Plant Applications 8.2

Overview

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.

© 2021, 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

Table of Contents

Products Overview

The primary role of Plant Applications in a plant information system is to bridge the gap between continuous time-based data and the manufacturing and business transaction-based data that is found throughout the plant. Plant Applications has powerful tools that segment the continuous data by the discrete events in order to make decisions concerning grade performance and process optimization. These tools perform this critical function by detecting the important events in the plant, reconciling continuous and event-based data, setting the context of the data through business rules and definitions, and summarizing key variables for long-term performance analysis.

Plant Applications, beginning with version 6.1, enhanced the implementation of holistic operations performance management solutions for batch, discrete, and mixed environment on one platform using equipment-, product-, and people-centric models. Although previous Plant Applications databases are supported, version 6.1 introduced a Unified Manufacturing Database (UMDB) that merges the Plant Applications data model with the Proficy S95 data model. The combined data model offers these useful features:

- Interoperability of the Proficy Client and Plant Applications user interfaces
- Data entry through Proficy Client interfaces
- Expanded list of data methods to create custom displays and forms to make use of the unified data models

Refer to the *Plant Applications Getting Started Guide*, which is accessible from the Installation Menu for the software, for further information. It includes a checklist for Plant Applications Implementation.

Installed Components

Plant Applications includes the following user interfaces and major components:

Plant Applications Server (Installed by ProficyServer.exe)

- Installs the Plant Applications database, or installs a Unified Manufacturing Database (UMDB) that merges data tables from Plant Applications data structures to Proficy SOA (S95) data structures.
- Installs several Proficy Server services. The Proficy Server services perform many functions, such as interfacing to the process historian, detecting events, performing calculations, and managing real-time interaction with client applications.

Plant Applications Client

- Contains all Plant Applications displays.
- Provides a customizable environment.
- Contains built-in cross-application automation.
- Allows for direct access to content on Plant Applications Web Server.
- Different Plant Applications modules enable different types of displays in the Plant Applications Client. Each display is also exposed as an ActiveX control, and may be used outside of the Plant Applications Client in any ActiveX container, such as the iFIX Workspace.

Details for the Plant Applications Client end-user configuration are stored on the server under the users' log-on names. This provides the users with their own look and feel, no matter which computer they use.

Plant Applications Administrator

- Provides all configuration and management for Plant Applications, including the aspecting of entities between Proficy SOA and Plant Applications when a Unified Manufacturing Database (UMDB) is installed.
- Includes all product management functionality, such as maintaining recipes.
- Includes management of all Proficy Servers, the Plant Applications Web Server, and Web Parts.

Proficy Client (Proficy Workflow Client)

- Enables data entry for Plant Applications entities through the convenient Proficy Client (Proficy Workflow Client) interface
- Provides data methods for Proficy Workflows (with license) to design custom displays and forms to gather, monitor, and analyze data related to plant operations

Note: To enable the Proficy Client UI and automatic aspecting, set the UseProficyClient parameter in Plant Applications to "true." When enabled, models created in Plant Applications are automatically aspected to SOA and SOA menus are displayed to work with the plant model with changes linking back to Plant Applications. Also, "unitless" variables (for example, material properties aspected to Plant Applications) are displayed under S95 Department when the Proficy Client UI is enabled. In the Plant Applications Administrator, follow these steps to enable the Proficy Client UI:

1. Open the Server Manager by connecting to a server and double-clicking the server name under Plant Applications Server.

*2. Select **Administer Site Parameters** under Global Configuration*

3. Set the value for the UseProficyClient parameter to True. (Click on the button in the Edit column to change the value.)

Refer to "MESCore Service Provider Documentation" at the Contents tab for information about using the Proficy Client.

Proficy Vision Displays

Web Client Displays are no longer supported as of Plant Applications 6.3.

- Provides access to displays, through the Proficy Vision interface, that are created in Proficy Workflow.

Plant Applications Web (Reports) Client

- Provides access to the Plant Applications Web server.
- Is the primary interface for users.
- Controls access to information.
- Provides access to standard reports.
- Provides access to user-defined or site-specific reports.

Plant Applications Add-In for Excel

- Exposes Plant Applications data in Microsoft Excel.
- Is a common tool for ad hoc analysis and reporting.
- Can publish Excel reports to the Web.

Approach to Implementation

In general, every site must perform certain steps regardless of the industry:

1. Add Plant Applications site users.
2. Build your plant model.
3. Capture production events.
4. Define products.

Because Plant Applications is highly configurable, GE Intelligent Platforms offers a complete array of support services to help you attain the highest value possible from your technology investments. For more information, visit us at https://digitalsupport.ge.com.

Administrator User Interface

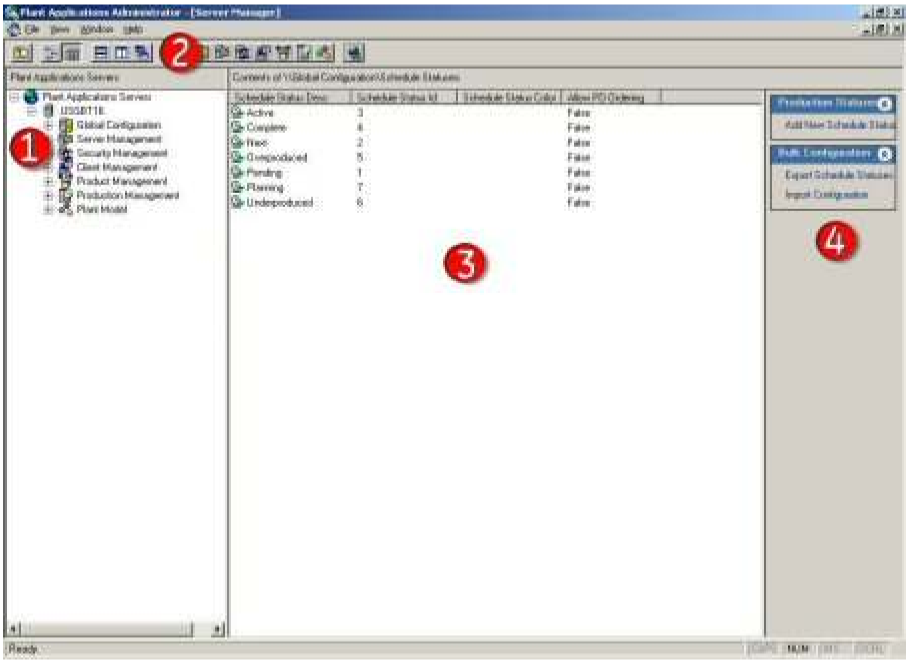
For information on administration common dialogs, refer to *Plant Applications Common Dialogs* help file.

Plant Applications Administrator Environment

The Administrator is the client application for administrators configuring the system. The Plant Applications Administrator is used for these tasks:

- Configuring data captured from the plant floor and other sources.
- Creating security rights for data input, data changes, data verification, viewing data, and so on.
- Processing collected information using calculations; and many other data functions.
- Providing the link and the structure to the SQL database where the data is stored.
- Designing displays that viewed through the Plant Applications Client application.
- Providing a method for importing and exporting Plant Applications data.

The Plant Applications Administrator environment is similar to using Windows® Explorer. On the left side of the main screen is the Plant Applications Server Manager tree. From this hierarchical tree all tasks can be performed. For the most part, every object in the Server Manager tree can be right-clicked to view a menu of the available commands. The center pane displays additional details about objects directly below the currently selected level. The right pane displays the hyperlink menu, if you have that option enabled.



Schedule Status Desc	Schedule Status Id	Schedule Status Color	Allow PO Ordering
Active	3		False
Complete	4		False
Next	2		False
Overproduced	5		False
Pending	1		False
Planning	7		False
Underproduced	6		False

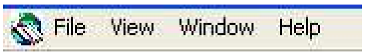
1 Right-click on most objects in the Plant Applications Server Manager tree to view a menu of available commands.

2 The Plant Applications toolbar provides a shortcut to the various levels in the Server Manager tree.

3 The Detail pane displays the contents of the object selected in the Server Manager tree.

4 If the Hyperlink button is active, all of the tasks that can be performed on the object selected in the Server Manager tree are presented as hyperlinks.

Administrator Menu Options



File menu: From the File menu, you can add, edit, or delete servers, and close the Plant Applications Administrator

View menu: From the View menu, you can expand or collapse the current node (or branch), expand or collapse the entire Administrator tree, change how product information is viewed, and display a list of icons.

Windows menu: From the Window menu, you can cascade or tile any open modal windows, arrange any minimized windows, and view other open windows.

Help menu: From the Help menu, you can view the error message log, access the Plant Applications online help file, and view versioning information about Plant Applications and your operating system.

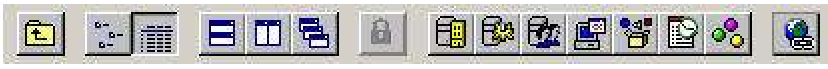
Logging On to the Administrator

Access rights are set up under Security Management. If you are using Windows Authentication and logging into a domain other than the default domain, you will need to include the fully qualified domain name (FQDN). Contact your site administrator if you need assistance.

If you are challenged for authentication, follow these steps:

1. Select **Proficy Authentication** to submit a user name and password defined in Plant Applications. Optionally, select **Windows Authentication** to submit credentials that are your Windows authentication credentials.
2. If choosing **Windows Authentication**, choose the domain name from the drop-down selection. If your domain is not listed, enter the fully qualified domain name (FQDN) for the Windows domain.

