return to the Reset Device page.
- Select any option on the Broadcast Queue Manager menu bar.

The selected printer(s) are reset and the selected Web page opens.

Step 3.4. Redirect a Device

1. Click Devices on the Broadcast Queue Manager menu bar.

2. Select Redirect Device.

A Redirect Device page opens.

3. Make selections as follows.

Field	Description
Source Device	Printing from the device selected from the list of devices that are in device groups will be redirected to another printer.
Target Device	The device selected from the list of devices that are in device groups will print forms that were initially sent to the source device.

4. Click Redirect Device.

A message asks you to confirm that you want to redirect the printing.

5. Click OK.

Broadcast Queue Manager displays the change you made.

6. Do one of the following.

- Click View Device Configuration to open the Configure Devices page.
- Click Redirect Devices to re-open the Redirect Devices page.
- Select any of the options on the Broadcast Queue Manager menu bar.

The selected printer(s) are redirected and the selected Web page opens.

Step 3.5. Work with Grouped Devices through the Broadcast Queue Manager

Step 3.5. Work with Grouped Devices through the Broadcast Queue Manager

Device groups offer a convenient way to make one target selection for an [ad hoc broadcast \(page 1111\)](#) and have the message printed in several different locations based on the printers selected for the group.

1. Click Devices on the Broadcast Queue Manager menu bar.
2. Select Group Devices.

A Group Devices page opens.

3. Use any of the following options.

Option 3.5.1 (page 1093)	Add a Group to the Broadcast Queue Monitor
Option 3.5.2 (page 1094)	Rename a Group in the Broadcast Queue Monitor
Option 3.5.3 (page 1094)	Delete a Group from the Broadcast Queue Monitor
Option 3.5.4 (page 1094)	Add a Device to a Group
Option 3.5.5 (page 1095)	Remove a Device from the Group

Option 3.5.1 Add a Group to the Broadcast Queue Manager

1. Click Add on the Device Group toolbar.

A Device Group field displays.

2. Enter a name for the group in the **Device Group** field.
3. Repeat the procedure to add as many groups as you want.

Note: You can add groups at any time.

Broadcast creates the groups you add. The fields become read only.

Option 3.5.2. Rename a Group in the Broadcast Queue Manager

1. Select the group you want to rename in the **Device Groups** box.
2. Click Rename on the Device Group toolbar.

The group's **Device groups** field is enabled.

3. Enter the new name in the field.

Broadcast changes the group name.

Option 3.5.3. Delete a Group from the Broadcast Queue Manager

1. Select the group you want to delete in the **Device Groups** box.
2. Click Delete on the Device Group toolbar.

A message asks you to confirm that you want to delete the group.

3. Click OK.

Broadcast deletes the group.

Option 3.5.4. Add a Device to a Group

1. Select the group in which you want to add the device.
2. Click Add Device on the Device Group toolbar.

A Device field is enabled in which you can select an available device.

3. Select a device from the drop-down list.

The device is added to the selected group.

4. Continue to add as many devices as you need to selected groups.

The devices for each selected group display under the group they are added to.

5. Use the + and - buttons to the left of a group to display or hide its list of devices.

+	Click to expand the tree.
-	Click to collapse the tree.

Note: You can add devices at any time in the Broadcast Queue Manager.

Option 3.5.5. remove a Device from the Group

1. Expand the group in which you want to remove the device.
2. Select the device you want to remove.
3. Click Remove Device on the Device Group toolbar.

A message asks you to confirm that you want the device removed.

4. Click OK.

Broadcast removes the printer from the selected group.

Step 3.6. Pause/Resume Devices

A device can be paused so it will temporarily stop printing and resumed when it is again ready to print. For example, a printer was paused because it ran out of paper, then resumed when the paper was reloaded.

! **Important:** A device needs to be [reset \(page 1091\)](#) after it has been paused in order to alert Broadcast Queue Manager that it is available. Once reset, Broadcast Queue Manager will resume sending it forms to print.

1. Click Devices>Resume/Pause Devices on the Broadcast Queue Manager menu bar.

A list of printers display each printer's paused or resumed status.

2. Select one or more printers that have the same status, PAUSED or RESUMED.
3. Click the button that will change the printer status.

Pause	Pauses printers whose status is currently RESUMED.
Resume	Resumes printers whose status is currently PAUSE.

A message displays asking you to confirm the status change, Pause or Resume.

4. Click OK.

The selected printers are either paused or resumed. A Web page displays the printer(s) status changes.

Step 4. Work with Jobs through the Broadcast Queue Manager

Step 4. Work with Jobs through the Broadcast Queue Manager UI

Step 4.1 (page 1096)	View Search Jobs in the Broadcast Queue Manager.
Step 4.2 (page 1097)	Configure job priority.
Step 4.3 (page 1107)	Archive or delete jobs from the History Queue.

Step 4.1. Configure Job Priority

1. Click Jobs on the Broadcast Queue Manager menu bar.
2. Select Configure Priority.

An Edit Job Priority page opens.

3. Edit the priority in any of the following Priority fields.

Priorities can range from 1 - 99; 99 is the highest.

rect 80, 81, 249, 108 [\(page 1096\)](#)

rect 77, 113, 246, 140 [\(page 1096\)](#)

rect 75, 144, 244, 171 [\(page 1097\)](#)

rect 80, 177, 249, 204 [\(page 1097\)](#)

rect 77, 208, 246, 235 [\(page 1097\)](#)

Job	Description
Ad Hoc Broadcast	Jobs that are sent out of sequence of the normal broadcast.
Redirect	Jobs that are redirected from a source to target printer.

Resend	Jobs will be resent to the printer. Note: Resend jobs are a duplicate of the original broadcast to the selected target device.
Normal Broadcast	Original Broadcast jobs that are sent in sequence directly to a selected device.
Test Broadcast	Jobs that are testing the Broadcast configuration.

4. Click Configure Job Priority.

Broadcast asks you to confirm that you want the priorities to change.

5. Click OK.

Broadcast changes the priorities and displays the current configuration.

Note: You can change the priorities again the next time you open the Edit Job Priority Page.

Step 4.2. View/Search Jobs in the Broadcast Queue Manager UI

Step 4.2. View/Search Jobs in the Broadcast Queue Manager UI

1. Click Jobs on the Broadcast Queue Monitor menu bar.

2. Select View/Search Jobs.

An enter search Criteria page opens.

3. Select search criteria for a selected job type.

Job types are as follows.

Option 4.2.1 (page 1097)	Search for active jobs.
Option 4.2.2 (page 1100)	Search for history jobs.
Option 4.2.3 (page 1104)	Search for active and history jobs.
Option 4.2.4 (page 1106)	Search for archive jobs.

Option 4.2.1. Search for Active Jobs

Active jobs are in the selected device's active queue, in the process of being printed.

1 <i>(page 1098)</i>	Select search criteria for active jobs.
2 <i>(page 1099)</i>	Use active jobs search results.

Select search criteria for active jobs.

Options are as follows.

rect 29, 110, 259, 137 [\(page 1098\)](#)

rect 30, 136, 215, 163 [\(page 1098\)](#)

rect 31, 160, 222, 187 [\(page 1098\)](#)

rect 223, 161, 370, 188 [\(page 1098\)](#)

Option	Description	
Device Name	List of devices available to the IIS server.	
Job Type	Selections are:	
	Active	Broadcast jobs that are in the selected device's active queue.
		Note: Select Active to filter active jobs only.
	History	Broadcast jobs that have been processed and are in the History queue.
	Active and History	All the jobs in the Active and History queues that have been sent to the selected device.
	Archived	All of the jobs that have been archived for the selected device.
Field	All the Tracker attributes configured in the genfld.cfg file as generic fields for the Broadcast are listed in the drop down list. Note: If you do not make a selection all of the jobs that are in the selected Device Status will be listed on the BQM report page.	
Value	(Required if you selected a field) value of the selected field as criterion for searching the jobs.	
Device Status	(Available when Active is selected as the job type) Options for the status you are looking for are as follows.	
	Status	Report
	ALL	All active jobs.
	RESUMED	Forms that have been printed after a printer was paused the
	PAUSED	All jobs that have been paused
	ERROR	
	DELETING	

	SPOOLING	
	PRINTING	
	OFFLINE	
	PAPEROUT	
	PRINTED	
	DELETED	
	BLOCKED	
	USER_INTERVENTION	
	RESTARTING	
	UNKNOWN	

Click Search.

Broadcast Queue Manager displays the results.

Use the active jobs search results.

Features are as follows.

rect 126, 75, 487, 108 ([page 1099](#))

rect 134, 116, 469, 149 ([page 1099](#))

rect 214, 153, 380, 186 ([page 1099](#))

rect 4, 231, 756, 264 ([page 1099](#))

A	Review of search selections.
B	Forms that fulfill the selection requirements are listed.
	Listed data includes the: <ul style="list-style-type: none"> • Form type. • Date and time the form came into the queue. • Printer status. • Form fields and values.
C	Number of jobs found that match your criteria.
D	Buttons let you control the status of selected forms as follows.
Cancel	Click to stop the selected jobs from being printed. The jobs are permanently removed from the active list; they do not display in a history list.
Pause	Click to pause printing of selected jobs.
Resume	Click to resume printing of selected jobs.

Note: Use the [Ctrl or Shift \(page 1088\)](#) key to select multiple jobs.

[Up \(page 1099\)](#)

Option 4.2.2. Search for History Jobs

Option 4.2.2. Search for History Jobs

History jobs are Broadcast jobs that have been processed and are in the History queue.

1 (page 1100)	Select search criteria for history jobs.
2 (page 1100)	Use the history jobs search results.

Select search criteria for history jobs.

Search criteria options are as follows.

Option	Description
Device Name	List of devices available to the IIS server.
Job Type	Selections are:
	Active Broadcast jobs that are in the selected device's active queue.
	History Broadcast jobs that have been processed and are in the History queue.
	Note: Select History to filter history jobs only.
	Active and History All the jobs in the Active and History queues that have been sent to the selected device.
	Archived All of the jobs that have been archived for the selected device.
Field	All the Tracker attributes configured in the genfld.cfg file as generic fields for the Broadcast are listed in the drop down list. Note: If you do not make a selection all of the jobs that are in the selected Device Status will be listed on the BQM report page.
Value	(Required if you selected a field) value of the selected field as criterion for searching the jobs.
Device Status	Disabled when History is selected.

Click **Search**.

Broadcast Queue Manager displays the results.

Use the history jobs search results.

Features are as follows

rect 151, 86, 524, 113 ([page 1101](#))

rect 216, 124, 471, 151 ([page 1101](#))

rect 254, 162, 426, 189 ([page 1101](#))


rect 10, 229, 742, 407 ([page 1101](#))

A	Review of search selections.	
B	Forms that fulfill the selection requirements are listed from oldest (top of list) to newest.	
	<p>Listed data includes the:</p> <ul style="list-style-type: none"> • Form type. • Date and time the form came into the queue. • Printer status. <p>Note: The printer status is always PRINTED for history jobs.</p> <ul style="list-style-type: none"> • Form fields and values. 	
C	Number of jobs found that match your criteria.	
D	Buttons let you control the status of selected forms as follows.	
	Option 4.2.2.1 (page 1101)	Resend Resends selected jobs.
	Option 4.2.2.2 (page 1102)	Redirect Redirects selected jobs.

Note: Use the [Ctrl or Shift \(page 1088\)](#) key to select multiple jobs.

[Up \(page 1100\)](#)

Option 4.2.2.1. Resend Selected History Jobs

 **Note:** This functionality is available when either History or Active and History is selected as the **Job Type**.

1. Select the jobs you want to resend.
2. Click Resend.

A page opens that enables you to review the specifications and enter the number of copies of each form to resend.

Fields to review and fill in are:

rect 91, 112, 216, 126 ([page 1102](#))

rect 95, 128, 220, 437 ([page 1102](#))

rect 93, 437, 218, 451 ([page 1102](#))

Field	Description	
Current Device	Read-only	The device that previously sent the jobs, will resend them
Jobs Selected	Read-only	List of jobs that are selected to be resent.
No. of copies	Read/Write	Number of copies that will be printed for each form.

[Up \(page 1101\)](#)

3. Click Resend Jobs.

A message asks you to confirm that you want to resend the jobs.


4. Click OK.

Broadcast Queue Manager displays details about the jobs that were resent.

5. Do one of the following.

- Click View Jobs in <Printer Name> to continue working with the list of history jobs or select another menu option.
- Select any option on the Broadcast Queue Manager menu bar.


The jobs have been resent and the page you select opens.

 **Note:** The number of copies printed for a form that display on the Print Options tab in the Form Properties dialog box when you open the form's history file reflects the number of copies originally specified in the either of four BCO function blocks.

Broadcast Form
Send Form to Device
Send WYSIWYG Form
Send WYSIWYG Form By Variable

The number entered in the Broadcast Queue Manager is not recorded.

Option 4.2.2.2. Redirect Selected History Jobs

 **Note:** This functionality is available when either History or Active and History is selected as the **Job Type**.

1. Select the jobs you want to redirect.
2. Click Redirect.

A page opens that enables you to review what jobs were selected and make selections as follows.

Fields are:

rect 75, 104, 233, 117 ([page 1103](#))

rect 73, 119, 279, 418 ([page 1103](#))

rect 69, 419, 307, 432 ([page 1103](#))

rect 69, 439, 307, 452 ([page 1103](#))

rect 67, 458, 305, 471 ([page 1103](#))

Field	Description	
Current Device	Read-only	The device that previously sent the jobs, will resend them
Jobs Selected	Read-only	List of jobs that are selected to be resent.
Select Device to Redirect to	Read/Write	A single device that will print the jobs.
Select Group to Redirect to	Read/Write	The Broadcast device group that will print the jobs. The primary or secondary device will print the jobs depending on which is the active printer when the jobs are sent. A device and/or a group can be selected.
No. Of Copies	Read/Write	Number of copies of each form that should be printed.

[Up \(page 1103\)](#)

3. Click Redirect Jobs.

A message asks you to confirm that the jobs should be redirected.

4. Click OK.


The jobs are redirected to the selected devices/device groups.

Broadcast Queue Manager reports when the jobs are redirected.

5. Do one of the following.

- Click View Jobs in <Printer Name> to continue working with the list of history jobs or select another menu option.
- Select any option on the Broadcast Queue Manager menu bar.

The jobs have been redirected and the page you select opens.

 **Note:** : The number of copies printed for a form that display on the Print Options tab in the Form Properties dialog box when you open the form's history file reflects the number of copies originally specified in the either of four BCO function blocks.

Broadcast Form
Send Form to Device
Send WYSIWYG Form
Send WYSIWYG Form By Variable

The number entered in the Broadcast Queue Manager is not recorded.

Option 4.2.3. Search for Active and History Jobs

History and Active jobs displays all of the jobs in both the active and history queues that fulfill the selected criteria.

1 (page 1104)	Select search criteria for active and history jobs.
2 (page 1105)	Use the active and history jobs search results.

Select search criteria for active and history jobs.

Search criteria options are as follows.

Option	Description
Device Name	List of devices available to the IIS server.
Job Type	Selections are:
	Active Broadcast jobs that are in the selected device's active queue.
	History Broadcast jobs that have been processed and are in the History queue.
	Note: Select Active and History to filter both active and history jobs.
	Active and History All the jobs in the Active and History queues that have been sent to the selected device.
	Archived All of the jobs that have been archived for the selected device.
Field	All the Tracker attributes configured in the genfld.cfg file as generic fields for the Broadcast are listed in the drop down list. Note: If you do not make a selection all of the jobs that are in the selected Device Status will be listed on the BQM report page.
Value	(Required if you selected a field) value of the selected field as criterion for searching the jobs.