Administrator Toolbar



Move up one level in the Server Manager tree.

Change the view in the Detail pane so that no detail information is displayed.

Change the view in the Detail pane so that additional detail information, such as product ID, is displayed. This button is active by default.

Tile any open windows horizontally.

Tile any open windows vertically.

Cascade any open windows.

Force authentication of changes made in the Administrator.

Open Global Configuration.

Open the Server Management node.

Open Security Management.

Open Client Management.

Open Product Management.
Open Production Management.
Open Plant Model.
Hide or display the hyperlinks menu.

Unified Manufacturing Database (UMDB)

Version 6.1 of Plant Applications introduced a Unified Manufacturing Database (UMDB) that provides access to Proficy SOA and Plant Applications data by using Aspect extensions to combine the databases. A new service provider, the MESCore Service Provider (SP) provides extensions to merge the databases along with methods for custom displays and forms. The UMDB enables you to define a plant environment in an S95 structure while providing a strict and formal definition that was not possible with Plant Applications.

You can continue to use the Plant Applications Administrator to do configuration when data is not aspected to create a UMDB. However, once aspected (linked), configuration is mostly done using the Proficy Client. For example, quality form and calculation manager functions are now available through the Proficy Client. Also, Proficy Workflow provides access to selected Plant Applications functionality and enables browsing of Plant Applications resources such as production lines and units, variables, products and events. The new architecture allows calls from the Proficy Client to access Plant Applications data and event services.

For information on how to work with a merged system and UMDB, refer to *Merged System and UMDB* help file.

Setup and System Configuration

For information on licensing, refer to *Plant Applications Licensing* help file.

For information on security management, refer to *Plant Applications Security Management* help file.

Administer Licensing

Previous releases of Plant Applications support one of two models for licensing:

- Server Module/Component Model: This model requires a license for each of the four server modules, along with separate concurrent user licenses for reports, Web parts, the SDK, and module-specific displays.
- Functional License Model: This model uses interchangeable Server Functional Licenses (SFLs) for individual functions and a Client Access License (CAL) that allows one user access to all of the displays, the SDK, Excel Add-in reports, and Web reports and Web parts. Introduced with Plant Applications 5.0, this model provides a more cost-effective entry point and is more scalable and flexible than the Server Module/Component model.

Beginning with Release 6.2, licensing is simplified for management of concurrent users. The Plant Applications Server is licensed based on the number of concurrent users that connect to the server, and a concurrent user has access to all modules and functions in Plant Applications.

Licensing of individual modules is an additional option. Licensing for Release 6.2 supports module-level licensing for customers having previous licensing for the Efficiency, Quality, and Production modules.

With Release 6.2, the Server Function Licensing (SFL) model has been discontinued. Refer the *Plant Applications Important Product Information (IPI)* or the *Getting Started Guide* for more information.

Server Management

For information on server management, refer to *Plant Applications Server Management* help file.

Historian Data Collection and Use

Plant Applications interacts with historian data to transform the time-series data in context to produce events, calculations, and summarization for state-of-the-art manufacturing execution. Without an historian, you are limited to manual entry of data into a Plant Applications model. Refer to "Historian Connections" to set up connections for particular historians including a Proficy Historian.

In a manufacturing environment, attributes and events related to the various components and events are captured in real time by control devices (for example, DCS or PLC devices), which send data to an historian that collects, archives, and distributes the real-time data. The data includes details such as a tag ID, date/time stamp, and value. You can configure Plant Applications to read the information from the historian, manipulate the data, and place the data in the context of a particular event. Plant Applications can trigger events such as an alarm and downtime. The data is available to include in a variety of displays, web reports, and Web Parts.

For example, consider a bottle labeling machine in a bottling plant. The labeler has many attributes such as number of labels applied per minute, quantity of glue available, glue temperature, machine status. You might create a Plant Applications variable to track glue temperature with a corresponding historian tag that captures the temperature data over time. By setting up a time-based event in Plant Applications, you can track temperature changes and perhaps set off an alarm if the glue temperature exceeds a specification. A report can be generated to show temperature fluctuations.

Setting Up an Historian

Refer to "Accessing Real-Time Manufacturing Data through an Historian" in the Plant Applications Getting Started Guide. When the Excel Add-In is installed, you will see an Historian menu just below the Excel main menu bar. The Getting Started Guide explains how to import historian tags. Refer to the historian documentation to set up tags.

Using Historian Data

Historian data can be used in the following ways:

- Event detection uses historian data to watch for triggers (for example, production of a part on a line) and generates events in Plant Applications such as production, downtime, waste, and product changes. There are many event models in Plant Applications.
- Event-based "summarization" provides a summary of data around an event, for example, average machine speed for a reel turn up on a paper machine.
- Time-based "summarization" calculates averages for daily, hourly, or even yearly periods.
- Data and specification transfer writes data back down to the historian. For example, you can write specification data based on a product change or lab entry done on an Autolog screen for an operator to use.
- Calculations, although not historian events, are related to what can be done with historian data as related to Plant Applications events. Plant Applications watches for events and calculates other information based on the data. For example, a calculation such as A + B + C can be run whenever an associated event occurs or at some time interval.

Functions Involving Plant Applications Services

Historian-related functions involve the following Plant Applications services:

- Event detection–handled by the event manager
- Event-based summarization–handled by PA Summary Manager
- Time-based summarization–handled by PA Reader
- Data and Spec Transfer–handled by PA Writer and, for specifications Summary Manager
- Calculations–handled by PA Calculation Manager

Historian Connectors

To interact with historian data, a connection must be established with the historian. Plant Applications offers a set of eight "out-of-the-box" connectors that are used by services to talk to particular historians. Each connector is an API (DLL), each having the same basic design. Because the DDLs are labeled "PR2," the connectors are commonly referred to as "PR2s." Connecting to particular historians is described elsewhere in Plant Applications Help. Although an historian typically runs on the Plant Applications server, a Remote Data Service (RDS) is offered so the historian can run remotely. For example, it is typical to have Plant Applications connect to a “PI” server through RDS on a remote node.

NOTE: If using an OPC connector, choose one that supports a History Data Access (HDA)interface in order to have historical data.

Performance Considerations

It is recommended to configure data compression for an historian; otherwise, processing the large data volume to do calculations, for example, can be slow. It is also recommended to enable caching to ensure the best reading of data by services.

Global Configuration Tasks

For information on global configuration tasks, refer to *Plant Applications Global Configuration* help file.

Electronic Signatures

For information on using electronic signatures, refer to *Plant Applications Electronic Signature* help file.

Plant Model

For additional information on a plant model, refer to *Plant Applications Plant Model* help file.

Introducing the Plant Model

The main function of the plant model in Plant Applications is to describe definable properties of discrete operating units (equipment) that make up production lines.

Plant Applications provides a five-level hierarchical tree for creating a model of your plant layout. The five levels are:

- **Department:** The Department is the top level of the tree and can be used to group production lines that produce similar products.

- **Production Line:** Production Lines are a collection of equipment that operates together to produce a product.
- **Production Unit:** Within the production line, production units are used to define a discrete unit of operation and which are capable of consuming materials and producing products. The production unit is the key element in the plant model where models, variables, calculations, products, recipes, and production events are associated. Because production units represent independent events, such as downtime, waste, and production runs, a production unit in the Plant Applications plant model may or may not be a physical piece of equipment on your plant floor. Production units must be defined by the events you need to capture.
- **Variable Group:** Variable Groups are used to group similar variables, which are associated with a specific production unit.
- **Variables:** Variables are discrete data elements that provide a location for data storage and generation. For example, variables can be lab tests or process values for a specific production unit.

Variables

Variables are measurements or calculations attached to a given production unit that produce values either by time or event. Variables can be defined as a discrete data element that provides a location for data storage or generation. Variables are usually lab tests, process values, and so on, for a specific production unit.

The data that populates a variable can be obtained from various data sources. These data sources may need to be interfaced to the system. There are many data types that can be applied to the variable, such as float, string, integer, and so on. You can also apply custom data types. If the data from a variable is being captured in time, rather than by event, then there are many different sampling types (pre-defined simple calculations) that can be applied to the variable.

- Historical variable values are stored in the database.
- Variable data, whether captured manually or automatically, is associated with a production unit event.
- Security applied to a variable overrides security applied to anything above the variable, such as the production unit or line.
- Variables can be moved from one variable group to another
- Variables can be copied between production lines, but they can no be moved.

Variable Properties

Variable properties must be set for a variable to function as intended and produce the correct data. Variable parameters are configured using the Variable Sheet. The Variable Sheet can be accessed at any level of the plant model.

The three key variable parameters are:

- The Data Source (where is the data coming from)
- The Data Type (is it a string, float, integer, custom data type, and so on)
- The Event Type, which dictates when data is collected. Time is the default event type unless an event is configured against the production unit.

Calculations

Calculations are manipulations of data to generate a desired result. A simple conversion on a variable's value from imperial to metric measurement is a calculation. Calculations can determine Key Performance Indicators and can perform statistical analysis on raw data. They can also filter out data and drive disposition and product movement.

There are two types of calculation definitions within Plant Applications:

- Predefined sampling types, which typically aggregate or summarize data
- User-defined calculations, allow almost unlimited capability to create new sampling logic by using flexible coding environments, such as equations, VBScripts, and stored procedures.