Device Status	ALL applies to both History and Active. Any other option will find only Active jobs.
---------------	--

Click **Search**.

Broadcast Queue Manager displays the results.

Use the active and history jobs search results.

Features are as follows.

rect 149, 66, 395, 96 ([page 1105](#))

rect 18, 104, 534, 134 ([page 1105](#))

rect 188, 141, 347, 171 ([page 1105](#))

rect 6, 172, 568, 330 ([page 1105](#))

A	Review of search selections.	
B	Forms that fulfill the selection requirements are listed from oldest (top of list) to newest.	
	Listed data includes the: <ul style="list-style-type: none"> • Form type. • Date and time the form came into the queue. • Printer status. <p>Note: The printer status is always PRINTED for history jobs.</p> <ul style="list-style-type: none"> • Form fields and values. 	
C	Number of jobs found that match your criteria.	
D	Buttons let you control the status of selected forms as follows.	
	For Active Jobs	
	Cancel	Click to stop the selected jobs from being printed. The jobs are permanently removed from the active list; they do not display in a history list.
	Pause	Click to pause printing of selected jobs.
	Resume	Click to resume printing of selected jobs.
	For History Jobs	
	Option 4.3.2.1 (page 1101)	Resend Click to resend selected Active jobs.
	Option 4.3.2.2 (page 1102)	Redirect Click to redirect selected History jobs.

Note: Use the [Ctrl or Shift \(page 1088\)](#) key to select multiple jobs.

Option 4.2.4. Search for Archived Jobs

Active jobs are in the selected device's active queue.

1 <i>(page 1106)</i>	Select search criteria for archived jobs.
2 <i>(page 1106)</i>	Use the archived jobs search results.

1. Select search criteria for archived jobs.

Search criteria options are as follows.

Option	Description
Device Name	List of devices available to the IIS server.
Job Type	Selections are:
	Active Broadcast jobs that are in the selected device's active queue.
	History Broadcast jobs that have been processed and are in the History queue.
	Active and History All the jobs in the Active and History queues that have been sent to the selected device.
	Archived All of the jobs that have been archived for the selected device.
	Note: Select Archived to filter archived jobs only.
Field	All the Tracker attributes configured in the genfld.cfg file as generic fields for the Broadcast are listed in the drop down list. Note: If you do not make a selection all of the jobs that are in the selected Device Status will be listed on the BQM report page.
Value	(Required if you selected a field) value of the selected field as criterion for searching the jobs.
Device Status	Disabled when Archived is selected.

Click Search.

Broadcast Queue Manager displays the results.

1. Use the archived jobs search results.

Features are as follows.

A	Review of search selections.
B	Forms that fulfill the selection requirements are listed.

	Listed data includes the: <ul style="list-style-type: none"> • Form type. • Date and time the form came into the queue. • Printer status. • Form fields and values.
C	Number of jobs found that match your criteria.
D	Buttons let you control the status of selected forms as follows.
	Requeue Job(s) (page 1107)
	Transfer selected jobs from the Archive folder back into the History folder.

Note: Single or multiple forms can be selected.

Hold down the

- Shift key to select forms listed consecutively.
- Ctrl key to select forms that are not listed consecutively.

Note: Use the [Ctrl or Shift \(page 1088\)](#) key to select multiple jobs.

Requeue Job(s)

1. Click Requeue Job(s).

A message asks you to confirm re-queuing the selected jobs.

2. Click OK.

Broadcast Queue Manager transfers the jobs to the History queue and displays a list of job files that were transferred from the Archive to the History folder.

3. Do one of the following.
 - Click View Archived Jobs in <Device> to return to the Archive results page.
 - Click View Active & History Jobs in <Device> to view active and history jobs that match the search criteria.
 - Select any other option on the Broadcast Queue Manager menu.

[Up \(page 1107\)](#)

Step 4.3. Archive or Delete Jobs from the History Queue

Step 4.3. Archive or Delete Jobs from the History Queue

Broadcast Queue Manager displays an alarm in the Alarm Viewer when the Active or History queue reaches or exceeds its [limits \(page 1089\)](#) .

1. Click Jobs on the Broadcast Queue Manager menu bar.
2. Select Archive Jobs.

An Archive / Delete Jobs page opens.

3. Select the one or more devices that have the jobs to be archived or deleted.

Information for your selection includes the following.

rect 190, 179, 330, 200 [\(page 1108\)](#)

rect 55, 216, 272, 310 [\(page 1108\)](#)

rect 276, 214, 454, 306 [\(page 1108\)](#)

rect 115, 143, 255, 164 [Option 4.3.1. Archive Jobs \(page 1108\)](#)

rect 260, 142, 395, 163 [Option 4.3.2. Delete Jobs \(page 1110\)](#)

Feature	Description
Number of devices	Number of devices that are included in the Broadcast device groups.
Device Name	Identification of devices that are included in the Broadcast device groups. Note: Multiple devices can be selected in the usual manner: using the Shift or Ctrl (page 1088) key.
Default No. Of Jobs	The number of maximum jobs specified for the History Queue through the Configure Device Groups (page 1088) page.

4. Do one of the following.

Option 4.3.1 (page 1108)	Archive jobs.
Option 4.3.2 (page 1110)	Delete jobs.

Option 4.3.1. Archive Jobs

1. Click Archive Jobs.

A message asks you to confirm that you want to archive jobs.

2. Click OK.

Broadcast Queue Manager offers two options.

	Archive jobs using preset values.
	Archive a specified number of jobs.

Archive jobs using preset values.

- a. Click OK to confirm archiving jobs with preset values.

Broadcast Queue Manager Reports the status as follows.

rect 30, 188, 114, 238 [\(page 1109\)](#)

rect 114, 187, 214, 237 [\(page 1109\)](#)

rect 214, 186, 312, 236 [\(page 1109\)](#)

rect 310, 190, 363, 240 [\(page 1109\)](#)

Field	Description
Device	Device whose jobs were archived.
Archive Status	Whether the jobs were archived or deleted.
Number of Jobs	Number of jobs entered for the High History Queue Limit (page 1088) on the Configure Devices page.
Result	Number of jobs that were archived. Note: This number will depend on how many jobs are in the History folder static to the number of jobs the preset values specify.

[Up \(page 1108\)](#)

Archive a specified number of jobs.

- a. Click Cancel to archive jobs based on a currently entered number.

An Explorer User Prompt dialog box opens.

- a. Enter the number of jobs you want to archive.
- a. Click OK.

Broadcast Queue Manager reports the archive status as follows.

rect 30, 188, 114, 238 [\(page 1109\)](#)

rect 114, 187, 210, 237 [\(page 1109\)](#)

rect 210, 189, 309, 239 [\(page 1109\)](#)

rect 306, 188, 360, 238 [\(page 1110\)](#)

Field	Description
Device	Device whose jobs were archived.
Archive Status	Whether the jobs were archived or deleted.
Number of Jobs	Number of jobs you entered to be archived.

Result	Number of jobs that were archived. Note: This number will depend on how many jobs are in the History folder static to the number of jobs you want to archive.
--------	--

3. Click BACK to return to the Archive Jobs page or select any of the menu options to continue working in the Broadcast Queue Manager.

[Up \(page 1108\)](#)

Option 4.3.2. Delete Jobs

1. Click Delete Jobs (A).

A message asks you to confirm that you want to delete jobs.

2. Click OK.

Broadcast Queue Manager offers two options.

- Delete jobs using preset values.
- Delete a specified number of jobs.

Delete jobs using preset values.

- a. Click OK.