The data sources for a calculation are the same for a variable, manual input (lab/test results), derived from another Plant Applications variable (base variable), or processed through an historian. All Plant Applications data is exposed for a calculation. One or many variables can be used as inputs to a calculation. Data from a calculation is either returned as a single result, or for stored procedure type calculations can additionally be returned in the form of a SQL result set. Result set returns can be used to return results for several variables at a time, or may be used to trigger additional events. The result of a calculation in Plant Applications is saved as a variable value.

Displays and Views

For information on client management, refer to *Plant Applications Client Management* help file.

Plant Applications Client

The Plant Applications Client provides a customizable environment for using Plant Applications views and displays to access plant data. Topics about the Client describe:

- Setting up and managing the Client
- Using the Client

> **TIP:** If you are using Windows Authentication and logging into a domain other than the default domain, you will need to include the fully qualified domain name (FQDN). Contact your site administrator if you need assistance.

Different Plant Applications modules enable different types of displays in the Plant Applications Client. Each display is also exposed as an ActiveX control. These displays can be used outside of the Plant Applications Client in any ActiveX container such as Intellution's iFIX Workspace.

Details for the Plant Applications Client end-user configuration are stored on the server under the users' log-on names. This provides the users with their own look and feel, no matter which computer they use.

Plant Applications Client sessions can be configured to time out and enter a protected state, thus requiring a user to log back in. Refer to Configuring the Client Session Timeout.

Selected Plant Applications displays can be viewed through the Proficy Client installed by Proficy Workflow. Plant Applications licensing is required. Refer to Using Plant Applications Displays in the Proficy Client.

For more information on Proficy client, refer to *Plant Applications Client* help file.

Models and Events

For more information on models and event detection, refer to *Plant Applications Models and Event Detection* help file.

Events Handling

Plant Applications models are used to capture events such as reel turn-ups, batch starts/stops, product changes, consumption of product on a production line, and so on. The Event Manager Service watches for different types of data to change and creates events based on that data. Events are created based on specific models, which describe what data to watch, and how to interpret that data to produce an event. For example, an event model may call for the event management service to watch a set of tags in the historian to capture when a batch was started or completed, capture or create the batch ID, and capture or create the product code of the batch.

Product Management

For more information on product management, refer to *Plant Applications Product Management* help file.

Products

Products (or "grades" in the paper industry) are the materials manufactured on the production lines and may be **in-process products** or **end products**. Plant Applications provides two fields for naming the products, the **product code** and the **product description**. Specifications are entered for products.

Product Families

Product families are used as the first grouping of products. A product must be a member of only one product family. the product family can be used to control security on products.

Product Groups

Products can be arranged into families called **product groups** and a product can reside in more than one product group. This helps organize the product structure and assist in the analysis of historical data.

Product Properties

Product properties are used to describe related parameters, such as color, packaging or recipes, associated with products. Within product properties, you can describe characteristics and define specifications.

- **Characteristics** describe a particular property. For example, the property of **color** can have a characteristic of red.
- **Specification variables** describe the attributes of a product property and are used to maintain recipe parameters and specifications that are plant-wide rather than specific to a particular piece of equipment. Specifications entered at the product level are automatically propagated to individual products and units that are associated with a particular property and characteristic. Typically, quality specifications are maintained at the product level.

Production Management

For information on production management, refer to *Plant Applications Production Management* help file.

Plant Performance Modules

Overview of Plant Performance Modules

Plant Applications offers configurable software packages designed for a rapid return on existing technology investments. As a result, the Plant Applications solution allows for an incremental approach to purchasing, installing, and implementing the right functionality at the right time for your business. Each module features specific capabilities designed to provide you with a scalable solution that precisely meets your needs.

Quality Management

This module helps you analyze and control product and process quality. For more information, refer to *Plant Applications Handling Quality* help file.

Production Management

Use this module to track and report on production schedule execution and product genealogy. For more information, refer to *Plant Applications Handling Production* help file.

Efficiency Management

This module helps you track and monitor overall equipment effectiveness (downtime, waste, production counts).

Batch Analysis

The Plant Applications Batch Analysis solution helps manufacturers to better understand and control variation in batch related production environments.

For more information, refer to *Plant Applications Batch Analysis* help file.

Specifications

A specification defines the target value and various upper and lower quality/alarm limits for a variable and a product combination. It can be used to define the amounts of materials in a batch process, measure the actual value of a quality variable against a target and set of limits, or compare the current values of a set of process related variables against the average of a number of past sets. Specifications are defined for a specific product. They can be as general or detailed as needed and can be Production Unit/Line specific or shared across many production units/lines.

Specifications can be configured and managed within Plant Applications by three different methods. Depending upon the type of production process that needs to be modeled, a combination of these three methods can be used.

- One method is to assign specifications at the variable level for a specific product on a specific production line. This method is called **Unit Level Specification Management**. In a typical Plant Applications software installation this method is used to manage the process variable specifications for a single production unit or machine.

  The Unit Level specifications are defined at the “machine” level and they override any other limits defined at the Central and Corporate levels. They are defined directly on the variable for each product. They are the best strategy for maintaining process operating settings such as target levels, pressures, temperatures etc. The typical application would be for a single-line facility where products and variables are fairly independent.

- The second method manages specifications centrally and assigns them to the multiple variables and/or products that have them in common. It normalizes specification limits for common products and variables. This method is called **Central Level Specification Management**. This second method is normally used to manage quality related product specifications on a plant-wide basis with multiple production lines.

  The specifications are organized by properties with characteristics and specification variables. Products are associated to characteristics and variables are associated to specification variables.
- The third method manages specifications at the corporate level where specifications can be shared amongst plants on one site or many sites. This method is called **Corporate Level Specification Management**.

Overview of Efficiency Management

The Efficiency Management module enables you to monitor and control overall equipment efficiency, providing a comprehensive look at factors such as equipment downtime waste production, production quantity, and event sequences.

Efficiency Management has four key areas of focus: downtime tracking, waste tracking, production counting and user-defined event tracking. Together, this comprehensive set of capabilities helps significantly improve operating efficiencies while greatly reducing costs.

Efficiency Management makes it possible to associate each downtime event with a specific cause on a real-time basis. It provides a window into why that downtime is occurring and allows you to calculate and improve your overall equipment effectiveness. In addition, the module calculates KPIs, such as overall equipment effectiveness, and allows you to create ad hoc reports on your performance that can be shared throughout the organization via the Web.

For more information, refer to *Plant Applications Handling Efficiency* help file.

Reports

Web Server and Web Client

For more information, refer to *Plant Applications Web Reporting* help file.

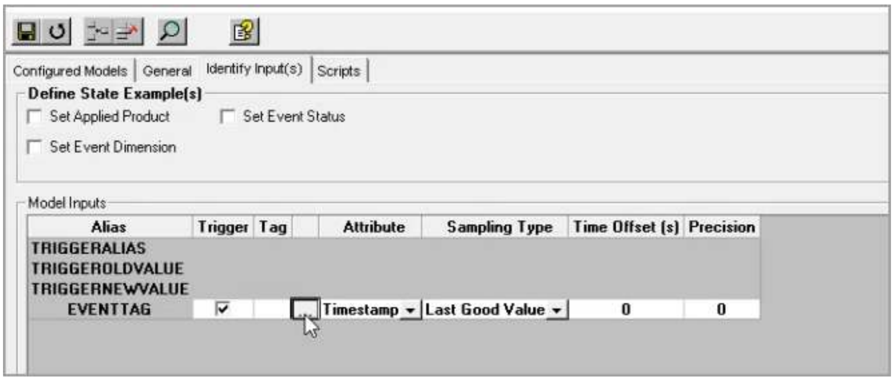
Excel Add-In for Plant Applications

The Plant Applications Add-In for Microsoft Excel is a tool you can use to create ad hoc reports that help you retrieve and analyze data stored in Plant Applications or in your historian. Because it is an Excel Add-in, you gain all the features and functionality of Excel, coupled with the ability to quickly and easily import Plant Applications data into your workbook. The combination of the two is a powerful assistant that helps you analyze and present information in unlimited ways.

For more information, refer to *PAXLA_Excel_Add-in* help file.

Configure the Administrator to Work with the Activities Application

1. Log in to the Plant Applications Administrator - [Server Manager].
2. In the **Plant Applications Servers** node, select .
3. Under the Plant Model node, add a department, production line, and production unit for the Activities application.
4. Right-click the production unit and click **Configure Events on <***production unit***>**. The Event Detection wizard appears.
5. From the **Model Type** list, select the model type as *Production Event*, and then click **Add New Model**. The model is added under **Configured Models**.
   - For user defined events, select the model type as *User-Defined Event*.
6. Double-click the production event model. The tabs for configuring the selected model type appear.
7. On the **Identify Input(s)** tab, select the browse button (...) to search and add Historian tags to the production event.



Configured Models | General | Identify Input(s) | Scripts

Define State Example(s)

☐ Set Applied Product ☐ Set Event Status

☐ Set Event Dimension

Model Inputs

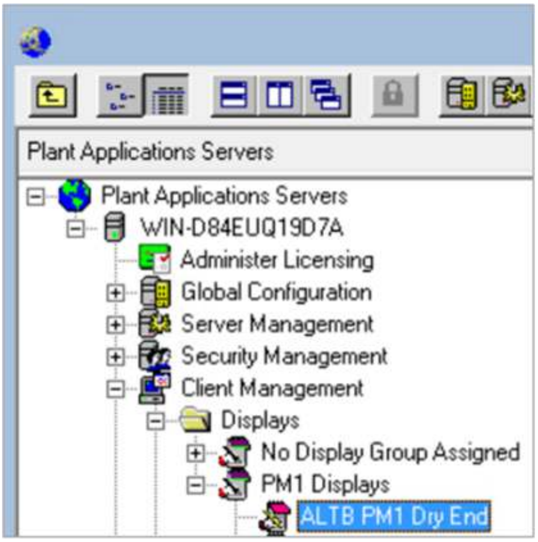
Alias	Trigger	Tag	Attribute	Sampling Type	Time Offset (s)	Precision
TRIGGERALIAS						
TRIGGEROLDVALUE						
TRIGGERNEWVALUE						
EVENTTAG	☑	...	Timestamp	Last Good Value	0	0

8. Click 💾 to save the configuration.
9. Click **Yes** when asked if you want to activate the model.
10. Select **OK** when asked to **Reload Service(s)**.
11. Under the Plant Model node, right-click the production unit created in Step 3, and create a variable group and variables.
12. Edit and configure variables in the **Variable Sheet** screen. For detailed information on the **Variable Sheet** fields, see Variable Property Definitions.
13. Save the configuration, and proceed to creating displays.

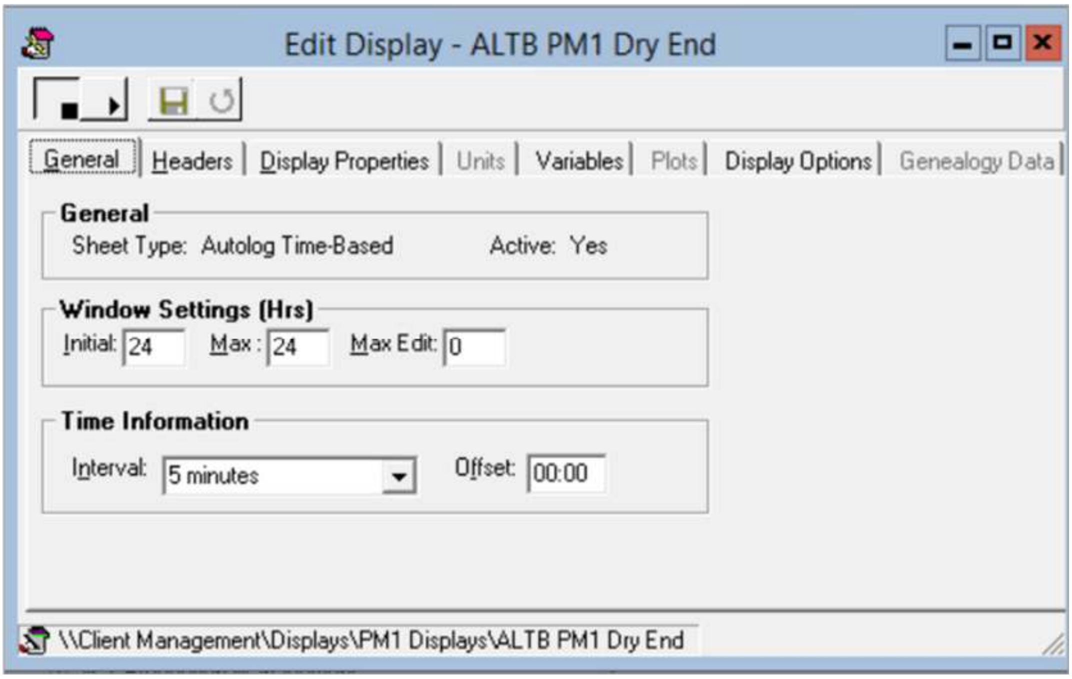
Create Displays

1. Under the Client Management node, right-click **Displays** and click **Add New Display Group**.
2. In the blank entry text box that appears, provide a unique name for the group, and press ENTER.
3. Right-click the newly created display group, and navigate to **Add New Display** > **Autolog Production Event**.
   - For user defined displays, select Add New Display > Autolog User-Defined Event.
   - For time based displays, select Add New Display > Autolog Time-Based.

4. In the blank entry text box that appears, provide a unique name for the display, and press ENTER. For example, in the following image, **PM1 Displays** is the name of the display group and **ALTB PM1 Dry End** is the name of the Autolog display associated with the Activities application.



5. Right-click the newly created display, and select **Edit <***display name***>.** The **Edit Display** screen appears.



General
Sheet Type: Autolog Time-Based   Active: Yes

Window Settings (Hrs)
Initial: 24   Max : 24   Max Edit: 0

Time Information
Interval: 5 minutes   Offset: 00:00

\\Client Management\Displays\PM1 Displays\ALTB PM1 Dry End

6. Visit these tabs to provide values:

Tab	Field	Description
General	Windows Settings (Hrs)	Select the default time window of the display by updating the Initial and Max window hours.
	Master Unit	Select the Production Unit from the list. The variables under this unit are linked to the display.
	SubType	Select the event type added for Activities application from the list.

	Time Information - Interval	For autolog time based displays, set the interval time to generate an event. If the interval is set to 1 hour, an event is generated every hour.
Display Properties	Column Headers	Select the check box for *Engineering Units*. You can also select other columns if you want to display them in the Web Client.
Variables	Add	Select to search for variables in the **Search Variable** screen, and add them for the display. Refer to the topic on *Search Variable dialog box* in this guide.
	Add Title	Select to create titles for activities. Use **Up (-)**, and **Down (+)** to move titles and arrange variables under them. The variables are grouped by title. Provide an order number next to each title to set a sequence for variables that appear for activities created in the Web Client. ***NOTE:*** *To set an order number for titles, set the display option ActivitiesPerTitle to True.* When you create titles and set their activity order in the Administrator, these activities become available to perform in the same order as configured, in Plant Applications Web Client.
Display Options	CreateActivities	By default, the value is **False**. Select ... in the **Edit** column to modify the value to **True**. See Display_Options for information of other options.
	Auto Complete	
	UnitForActivities	

7. Click 💾 to save the configuration.

Display Options

To modify the value of an option, select ... in the **Edit** column.

Activities

Display Option	Description
ActivitiesPerTitle	Creates activities for each title configured in the **Variables** page within the **Edit Display** page. • If set to **True**, when you create an event in the Activities application, the activities are created for each title configured in the Variables page. ***NOTE:*** *You can set this to **True** only when you have at least one **Title** set in the **Variables** tab. If there is no title available, a message appears - "There is no Title set, 'ActivitiesPerTitle' display option is turned on".* • If set to **False**, only a single activity is created for an event in the Activities application in the Web Client. By default, the option value is set to **False**.

Activity Status Security	Enables you to override the standard locking security of the activity for a role. If you as an operator have access equal to or higher than what is set, you can override the security and release the lock for an activity. The available values are: • **Manager**: Enables users in the Manager role or in a role with access same or greater than the Manager role to override the standard locking security of the activity and unlock the activity. • **Admin**: Enables users in the Admin role to override the standard locking security of the activity and unlock the activity. • **Read**: Enables all users to only read a locked activity. • **Read/Write**: Enables all users to read and modify a locked activity. By default, the option value is set to **Manager**.
ActivityAlias	Enables the entry of an alternative name for the activities in the Web Client. If no value is provided, the following values are the default names for the activities in the Web Client: • Timed-based activities – [Sheet_Description], where Sheet_Description is the description entered for the autolog sheet. • Production-based activities – [Sheet_Description – Event_Number], where Sheet_Description is the description entered for the autolog sheet and Event_Number is the number entered for the production-based event. • User-defined activities – [Sheet_Description – UDE_Number], where Sheet_Description is the description entered for the autolog sheet and UDE_Number is the number entered for the user-defined event. By default, no value is set for the option
ActivityDisplayType	Enables the definition of the type of activity created in the Activities application The available values are: • **Standard**: Enables the use of an autolog sheet for a display in the Activities application • **Custom**: Enables the use of custom forms for a display in the Activities application. When you select this option, the **Custom Form Settings** section appears in the **Display Options** page. In the section, you define the settings for enabling custom forms. For information about how to build custom forms for use in the Activities application, refer to the About Custom Forms section in the Plant Applications Web Client Help. ***NOTE:*** *The **Custom Form Settings** section appears in the **Display Options** page only when the value of the **ActivitiesPerTitle** option is set as **False**.* *If the value of the **ActivitiesPerTitle** option is set as **True**, you can enter the value for custom form settings in the **Activity Data** window. To access the window, select the **Variables** tab, then select a variable title, and then select **Activity**.*
Auto Complete	Enables to automatically set activities to Complete.