Broadcast Queue Manager Reports the delete status as follows.

rect 30, 188, 114, 238 [\(page 1110\)](#)

rect 112, 188, 211, 238 [\(page 1110\)](#)

rect 211, 189, 313, 239 [\(page 1110\)](#)

rect 312, 192, 361, 242 [\(page 1110\)](#)

Field	Description
Device	Device whose jobs were archived.
Archive Status	Whether the jobs were archived or deleted.
Number of Jobs	Number of jobs entered for the High History Queue Limit (page 1088) on the Configure Devices page.
Result	Number of jobs that were archived. Note: This number will depend on how many jobs are in the History folder static to the number of jobs the preset values specify.

[Up \(page 1110\)](#)

Delete a specified number of jobs.

- a. Click Cancel.

An Explorer User Prompt dialog box opens.

- a. Enter the number of jobs you want to delete.
- a. Click OK.

Broadcast Queue Manager reports the status as follows.

rect 30, 188, 114, 238 ([page 1111](#))

rect 112, 188, 209, 238 ([page 1111](#))

rect 206, 187, 312, 237 ([page 1111](#))

rect 312, 188, 360, 238 ([page 1111](#))

Field	Description
Device	Device whose jobs were deleted.
Archive Status	Whether the jobs were archived or deleted.
Number of Jobs	Number of jobs you entered to be deleted.
Result	Number of jobs that were deleted. Note: This number will depend on how many jobs are in the History folder static to the number of jobs you want to archive.

3. Click BACK to return to the Archive Jobs page or select any of the menu options to continue working in the Broadcast Queue Manager.

[Up \(page 1110\)](#)

Step 5. Send an Ad hoc Broadcast through the Broadcast Queue Manager

1. Click Ad hoc Broadcast on the Broadcast Queue Manager UI menu bar.

An Ad hoc Broadcast page displays.

2. Create an ad hoc broadcast.

Fields for configuring the ad hoc broadcast are as follows.

rect 30, 139, 467, 231 ([page 1111](#))

rect 27, 265, 464, 426 ([page 1112](#))

rect 27, 230, 119, 268 ([page 1112](#))

Field	Description		
Device Options	Select one or both of the following.		
	<table border="1"> <tr> <td>Devices</td> <td>All devices configured for the IIS server. A selected device will print the ad hoc broadcast.</td> </tr> </table>	Devices	All devices configured for the IIS server. A selected device will print the ad hoc broadcast.
Devices	All devices configured for the IIS server. A selected device will print the ad hoc broadcast.		

	Device Groups	Groups (page 1093) configured on the Device groups page. All of the printers in a selected group will print the Broadcast message.
Priority		The priority (page 1096) given to an ad hoc broadcast static to other broadcasts. The priority can be changed on this page. Broadcast Queue Manager will update it in the system.
Data		Message that you enter to be broadcast.

[Up \(page 1111\)](#)

3. Click Send Data when you are ready to broadcast.

Broadcast sends the message. Exactly when the message will be printed depends on the ad hoc broadcast priority and the length of the queue(s) for the selected printer(s).

Broadcast Queue Manager Job File Locations

1. Right-click CimForms in the TrackerCfg_UI.
2. Select Properties on the Popup menu.

A CimForms Properties dialog box opens.

3. Change any default paths, if needed.

Paths are as follows.

rect 9, 23, 382, 61 ([page 1112](#))

rect 6, 81, 379, 106 ([page 1113](#))

rect 6, 58, 379, 83 ([page 1112](#))

Field	Description	
Text Editor Path	Location of the text editor that you want to use to create and edit Control Character Token files and ASCII forms.	
		Opens a Select the Text Editor browser.
	Default	...Program Files\Proficy\Proficy CIMPLICITY\exe\CimFormsEditor.exe.

Spool Location	Location for active and history job files. Broadcast organizes folders and jobs as follows. <ol style="list-style-type: none"> A project folder is created in the specified location. Active and History folders are created in the project folder. Printer folders are created in both the Active and History folder. Active and History job files are moved to the appropriate folder. 	
		Opens a Browse for Folder browser.
	Default	C:\Temp\IBC1

Location for archived files. Broadcast organizes the folders and jobs as follows.	
a. A project folder is created in the specified location.	
b. An Archive folder is created in the project folder.	
c. Printer folders are created in the Archive folder.	
d. Archived job files are moved to the appropriate folder.	
	Opens a Browse for Folder browser.
Default	C:\Temp\Archive

Broadcast Alarms

Broadcast includes several alarms that display in the Alarm Viewer and alert you to problems when they occur.

The alarms are listed in the Workbench right pane under **Advanced>Alarms** .

Alarm ID	Is triggered when a/the:
ACTIVEQ_ALARM	Active Queue has reached or exceeded its high limit (page 1089) .
ACTIVEQ_WARNING	Active Queue has reached or exceeded its low limit (page 1089) .
BROADCAST_STATUS	Broadcast is down.
DEVICE_STATUS	Device is down.
HISTORYQ_ALARM	History Queue has reached or exceeded its high limit (page 1089) .
HISTORYQ_WARNING	History Queue has reached or exceeded its low limit (page 1089) .
JOB_TIMEOUT	Job has exceeded its time out (page 1089) and is not printed.

Alarms are all in the BCO alarm class.

Broadcast Troubleshooting

Broadcast Troubleshooting

Broadcast login.asp
Debug files
Log/alarms files configuration
Log / alarms for queue limits
Errors for devices and jobs
Broadcast success messages

Broadcast alarms
Frequently asked questions

Broadcast Login.asp

Problem

You try to log in to the Broadcast Queue Manager and you get an error message that tells you:

Login.asp, line 207 line 207 in logon.asp is

```
Set obj= Server.CreateObject("Scripting.FileSystemObject")
```

Problem Source

The source of the problem is that `Sccrun.dll` has not been registered.

This is a Microsoft bug.

The `FileSystemObject` object model is contained in the Scripting type library, which is located in the `Sccrun.dll` file.

Therefore, you must have `Sccrun.dll` in the appropriate system directory on your Web server to use the `FileSystemObject` object model.

Solution

Register `sccrun.dll` to solve the problem.

Broadcast Debug Files

Broadcast Debug Files

- Debug modules.
- Turn debug information on and off.

Debug modules

Broadcast debug information helps to find the root cause of a problem when Broadcast fails.

Broadcast debug can be broken into three main modules:

BcoComp.
BQM and BQM_Alr.
WYSIWYG Editor.

Debug information:

- Is generated separately for each object.
- Can be turned on or off, even when the project is running.

Turn debug information on and off

A file named debugbco.txt is generated in the C:\TEMP directory.

If debugbco.txt is not available, create an empty file.

The process to turn debug information on and off is as follows.

1. Add the following statements to debugbco.txt.

DEBUG_BCO=<n>

DEBUG_BQM=<n>

DEBUG_BQMA=<n>

DEBUG_EDITOR=<n>

Where

<n> does the following:

<n> is	Result
1	Turns debug on. The statements are: DEBUG_BCO=1 DEBUG_BQM=1 DEBUG_BQMA=1 DEBUG_EDITOR=1
0	Turns debug off. The statements are: DEBUG_BCO=0 DEBUG_BQM=0 DEBUG_BQMA=0 DEBUG_EDITOR=0

Note: Debug information for Editor is available in CIMPLICITY version 6.1 SP3 and higher.

2. Save debugbco.txt.

Whether or not debug information is generated will depend on the statement values.

guide: Guidelines

- There should not be any blank spaces in the file.
- Sequence is not important.
- Debug is not case sensitive.

BcoCompBcoComp

- BcoComp debug file.
- BcoComp debug information.

When triggered, BcoComp:

1. Generates a job.
2. Places the job in a <SpoolLocation> that is specified in a spoollocation.txt file.
 - Spoollocation.txt is located in the ...[CIMPLICITY path]\RCO\ directory.
 - <SpoolLocation> is defined as <ProjectName>\ACTIVE\<<DeviceName>
3. Updates the DeviceGroup List when BcoComp successfully places the job in the folder.

BcoComp debug file

If `Debug_Bco=1 (page 1115)` is in debugbco.txt.