	- If set to **True**, after entering the values for mandatory variables, the activity is automatically set to **Complete** in the Web Client. - If set to **False**, the activity is manually completed only after you select **Complete** for the activity in the Activities application in the Web Client. By default, the option value is set to **False**.
CreateActivities	Enables the creation of an activity after you create an event in the Activities application in the Plant Applications Web Client. To use the option, set the option value to **True**. Else, set the value to **False**. By default, the option value is set to **False**.
Execution Start (min)	Indicates the time in minutes after which an activity must start after the activity was created. By default, the option value is set to **0**. ***NOTE**:* *This option appears in the **Display Options** page only when the value of the **ActivitiesPerTitle** option is set as **False**.* *If the value of the **ActivitiesPerTitle** option is set as **True**, you can enter the value for the **Execution Start** option in the **Activity Data** window. To access the window, select the **Variables** tab, then select a variable title, and then select **Activity**.*
Lock Activities	Enables to lock activities by the user. - If set to **True**, the activity is locked when you select **Perform** for the activity in the Activities application in the Web Client. Other operators can view but cannot perform the locked activity until you release the activity. - If set to **False**, any user can perform an activity at any time. By default, the option value is set to **True**.
Overdue Task Requires Comment	Enables to enter an explanatory comment for an overdue activity. - If set to **True**, the overdue activities require an explanatory comment. - If set to **False**, the overdue activities do not require any explanatory comment. By default, the option value is set to **False**.
System Complete Activity On New Event	Automatically sets the existing activities to Complete when an activity created in the Plant Applications Web Client has the same product ID, process order ID, and unit ID of any existing activities. To use the option, select one of the following values: - **Do Not System Complete**: Does not automatically set the activities status to complete. - **System Complete Honour Mandatory**: If the activities have mandatory variables assigned to it, then the status of the activities is set to complete. - **System Complete Ignore Mandatory**: The status of activities is set to complete irrespective of whether mandatory variables are assigned to the activities or not. ***NOTE**:* *If you use this option, then you must set the **System Complete Activity Frequency** parameter value to indicate the frequency in*

	minutes at which the application triggers the event that completes the activities. You can set the value of the parameter in the range of ***1*** *to* ***60****. The default value is* ***5****. You set a parameter value in* ***Administrator Site Parameters*** *in the* ***Global Configuration*** *section of the administrator.* *If the value for the* ***Target Duration (min)*** *and* ***System Complete Activity On New Event*** *options are same, then it implies that an activity must be set to Complete when the activity is marked as overdue*
System Complete Duration (min)	Indicates the time in minutes after which an activity is automatically set to Complete. By default, no value is set for the option. ***NOTE****:* *This option appears in the* ***Display Options*** *page only when the value of the* ***ActivitiesPerTitle*** *option is set as* ***False****.* *If the value of the* ***ActivitiesPerTitle*** *option is set as* ***True,*** *you can enter the value for the* ***AutoComplete Duration*** *option in the* ***Activity Data*** *window. To access the window, select the* ***Variables*** *tab, then select a variable title, and then select* ***Activity****.*
Target Duration (min)	Indicates the time in minutes after which an activity must end after the activity was started. If the operator exceeds the time, the activity is categorized as an overdue activity. By default, the option value is set to **0**. ***NOTE****:* *This option appears in the* ***Display Options*** *page only when the value of the* ***ActivitiesPerTitle*** *option is set as* ***False****.* *If the value of the* ***ActivitiesPerTitle*** *option is set as* ***True****, you can enter the value for the* ***Target Duration*** *option in the* ***Activity Data*** *window. To access the window, select the* ***Variables*** *tab, then select a variable title, and then select* ***Activity****.*
UnitForActivities	Specifies the unit to be associated for activities. This option appears only for the autolog time-based display. By default, no value is set for the option.
VariableScrolling	Specifies the scroll type of the variables view in the autolog screen. • If set to **Scroll**, the variables view in the autolog screen scrolls. • If set to **Pagination**, the variables view in the autolog screen is paginated based on titles of the variables. By default, the option value is set to **Scroll**. ***NOTE****: If the* ***ActivityDisplayType*** *option is set to Custom and the Open URL Configuration option is set to Autolog payload, then only the Scroll option is available.*

Custom Form Settings

Display Option	Description
External URL Link	Specifies the URL of the custom form. This field is mandatory.

	***NOTE**: If the value of the **ActivitiesPerTitle** option is set as **True**, you can enter the value for the **External URL Link** option in the **Activity Data** window. To access the window, select the **Variables** tab, then select a variable title, and then select **Activity**.*
Open URL Configuration	Specifies the area in the Activities application where the custom form appears The available values are: • If set to **Autolog Embedded**, the custom form appears as a replacement of the autolog sheet within the Activities application. • If set to **Autolog payload**, the custom form appears in an embedded pane in the page displaying the autolog sheet. • If set to **New Browser**, the URL of the custom form opens in a separate page. ***NOTE**: If the value of the **ActivitiesPerTitle** option is set as **True**, you can enter the value for the **Open URL Configuration** option in the **Activity Data** window. To access the window, select the **Variables** tab, then select a variable title, and then select **Activity**.*
Password	Specifies the password associated with the **User Login** option to log in to the URL of the custom form. ***NOTE**: If the value of the **ActivitiesPerTitle** option is set as **True**, you can enter the value for the **Password** option in the **Activity Data** window. To access the window, select the **Variables** tab, then select a variable title, and then select **Activity**.*
User Login	Specifies the user name to log in to the URL of the custom form. ***NOTE**: If the value of the **ActivitiesPerTitle** option is set as **True**, you can enter the value for the **User Login** option in the **Activity Data** window. To access the window, select the **Variables** tab, then select a variable title, and then select **Activity**.*

Security

Display Option	Description
LockUnavailableCells	Disables modifying values of variables that are blank for users other than administrators. • If set to **True**, only users in the Administrator security group with the Access Level other than Read can enter values for the modifiable autolog variables that have no available data. • If set to **False**, any user with the Access Level defined as Admin, Manager, or Read/Write can modify all variable values. By default, the option value is set to **False**.

Configure the Administrator to Work with the Downtime Application

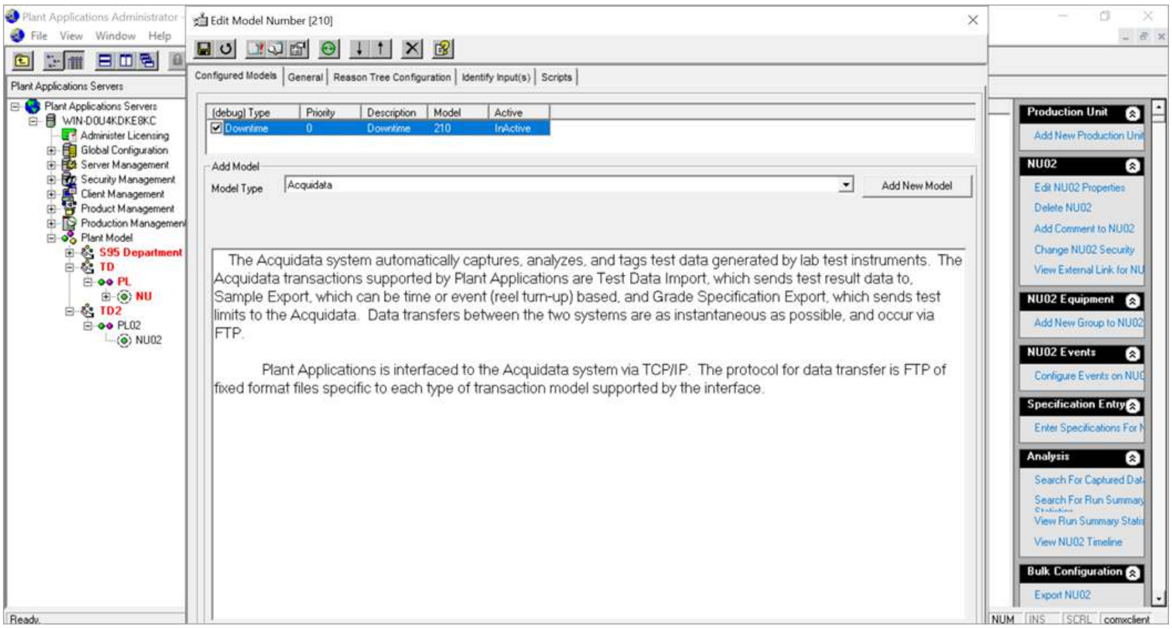
1. Log in to the Plant Applications Administrator - [Server Manager].

2. In the **Plant Applications Servers** node, select .

3. Under the Plant Model node, add a department, production line, and production unit for the Downtime application.

4. Right-click the production unit and click **Configure Events on <*production unit*>**. The Event Detection wizard appears.

5. From the **Model Type** list, select the model type as Downtime, and then click **Add New Model**. The model is added under **Configured Models**.



(debug) Type	Priority	Description	Model	Active
☑ Downtime	0	Downtime	210	InActive

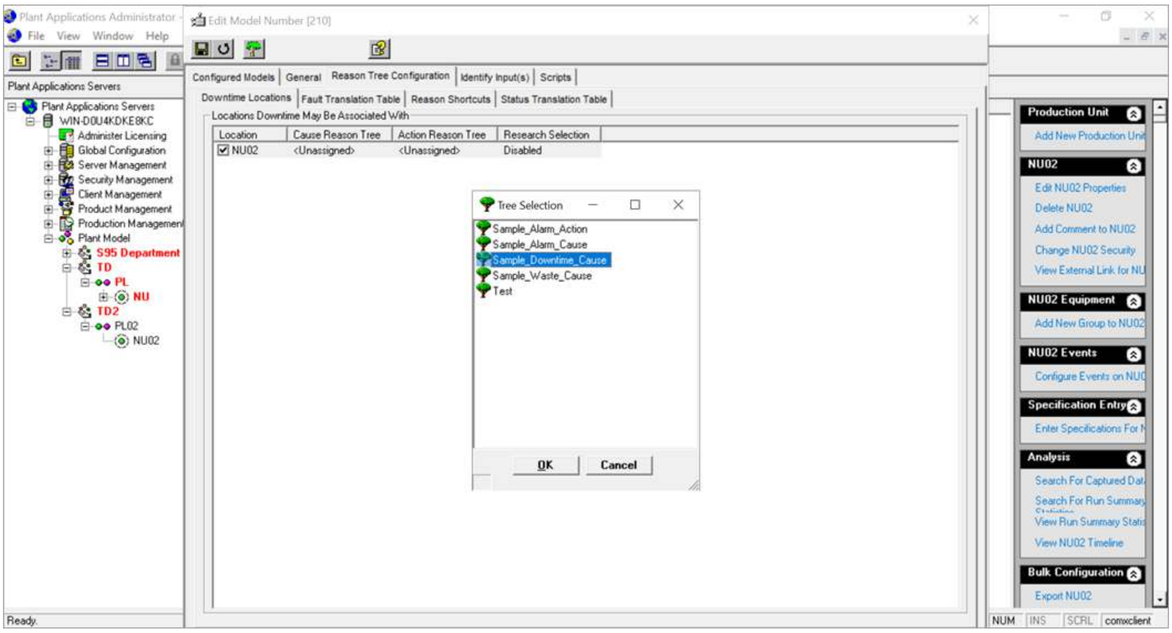
Model Type: Acquidata

The Acquidata system automatically captures, analyzes, and tags test data generated by lab test instruments. The Acquidata transactions supported by Plant Applications are Test Data Import, which sends test result data to, Sample Export, which can be time or event (reel turn-up) based, and Grade Specification Export, which sends test limits to the Acquidata. Data transfers between the two systems are as instantaneous as possible, and occur via FTP.

Plant Applications is interfaced to the Acquidata system via TCP/IP. The protocol for data transfer is FTP of fixed format files specific to each type of transaction model supported by the interface.

6. Double-click the downtime model. The tabs for configuring the selected model type appear.

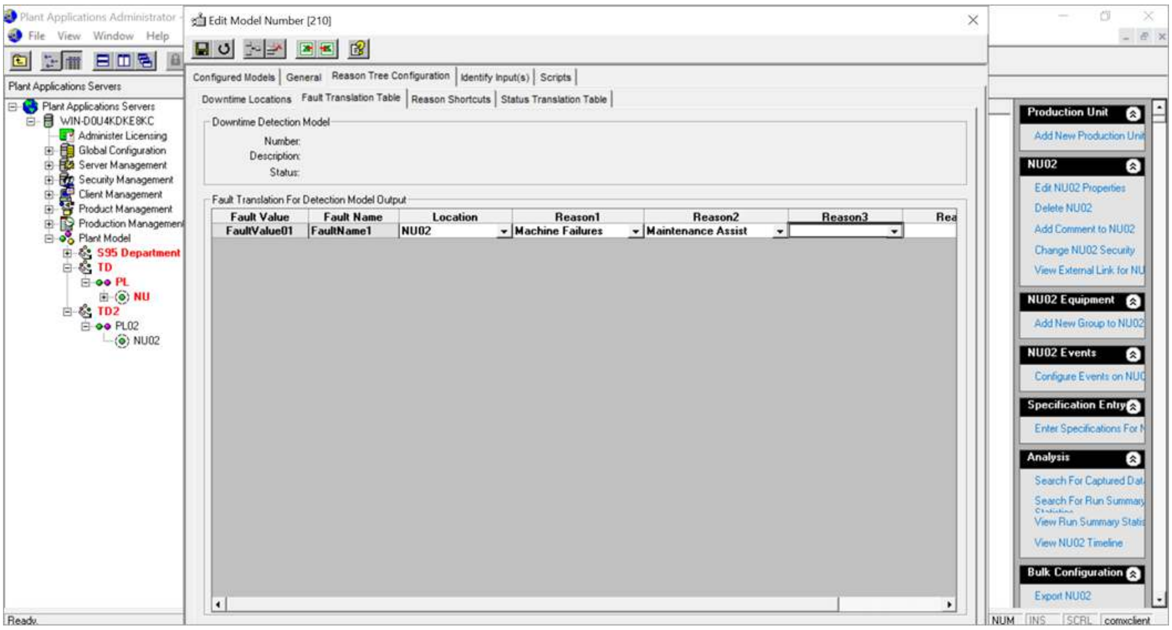
7. Navigate to the **Reason Tree Configuration** > **Downtime Locations** tab, and select the unit location to be associated with the downtime model. The **Tree Selection** appears.



Location	Cause Reason Tree	Action Reason Tree	Research Selection
☑ NU02	<Unassigned>	<Unassigned>	Disabled

8. Select a downtime reason, and click **OK**. The selected reason is assigned to the unit and appears in the **Cause Reason Tree** column.

9. Click **Assign Action Tree** to assign an action for the downtime reason. The **Tree Selection** appears.
10. Select an action and click **OK**. The selected action is assigned to the unit and appears in the **Action Reason Tree** column.
11. Navigate to the **Reason Tree Configuration** > **Fault Translation Table** tab, and add downtime faults.

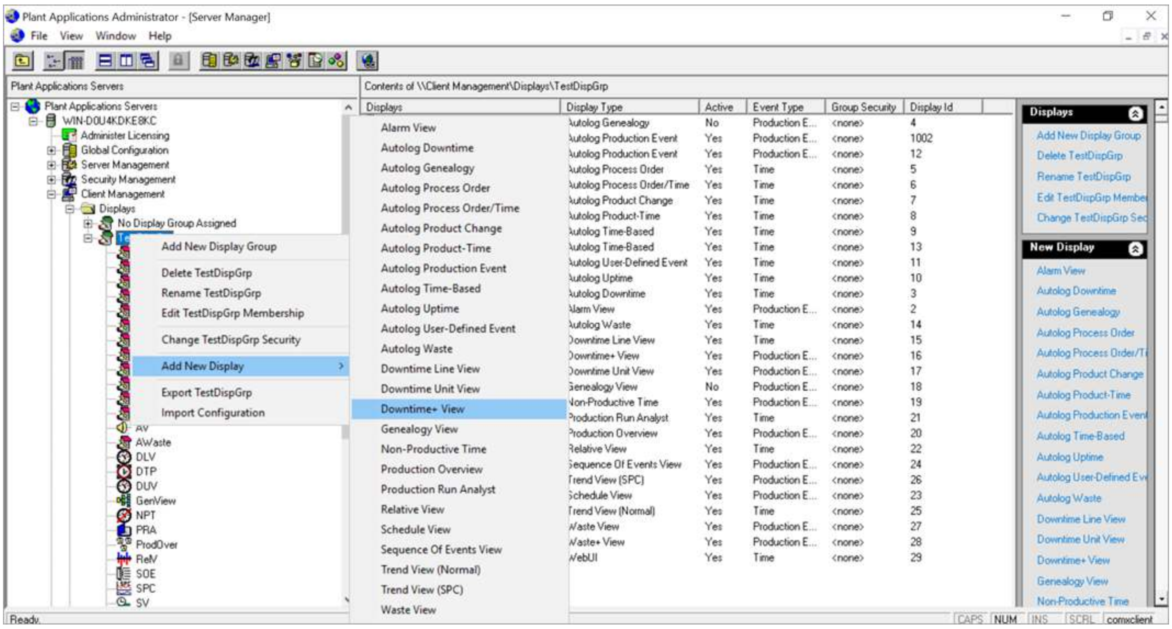


Fault Value	Fault Name	Location	Reason1	Reason2	Reason3	Rea
FaultValue01	FaultName1	NU02	Machine Failures	Maintenance Assist		

12. Click to save the configuration.
13. Click **Yes** when asked if you want to activate the model.
14. Select **OK** when asked to **Reload Service(s)**.

Create Displays

1. Under the Client Management node, right-click **Displays** and click **Add New Display Group**.
2. In the blank entry text box that appears, provide a unique name for the group, and press ENTER.
3. Right-click the newly created display group, and navigate to **Add New Display** > **Downtime+ View**.