BcoComp:

- Generates a debug file named LogFile_<ProjectName><UniqueID>.txt.
- Places the file in in c:\TEMP.

BcoComp Debug information

Information in LogFile_<ProjectName><UniqueID>.txt includes:

Success conditions

When a form is successfully created, a success message is written into a file as follows:

Form	File name
ASCII	ASCII Broadcast Success
WYSIWYG	WYSIWYG Broadcast Success

BcoComp Sample

```
In Constructor
Object Instance:1.
Install path:C:\Program Files\Proficy\Proficy CIMPLICITY\
In CreateMemoryMapFile:
File Path:D:\CimPE\examples\HMI\Yane\rco\cma.sta
The Return Code for CMA file: 183
```

```

Out CreateMemoryMapFile:
In CreateMemoryMapFile:
File Path:D:\CimPE\examples\HMI\Yane\rco\DeviceList.sta
The Return Code for CMA file: 183
Out CreateMemoryMapFile:
In CreateMemoryMapFile:
File Path:D:\CimPE\examples\HMI\Yane\rco\seqnum.sta
The Return Code for CMA file: 183
Out CreateMemoryMapFile:
D:\CimPE\examples\HMI\Yane\CIMForms\WYSIWYGForms\prod\Uticor.cwf
In SetRCOScriptName
Out SetRCOScriptName
In SetDevice
In ReadCMA
FaultDevices:GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC / TEXT
ONLY:GENERIC / TEXT ONLY (COPY
2):0:0:3:1:$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out ReadCMADData
In WYSIWYG_Send_Device
After CoCreate
Load Form success
In GenFld
c:\Program Files\Proficy\Proficy CIMPLICITY\
In QUERY_EXT_TRACKER_ATTRIBUTE
In ReadCMA
FaultDevices:GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC / TEXT
ONLY:GENERIC / TEXT ONLY (COPY
2):0:0:3:1:$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out ReadCMADData
In ReadCMA
FaultDevices:GRP1:0:1;
Out ReadCMADData
In UpdateSeqNumNStat
SEQNUM Data in File:GRP1:0:1;
Modify - Old Token
Out UpdateSeqNumNStat
JobName:0001087203094343;W;---;---;---;---;---;---;.cwf
In GetPathToStore
c:\Program Files\Proficy\Proficy CIMPLICITY\rco\spoollocation.txt
Merged Data:
Saved Data
In UpdateDevList
Out UpdateDevList
Error in WYSIWYG Fetch N Merge.
Out WYSIWYG_PostToDevice
WYSIWYG Broadcast Success.
In SetRCOScriptName
Out SetRCOScriptName
In SetDevice
In ReadCMA
FaultDevices:GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC / TEXT
ONLY:GENERIC / TEXT ONLY (COPY
2):0:0:3:1:$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;

```

```
Out ReadCMADData
In FetchNMerge
In GenFld
d:\CIMPLICITY\HMI\
In QUERY_EXT_TRACKER_ATTRIBUTE
In Parser
ASCII file opened
D:\CimPE\examples\HMI\Yane\CimForms\ASCIIForms\Prod\All in One-
SOLVE_PRT_EXPRT.caf
content:ASCIIICH 13
content:TEXT "Tracker Attributes"
content:ASCIIICH 13
content:TEXT "~~~~~"
content:TEXT "ATT1:"
content:PRTATTR ATT1
content:ASCIIICH 13
content:TEXT "ATT2:"
content:PRTATTR ATT2
content:ASCIIICH 13
content:TEXT "ATT3:"
content:PRTATTR ATT3
content:ASCIIICH 13
content:TEXT "ATT4:"
content:PRTATTR ATT4
content:ASCIIICH 13
content:TEXT "ATT5:"
content:PRTATTR ATT5
content:ASCIIICH 13
content:TEXT "ATT6:"
content:PRTATTR ATT6
content:ASCIIICH 13
content:ASCIIICH 13
content:ASCIIICH 13
content:TEXT "Tracker EXATtributes"
content:ASCIIICH 13
content:TEXT "~~~~~"
content:ASCIIICH 13
content:TEXT "EXATT1:"
content:PRTEXATTR EXATT1
content:ASCIIICH 13
content:TEXT "EXATT2:"
content:PRTEXATTR EXATT2
content:ASCIIICH 13
content:TEXT "EXATT3:"
content:PRTEXATTR EXATT3
content:ASCIIICH 13
content:TEXT "EXATT4:"
content:PRTEXATTR EXATT4
content:ASCIIICH 13
content:TEXT "EXATT5:"
content:PRTEXATTR EXATT5
content:ASCIIICH 13
content:TEXT "EXATT6:"
```



```

content:PRTEXATTR EXATT6
content:ASCIICH 13
content:ASCIICH 13
content:ASCIICH 13
content:TEXT "SOLVES Attributes"
content:ASCIICH 13
content:TEXT "~~~~~"
content:TEXT "Bit_VAR.Bit : "
content:SOLVE $SOLVE122$
In QUERY SOLVE
In QUERY SOLVE
content:ASCIICH 13
content:TEXT "AUTO_GEAR.GEAR : "
content:SOLVE $COM1$
In QUERY SOLVE
In QUERY SOLVE
content:ASCIICH 13
content:TEXT "BodyIDs.CheckDigit : "
content:SOLVE $Check Digit$
In QUERY SOLVE
In QUERY SOLVE
content:ASCIICH 13
content:TEXT "GEARL_EVEL.GEAR COUNT : "
content:SOLVE $C8$
In QUERY SOLVE
In QUERY SOLVE
content:ASCIICH 13
content:TEXT "ENGINE.V-Cylinders : "
content:SOLVE $C7$
In QUERY SOLVE
In QUERY SOLVE
content:ASCIICH 13
content:TEXT "Crank.Crk1_2 : "
content:SOLVE $C6$
In QUERY SOLVE
In QUERY SOLVE
content:ASCIICH 13
content:TEXT "COLOR.COLOR : "
content:SOLVE COLOR.COLOR
In QUERY SOLVE
In QUERY SOLVE
content:ASCIICH 13
content:TEXT "END. "
content:
Merged Data: Tracker
  Attributes~~~~~ATT1:111ATT2:222ATT3:333ATT4:444ATT5:555ATT6:666
Tracker
  EXATTributes~~~~~EXATT1:EX111EXATT2:EX222EXATT3:EX333EXATT4:---
EXATT5:---E
XATT6:---SOLVES Attributes~~~~~Bit_VAR.Bit :---
AUTO_GEAR.GEAR :---BodyIDs.CheckDigit :---G
EARL_EVEL.GEAR COUNT :---ENGINE.V-Cylinders :---Crank.Crk1_2 :---
COLOR.COLOR :---END.

```

```
In PostToDevice
In GetPathToStore
d:\CIMPLICITY\HMI\rco\spoollocation.txt
Directory Path: C:\TEMP\IBC1\YANE\ACTIVE\FINEPRINT1
In UpdateDevList
Out UpdateDevList
Saved Data - Successfully
LastErrorCode:135.
Out PostToDevice
ASCII Broadcast Success.
In Destructor
```

BQM_Alr and BQM

BQM_Alr and BQM

- BQM_Alr
- BQM

BQM_Alr

BQM_Alr

BQM_Alr begins executing when the project starts.

BQM_ALR.exe:

- Is located in the CIMPLICITY process control.
- Picks the job from the directory and posts it to specific device.
- BQM_Alr debug file.
- BQM_Alr debug information.

BQM_Alr debug file

If [Debug Bqma=1 \(page 1115\)](#) is in debugbco.txt.

BQM_Alr:

- Generates a debug file named ErrorInBQM_ALR_<ProjectName>.err.
- Places the file in c:\TEMP.

BQM_Alr Debug information

Information in ErrorInBQM_ALR_<ProjectName>.err includes:

Success conditions

Data about successfully printing and moving the job from the Active to the History folder is located after the statements.