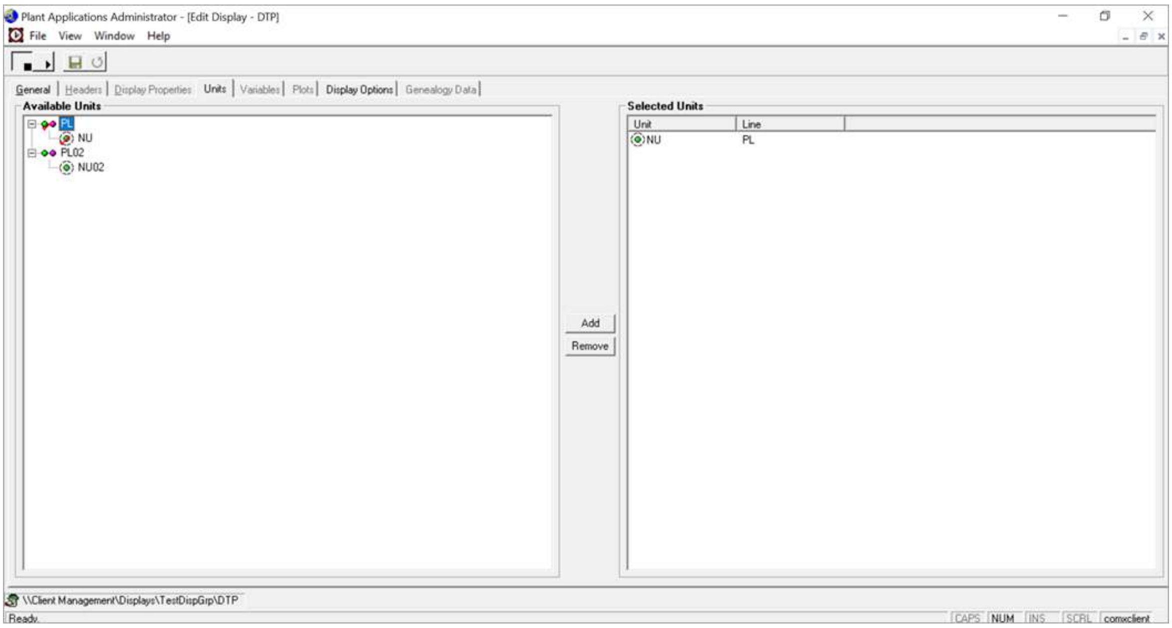
Contents of \\Client Management\Displays\TestDispGrp

Displays	Display Type	Active	Event Type	Group Security	Display Id
	\utolog Genealogy	No	Production E...	<none>	4
	\utolog Production Event	Yes	Production E...	<none>	1002
	\utolog Production Event	Yes	Production E...	<none>	12
	\utolog Process Order	Yes	Time	<none>	5
	\utolog Process Order/Time	Yes	Time	<none>	6
	\utolog Product Change	Yes	Time	<none>	7
	\utolog Product-Time	Yes	Time	<none>	8
	\utolog Time-Based	Yes	Time	<none>	9
	\utolog Time-Based	Yes	Time	<none>	13
	\utolog User-Defined Event	Yes	Time	<none>	11
	\utolog Uptime	Yes	Time	<none>	10
	\utolog Downtime	Yes	Time	<none>	3
	\larm View	Yes	Production E...	<none>	2
	\utolog Waste	Yes	Time	<none>	14
	)owntime Line View	Yes	Time	<none>	15
	)owntime+ View	Yes	Production E...	<none>	16
	)owntime Unit View	Yes	Production E...	<none>	17
	3enealogy View	No	Production E...	<none>	18
	\on-Productive Time	Yes	Production E...	<none>	19
	?roduction Run Analyst	Yes	Time	<none>	21
	?roduction Overview	Yes	Production E...	<none>	20
	?elative View	Yes	Time	<none>	22
	;equence Of Events View	Yes	Production E...	<none>	24
	Trend View (SPC)	Yes	Production E...	<none>	26
	;chedule View	Yes	Production E...	<none>	23
	Trend View (Normal)	Yes	Time	<none>	25
	Vaste View	Yes	Production E...	<none>	27
	Vaste+ View	Yes	Production E...	<none>	28
	VebUI	Yes	Time	<none>	29

4. In the blank entry text box that appears, provide a unique name for the display, and press ENTER.

5. Right-click the newly created display, and select **Edit <*display name*>.** The **Edit Display** screen appears.

6. On the Units tab, select and add the production unit created in Step 3 from the list of **Available Units**.



Unit	Line
NU	PL

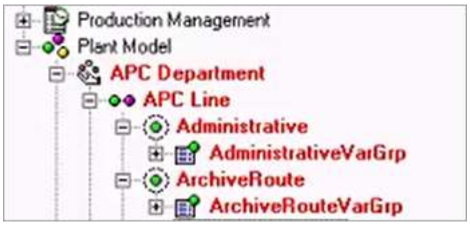
\\Client Management\Displays\TestDispGrp\DTP

7. Click 💾 to save the display configuration.

8. Click **Yes** when asked if you want to activate the display.

Configure the Administrator to Work with the Approval Cockpit Application

1. Log in to the Plant Applications Administrator - [Server Manager].
2. In the **Plant Applications Servers** node, select .
3. Under the Plant Model node, add a department and production line for the Approval Cockpit application.



4. Right-click the Approval Cockpit production line (APC Line), and create production units. The production units are identified as Approval Cockpit forms in the Web Client.
5. Add variable groups for all production units (forms).
6. Under variable groups, create variables (questions) related to Approval Cockpit action events. These variables serve as an approval questionnaire in Approval Cockpit.
7. To configure variables, right-click each production unit (form) and navigate to **Variable Sheet** > **Edit All Variables on <***form name***>**. The **Variable Sheet** screen appears.
8. For detailed information on the **Variable Sheet** fields, see Variable Property Definitions.

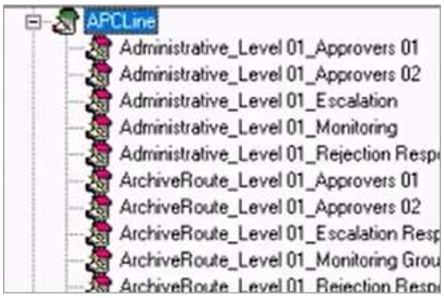
Field	Description
Data Type	Determines what type of data can be entered against each variable question.
Default Test Frequency (Event Based)	If the value is set to 0, the question is not mandatory. If the value is set to 1, the question is mandatory.

9. Save the configuration, and close the **Variable Sheet**.

NOTE: *You must now return to Approval Cockpit, and configure workflows prior to creating display sheets. Refer to Plant Applications Web Client Help.*

Create Display Sheets and Link Variables

1. Under the Client Management node, create Autolog User-Defined Event display sheets with the naming convention: <*ProductionUnitName_WorkflowLevel_WorkflowGroup*>. These displays are based on the configured workflow levels in the Web Client. You can also import the display configuration from an Excel file. See Importing Display Configuration.



APCLine
- Administrative_Level 01_Approvers 01
- Administrative_Level 01_Approvers 02
- Administrative_Level 01_Escalation
- Administrative_Level 01_Monitoring
- Administrative_Level 01_Rejection Resp
- ArchiveRoute_Level 01_Approvers 01
- ArchiveRoute_Level 01_Approvers 02
- ArchiveRoute_Level 01_Escalation Resp
- ArchiveRoute_Level 01_Monitoring Grou
- ArchiveRoute_Level 01_Rejection Resp

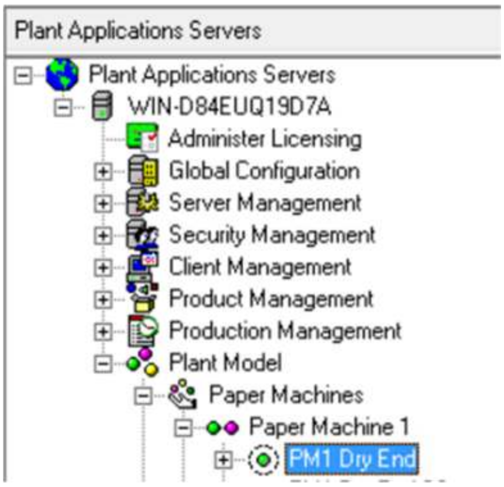
2. Double-click each display sheet to modify the display and link variables. The **Edit Display** screen appears.
3. Visit these tabs to provide values:

Tab	Field	Description
General	Windows Settings (Hrs)	Select the default time window of the display by updating the Initial and Max window hours.
	Master Unit	Select the Production Unit (form) from the list. The variables under this unit are linked to the display.
	SubType	Select the event type added for Approval Cockpit application from the list.
Variables	Add	Select to search for variables in the **Search Variable** screen, and add them for the display. Refer to the topic on *Search Variable dialog box* in this guide.
Display Options	CreateActivities	By default, the value is **False**. Select ... in the **Edit** column to modify the value to **True**.
	Auto Complete	

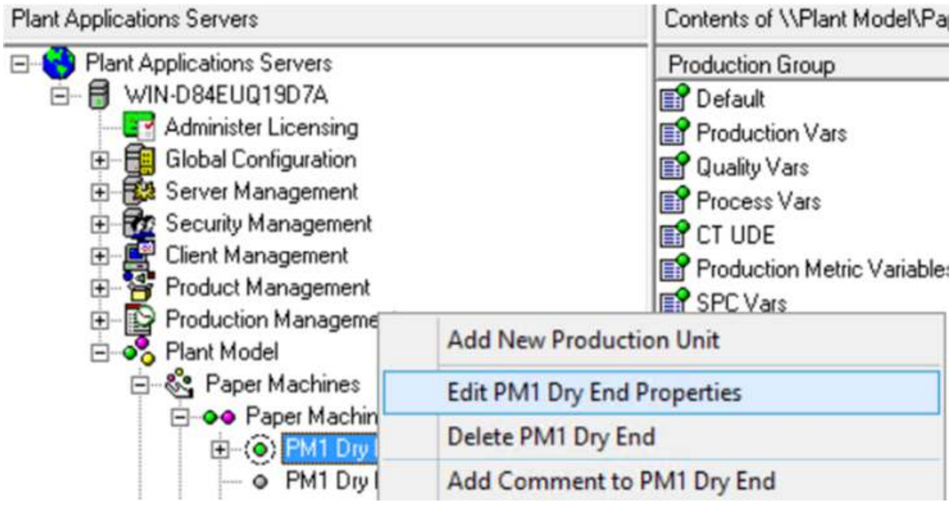
4. Save the configuration, and close the window.