- In GetCMADData
- In GetDevList

Failure conditions

Data about failure conditions includes:

- Missing data is located after the statements:
- GetCMADData
- GetDevList
- Search for token "PATH:" and the path should be <SpoolLocation>\<ProjectName>\ACTIVE \<DeviceName>.

Example

Path: C:\Temp\IBC1\Yane\Active\FINEPRINT1

- Search for tokens "pPrinterInfo->Status: " & "pJob->Status: "

If any one token is not 0, there is an Error State on the device.

- Search for token "No of GenFlds: " & "No of TokensInFile: ".

If the count does not match, the uob will be moved to an Invalid directory.

BQMAlr Debug Sample

```
New Sweep Time - 15:57:50 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
```

```
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 10 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:57:52 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 10 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:57:53 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
```

```
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 10 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:57:54 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 10 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:57:55 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
```

```
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 10 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:57:56 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 10 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
```

```
New Sweep Time - 15:57:57 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 10 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:57:58 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
```

```
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 10 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:57:59 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
No of GenFlds:6
No of TokensInFile:6
Out GetJobToPost
Path: C:\Temp\IBC1\Yane\Active\FINEPRINT1
JobName: 0001087208879359;W;---;---;---;---;---;---;.cwf
Posted Job
In GenerateAlarms
nNumofJob: 2 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 10 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:00 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
```



```
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
No of GenFlds:6
No of TokensInFile:6
Out GetJobToPost
Path: C:\Temp\IBC1\Yane\Active\FINEPRINT1
JobName: 0001087208879640;A;---;---;---;---;---;---;.caf
Posted Job
In GenerateAlarms
nNumofJob: 1 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 11 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:01 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:02 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
```

```
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:03 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:04 Date - 14/06/04
```

```
In GetCMAData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMAData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:05 Date - 14/06/04
In GetCMAData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMAData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
```

```
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:06 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:07 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
```

```
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:08 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:09 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
```

```
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:10 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:11 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
```

```
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:12 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMADData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:13 Date - 14/06/04
In GetCMADData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
```

```
Out GetCMAData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
Out GenerateAlarms
New Sweep Time - 15:58:14 Date - 14/06/04
In GetCMAData
GRP1:FINEPRINT1:FINEPRINT2:0:0:3:2:$MAC_FR;GRP2:GENERIC /
TEXT ONLY:GENERIC / TEXT ONLY (COPY 2):0:0:3:1:
$SYSTEM;GRP3:VINAY:PRINT1:0:0:2:1:$SYSTEM;
Out GetCMAData
In GetDevList
Data of DevList:CIMPRINTERBCI;FINEPRINT1;
AlarmConfig:Grp1:12:13:14:15:10;GRP2:80:100:80:100:10;GRP3:80:100:80:100:10;
Device extracted from MAP:CIMPRINTERBCI
Not Available in CMA
GENERIC / TEXT ONLY
VINAY
FINEPRINT1
Device extracted from MAP:FINEPRINT1
In AutoSendNRedirect -- FINEPRINT1
DeviceToPost: FINEPRINT2
pPrinterInfo->Status: 0.
ErrorID on Printer: 0.
No Error on Printer.
PrinterSpooler is Empty. Searching for Job to send...
C:\Temp\IBC1\Yane\Active\FINEPRINT1
In GetJobToPost
Out GetJobToPost
No Jobs to Post
In GenerateAlarms
nNumofJob: 0 PresentAlr.nAct_WAlarm: 0 PresentAlr.nAct_Alarm: 0
nNumofJob: 12 PresentAlr.nHis_WAlarm: 0 PresentAlr.nHis_Alarm: 0
```



```
Out GenerateAlarms
```

BQM

BQM

BQM:

- Helps BQM_Alr post the job to the device.
- Serves a request made from the Broadcast Web pages.
- BQM debug file.
- BQM debug information.

BQM debug file

If [Debug Bqm=1 \(page 1115\)](#) is in debugbco.txt.

BQM:

- Generates a debug file named Error_BQM_<ProjectName>_xxx.txt.
- Places the file in c:\TEMP.

BQM Debug information

Separate files are generated for every request made from the Web pages.

But of all these, only one file will be very big in size, which will help to drill down when the Jobs are stuck in the ACTIVE folder.

Failure condition

Data about failure conditions includes:

- Search for the token Exception:.

This helps to identify the failure of function.

BQM Sample

```
NewSweep :
NewSweep :
In ReadCMADData
Out ReadCMADData
In ReadCMADData
Out ReadCMADData
```

```
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMADData
Out ReadCMADData
In ReadCMADData
Out ReadCMADData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMADData
Out ReadCMADData
In ReadCMADData
Out ReadCMADData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMADData
Out ReadCMADData
In ReadCMADData
Out ReadCMADData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMADData
Out ReadCMADData
In ReadCMADData
Out ReadCMADData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
```

```
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In: GetFields
In: PostJob
In: GetPath
In SendWYSIWYGFormToPrinter
LoadForm success
ErrorMsg: The operation completed successfully.
PrintFormWithData success
In GetNumofJobsIn: GetFields
```

```
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In: GetPath
In: GetPath
In: GetFields
In: PostJob
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In: GetPath
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
```

```
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
```

```
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
Out ReadCMAData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMAData
Out ReadCMAData
In ReadCMAData
```

```

Out ReadCMADData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMADData
Out ReadCMADData
In ReadCMADData
Out ReadCMADData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs
NewSweep:
In ReadCMADData
Out ReadCMADData
In ReadCMADData
Out ReadCMADData
In: ReadAlarmConfig
In: GetPath
In GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobsIn GetNumofJobsIn: GetFields
In: GetPath
Out GetNumofJobs

```

Editor

Editor

The Editor is:


- Involved only when File>Test Print is used for a WYSIWYG form.
- Instantiated by both:
 - [BCOComp \(page 1116\)](#)

For Fetch&Merge and then saving the form in the Active folder.

- [BQM \(page 1120\)](#) components

Picks the job from Active folder and then posts the job to the specified device for each broadcast.

So, there will be two debug files getting generated for each broadcast as shown in “sample” below.

 **Note:** When a user selects to Resent or Redirect a job through the Broadcast Web pages, the Editor is instantiated only by the BQM. This insures that there will be only [one debug \(page 1142\)](#) file that is being generated.

Editor debug file

If `Debug_EDITOR=1` ([page 1115](#)) is in debugbco.txt.

Editor:

- Generates a debug file named LoginInWYS_Editor + <UniqueID><ProjectName>.err.
- Places the file in c:\TEMP.


Editor Debug information

Data in the debug file includes:

- Messages are logged with time stamp.
- List of all the messages in the Broadcast process flow
- Logs any errors or warnings logs appropriately

Example

“Printer Error Message:- Print Form with Height 635.00 and Width 213.31, (in mm) does not exist. This Job is force printed”.

 **Note:** Messages for `BroadcastFetchNMerge` will be logged when `BCOComp` is included in Broadcast. However, these messages are not logged if `BQM` participates in the Broadcast process.

Editor Sample 1

```
06/24/04 08:54:20 PM In EditorDoc : Constructor
06/24/04 08:54:20 PM In Editor : Constructor
06/24/04 08:54:20 PM *** In Editor::SetFormPath ***
06/24/04 08:54:20 PM In Editor::SetFormPath : C:\Temp\IBC1\BROADCASTPRJ
\Active\FINEPRINT\0001088090658562;W;71;.cwf
06/24/04 08:54:20 PM *** In Editor::LoadForm ***
06/24/04 08:54:20 PM *** In Editor::OpenMainStorage ***
06/24/04 08:54:20 PM In Editor::OpenMainStorage : Open Storage in Read Mode
06/24/04 08:54:20 PM In Editor::LoadForm : Open Storage Success
06/24/04 08:54:20 PM In Editor::LoadForm : Loading Document(LoadDoc)
Successful
06/24/04 08:54:20 PM In Editor::LoadForm : Loading Form Data.... i.e
Version, No of objects etc.
```



```

06/24/04 08:54:20 PM In Editor::LoadForm : Form Data:- No. of Objects 342 -
Version 3
06/24/04 08:54:20 PM In Editor::LoadForm : Objects Loading.....
06/24/04 08:54:20 PM *** In Editor::PrintFormWithData ***
06/24/04 08:54:20 PM In Editor::PrintFormWithData : Job Name:-
0001088090658562;W;71;.cwf
06/24/04 08:54:20 PM In Editor::PrintFormWithData : Device Name:- FINEPRINT
06/24/04 08:54:20 PM In Editor::PrintFormWithData : OpenPrinter Successful
06/24/04 08:54:20 PM *** In Editor::GetPrintFormNTSize ***
06/24/04 08:54:20 PM In Editor::PrintFormWithData : Printer Error Message:-
Print Form with Height 635.00 and Width 213.31, ( in mm ) does not
exists. This Job is force printed
06/24/04 08:54:20 PM In Editor::PrintFormWithData : Printing Form with
Width 213.31 and Height 635.00 ( in mm )
06/24/04 08:54:20 PM In Editor::PrintFormWithData : Created Device
Context(PrinterDC) Successfully
06/24/04 08:54:20 PM In Editor::PrintFormWithData : Printing the job.....
06/24/04 08:54:20 PM In Editor::PrintFormWithData : Printed the job
Successfully :- 0
06/24/04 08:54:20 PM In Editor : Destructor
06/24/04 08:54:20 PM In EditorDoc : Destructor

```

Editor Sample 2

```

06/24/04 08:54:18 PM In EditorDoc : Constructor
06/24/04 08:54:18 PM In Editor : Constructor
06/24/04 08:54:18 PM *** In Editor::SetFormPath ***
06/24/04 08:54:18 PM In Editor::SetFormPath : D:\Temp\BroadcastPrj\CIMForms
\WYSIWYGForms\prod\A1_T5_BLD CRD.cwf
06/24/04 08:54:18 PM *** In Editor::LoadForm ***
06/24/04 08:54:18 PM *** In Editor::OpenMainStorage ***
06/24/04 08:54:18 PM In Editor::OpenMainStorage : Open Storage in Read Mode
06/24/04 08:54:18 PM In Editor::LoadForm : Open Storage Success
06/24/04 08:54:18 PM In Editor::LoadForm : Loading Document(LoadDoc)
Successful
06/24/04 08:54:18 PM In Editor::LoadForm : Loading Form Data.... i.e
Version, No of objects etc.
06/24/04 08:54:18 PM In Editor::LoadForm : Form Data:- No. of Objects 342 -
Version 2
06/24/04 08:54:18 PM In Editor::LoadForm : Objects Loading.....
06/24/04 08:54:18 PM *** In Editor::SetRCOSiteInWYSIWYG ***
06/24/04 08:54:18 PM In Editor::SetRCOSiteInWYSIWYG : New_Site.WYS_BC
06/24/04 08:54:18 PM In Editor::BroadcastFetchNMerge : PROJECT Name:-
BROADCASTPRJ
06/24/04 08:54:18 PM In Editor::BroadcastFetchNMerge : ITEM ID:- A5
06/24/04 08:54:18 PM In Editor::BroadcastFetchNMerge : FORM Name:-
A1_T5_BLD CRD
06/24/04 08:54:18 PM In Editor::BroadcastFetchNMerge : Fetching and
Merging.....
06/24/04 08:54:20 PM In Editor::BroadcastFetchNMerge : Fetching and Merging
Completed
06/24/04 08:54:20 PM *** In Editor::SetFormPath ***

```

```

06/24/04 08:54:20 PM In Editor::SetFormPath : C:\TEMP\IBC1\BROADCASTPRJ
\ACTIVE\FINEPRINT\0001088090658562;W;71;.cwf
06/24/04 08:54:20 PM *** In Editor::SaveForm ***
06/24/04 08:54:20 PM *** In Editor::OpenMainStorage ***
06/24/04 08:54:20 PM In Editor::OpenMainStorage : Open Storage in Write
Mode
06/24/04 08:54:20 PM In Editor::SaveForm : Create Storage Success
06/24/04 08:54:20 PM In Editor::SaveForm : Saving Document(LoadDoc)
Successful
06/24/04 08:54:20 PM In Editor::SaveForm : Saving Form Data.... i.e
Version, No of objects etc.
06/24/04 08:54:20 PM In Editor::SaveForm : Form Data:- No. of Objects 342 -
Version 3
06/24/04 08:54:20 PM In Editor::SaveForm : Objects Saving.....
06/24/04 08:54:20 PM In Editor : Destructor
06/24/04 08:54:20 PM In EditorDoc : Destructor

```

Log/Alarms Files Configuration

- Log/alarms configuration files.
- Row format in the configuration files
- Device pause/offline.

Log/alarms configuration files

1. Configuration files for log/alarms are:

File	Configurationfor:
Printer_AlrNLogConfig.csv	Errors on the Printer.
Job_AlrNLogConfig.csv	Errors in individual jobs.

2. The two configuration files are located in the [CIMPPLICITY Install]\RCO directory.

Row format in the configuration files


Each row in the configuration file has the following data:

```
| ErrorCode | Log-0 / Alarm-1 / Both-2 | DFO-1 | Message | ErrorCodeInHex
```

Where elements are:

Element	Description
First	ErrorCode
Second	Log-0
	Alarm-1
	Both-2

	DFO-1	No alarming or logging
Third	Message	Actual message when the corresponding error occurs. The message: <ul style="list-style-type: none"> • Is seen in the log or alarm based on the user setting • Provides a description of the error at the printer or job Recommendation: Keep the default settings.
Fourth	ErrorCodeInHex	Hex code returned by the printer/job.
	1	Perform automatic device fail over.
	0	Do not perform automatic device fail over.

 **Note:** The printer_alrnlogConfig.csv and job_alrnlogConfig.csv files are available in the ...<CIMPLICITY install>\rco folder

Example

1|2|1|PRINTER_STATUS_PAUSED|0x00000001

Device Pause/Offline

The following happens when a user pauses the device or sets it to off line.

Pause/Offline	Result
Pause	Logging and/or alarms respond immediately.
Offline	The status is not updated immediately. It will take a noticeable period of time to respond; the behavior of different devices will differ for the same error condition. As a result, you need to do the configuration settings. Currently the: <ul style="list-style-type: none"> • Error Configuration setting applies to all the printers/devices. • Setting is common for all Broadcast projects on a system; it is not project specific.

Log/Alarms for Queue Limits

There are four main alarms for Queue limits.

BQM_Alr raises these alarms.

The alarms are:

Alarm	Raised when the job in the:	
ActiveQ_Warning	ACTIVE folder exceeds the configured LOW limit value.	
	MSG	XX device's active queue has exceeded the warning limit.
ActiveQ_Alarm	ACTIVE folder exceeds the configured HIGH limit value.	
	MSG	XX device's active queue has exceeded the limit.
HistoryQ_warning	HISTORY folder exceeds the configured LOW limit value.	
	MSG	XX device's history queue has exceeded the warning limit.

HistoryQ_Alarm	HISTORY folder exceeds the configured HIGH limit value.	
	MSG	XX device's history queue has exceeded the limit.

Errors for Devices and Jobs

Errors for which an alarm or log will be generated based on the [user configuration log/alarms \(page 1144\)](#) files.

- Errors on a device.
- Errors on a job.

Errors on a device

```

PRINTER_STATUS_PAUSED
PRINTER_STATUS_ERROR
PRINTER_STATUS_PENDING_DELETION
PRINTER_STATUS_PAPER_JAM
PRINTER_STATUS_PAPER_OUT
PRINTER_STATUS_MANUAL_FEED
PRINTER_STATUS_PAPER_PROBLEM
PRINTER_STATUS_OFFLINE
PRINTER_STATUS_IO_ACTIVE
PRINTER_STATUS_BUSY
PRINTER_STATUS_PRINTING
PRINTER_STATUS_OUTPUT_BIN_FULL
PRINTER_STATUS_NOT_AVAILABLE
PRINTER_STATUS_WAITING
PRINTER_STATUS_PROCESSING
PRINTER_STATUS_INITIALIZING
PRINTER_STATUS_WARMING_UP
PRINTER_STATUS_TONER_LOW
PRINTER_STATUS_NO_TONER
PRINTER_STATUS_PAGE_PUNT
PRINTER_STATUS_USER_INTERVENTION
PRINTER_STATUS_OUT_OF_MEMORY
PRINTER_STATUS_DOOR_OPEN
PRINTER_STATUS_SERVER_UNKNOWN
PRINTER_STATUS_POWER_SAVE

```

Errors on a job

```

JOB_STATUS_PAUSED
JOB_STATUS_ERROR
JOB_STATUS_DELETING
JOB_STATUS_SPOOLING
JOB_STATUS_PRINTING
JOB_STATUS_OFFLINE
JOB_STATUS_PAPEROUT
JOB_STATUS_PRINTED

```

```
JOB_STATUS_DELETED
JOB_STATUS_BLOCKED_DEVO
JOB_STATUS_USER_INTERVENTION
JOB_STATUS_RESTART
```

Broadcast Success Messages

Broadcast success messages are generated in:

- An log file named <rcosite name>.out.

<Rcosite name>.out:

- Carries the name of the site from which the log is generated.
- Is located in the project\log folder.
- Provides messags that the function call is a success; however, the data may not be correct.

Messages are:

- ASCII Broadcast successful.
- WYSIWYG Broadcast successful.
- Solve 'SOLVE_NAME' used in form 'FORM_NAME' returned multiple values.
- A log file named <project name>.log

<Project name>.log:

- Carries the name of the project from which the log is generated.
- Is located in the project\log folder.
- Provides several success messages.

Messages are:


- Solve 'SOLVE_NAME' used in form 'FORM_NAME' returned multiple values.
- Point is currently unavailable
- Problem in Saving object's TYPE Data
- Problem in Loading object's Data

Broadcast Alarms

Broadcast_Status

BcoComp generates a Broadcast_Status alarm, when it encounters an error while it is generating the job or placing it in the active folder of the device.

An alarm is raised with one of the following error messages.

 **Note:** The message depends on the situation.

Problem in Opening GenFld.

Problem in Opening the CAF File XX.

Device Not selected in Function Block by user.

Problem in Opening CMA.sta (Common Memory Area) File)

Device XX Not Available (Problem in getting Device handler).

Problem with the Spooler of device XX.

Problem in creating Page in Spooler of device XX.

Problem in posting the Data to device XX.

Failed to retrieve Device Information.

Problem in creating MemoryMapFile.

Failed to retrieve Device Information.

Problem in creating MapView for MemoryMapFile.

Unable to retrieve job information from the device.

Problem in getting Device handle.

Problem is setting the Job Priority.

Problem in creating Directory for spooling files in Device XX.

Problem in saving the data in spool Directory.

Problem in Creating AccessToken for CMA.

Fetch & Merge Failed

Device XX is down.

Problem with External Source (Solve API).

Problem with External Source (Tracker API).

Problem with External Source (Extended Tracker API).

Data Fetch Failed on WYSIWYG Broadcast.

The print job was aborted.

Problem in Opening the CWF File XX.

WYSIWYG Form has no Objects.

Problem loading the WYSIWYG Form.

Invalid WYSIWYG form, may be an ASCII form or any Text File rename to WYSIWYG.

Solve 'XX' used in form 'XX' returned multiple values.

The print job was aborted for an unspecified reason.

The print job was aborted because the user clicked the Cancel button in the dialog box that displays the status of the print job.

The print job was aborted because the user canceled it through the operating system shell.

The system is out of disk space, so no further printer data can be spooled.


The system is out of memory, so no further printer data can be spooled.

Data is not available for some of the Generic Fileds.

Device_Status

BcoComp generates a Device_Status alarm, when it encounters an error on the device or job that is sent to the device.

An alarm is raised with one of the following error messages.

 **Note:** The message depends on the situation.

Device XX is paused.

Device XX is in an error state.

Device XX is deleting a print job.

Paper Jam in the Device XX.

Device XX is out of paper.

Device XX is in a manual feed state.

Device XX has a paper problem.

Device XX is offline.

Device XX is in an active input/output state.

Device XX is busy.

Device XX is printing.

Device XX's output bin is full.

Device XX is not available for printing.

Device XX is waiting.

Device XX is processing a print job.

Device XX is initializing.

Device XX is warming up.

Device XX is low on toner.

Device XX is out of toner.

Device XX cannot print the current page.

Device XX has an error. User intervention needed.

Device XX has run out of memory.

Device XX door is open.

Device XX status is unknown.

Device XX is in power save mode.

Job on device JJ is paused.

An error with the job in device JJ.

Job is being deleted on device JJ.

Job is spooling on device JJ.

Job is printing on device JJ.

Device JJ is offline.

Device JJ is out of paper.

Job has printed on device JJ.

Job has been deleted from device JJ.

The driver cannot print the job on device JJ.

Device JJ has an error. User intervention needed.

Job on device JJ has been restarted.

Job is sent to the device JJ.

Frequently Asked Questions

Jobs are getting stuck in the active folder of the device, what do I do?

- Check the status of the device and the jobs in the device using windows printer control panel or Printnet for Printronix printers. If there are any errors correct the error.

The jobs should start being printed.

- If in spite of clearing errors at the printer does not help, check the log files for information on reason of failure and take necessary corrective actions, most of the times restart of bqm_alr.exe from CIMPPLICITY Process control helps.
- If the above two steps do not help restart the project, as a worst case scenario, you may have to restart the server.

The printer is printing junk characters or black patches. What should I do?

- Check the TCP/IP printer properties.
 1. Select a TCP/IP printer in the Windows Printers (or Printers and Faxes) window.
 2. Click File>Properties on the window's menu bar.

The <Printer name> Properties dialog box opens.

1. Select the Ports tab.
2. Make sure the TCP/IP printer is selected in the list.
3. Click Configure Port.

The Configure Standard TCP/IP Port Monitor dialog box opens.

Note: Select the Port Settings tab if it is not the only tab.

1. Configure the settings as follows.

A	Check LPR
B	Check _PR Byte Counting Enabled.

- Restart the printer; many times this can help resolve the issue.
- (If the settings are correct and restarting the printer do not resolve the issue) try reinstalling the printer drivers.

How do I resolve problems when I use Cimprinter as a port for the broadcast device:

Ensure that you use TEXT as your default datatype.

1. Select the printer that uses Cimprinter as a default the Windows Printers (or Printers and Faxes) window..
2. Click File>Properties on the window's menu bar.

The <Printer name> Properties dialog box opens.

1. Select the Advanced tab.
2. Click Print Processor.
3. Select TEXT in the **Default datatype** box.

Broadcasts to suppliers on the FTP are missing. How do I check this?

Check the following as follows.

1. Jobs in the history folder of the supplier device.

If you find the jobs in history folder:

1. Check for the same jobs in the document delivery project directory.

The path is:

<Project name>\DocumentDelivery\<FTP object name>Flush.

1. Check the `order_id` of any jobs that are in flush folder.

If the jobs you are looking for are in the flush folder:

1. Check for documentdelivery.log in the Document Delivery project\log folder.
2. Check for failures with the selected FTP connection,

Failures may be caused by connection errors while document delivery tried to post the jobs to the supplier FTP.

1. Addition checks include:

Check if you can:

- Connect to the FTP server manually.
- Put files on the FTP server.