Configure the Administrator for a Time-Based Unit Calculation in the Web Client

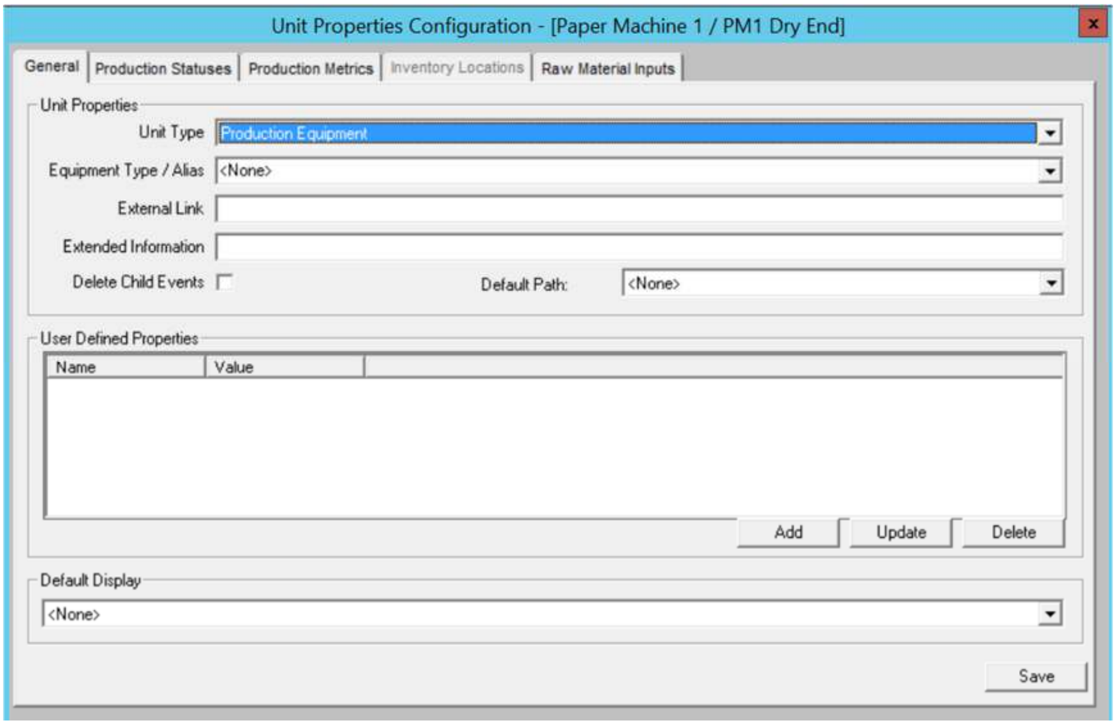
1. Log in to the Plant Applications Administrator - [Server Manager].
2. In the **Plant Applications Servers** node, select ⊞.
3. For the server that includes your plant model associated with the time-based unit, select ⊞.
4. In the **Plant Model** node, select ⊞.
5. Select the department, then the production line, and then the production unit associated with the NPT event.
For example, in the following image, Paper Machines is the selected department, Paper Machine 1 is the selected production line, and PM1 Dry End is the selected production unit.



6. Right-click the selected production unit, and then select **Edit <production_unit_name> Properties**.
For example, as shown in the following the option name is Edit PM1 Dry End Properties.



The **Unit Properties Configuration** window appears as shown in the following image.



Unit Properties Configuration - [Paper Machine 1 / PM1 Dry End]

Unit Properties

Unit Type: Production Equipment

Equipment Type / Alias: <None>

External Link:

Extended Information:

Delete Child Events □

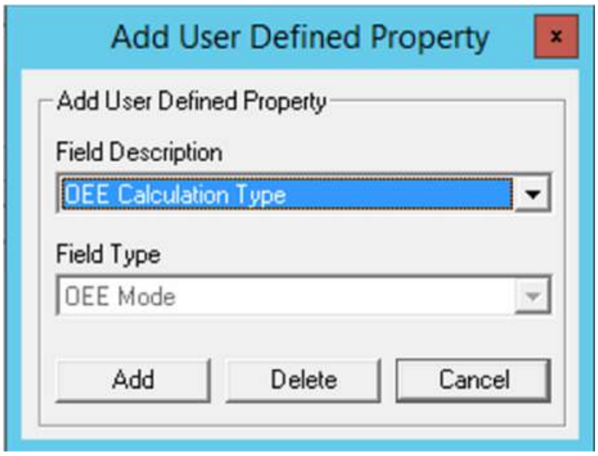
Default Path: <None>

User Defined Properties

Name	Value

Default Display: <None>

7. In the **User Defined Properties** section, select **Add**.
   The **Add User Defined Property** window appears.



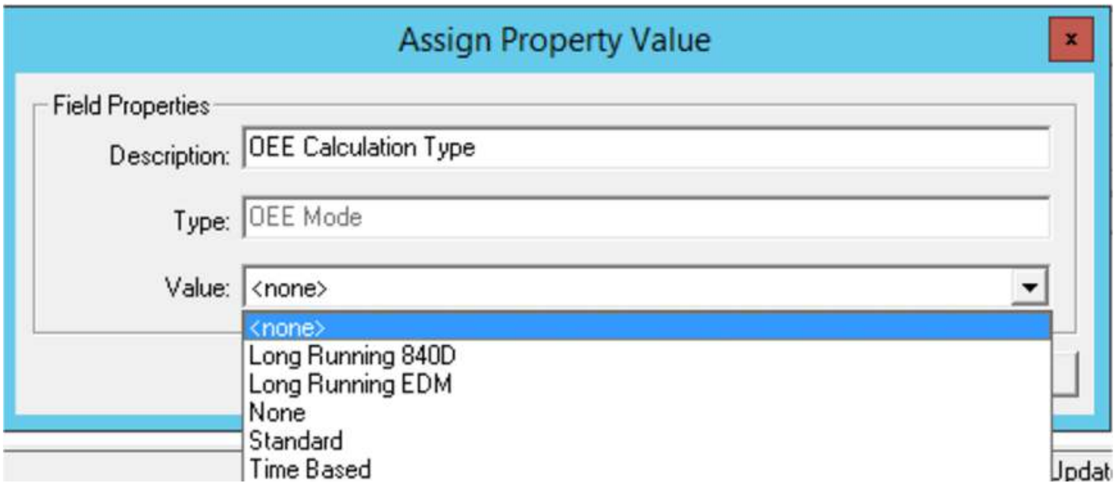
Add User Defined Property

Add User Defined Property

Field Description: OEE Calculation Type

Field Type: OEE Mode

8. In the **Field Description** list, select **OEE Calculation Type**.
9. Select **Add**.
   A confirmation message window appears.
10. Select **OK** in the message window.
11. In the User Defined Properties section, select OEE Calculation Type, and then select **Update**.
    The **Assign Property Value** window appears.



Assign Property Value

Field Properties

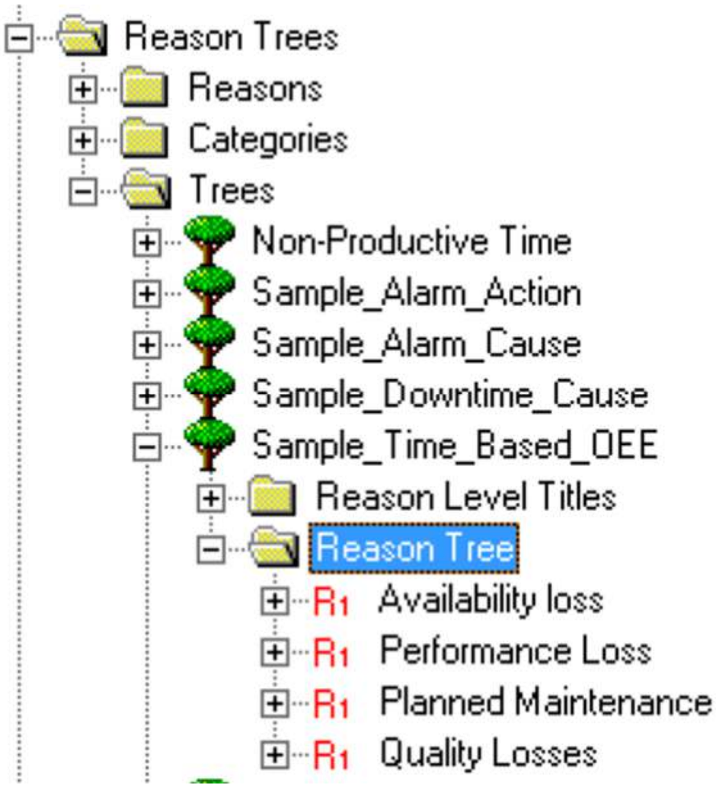
Description: OEE Calculation Type

Type: OEE Mode

Value: <none>

12. In the **Value** list, select **Time Based**.
13. Select **OK**.
14. In the **Unit Properties Configuration** window, select **Save**.
    The configuration changes are saved.

After you configure the administrator, you must create a reason tree in the **Global Configuration** section in the administrator and associate the tree with the downtime event for the time-based unit OEE calculation. The reason tree associated with the downtime event has a structure whose R1 reasons are the same as those defined in the Sample_Time_Based_OEE tree as shown in the following image.



About Enabling a Production Line for Using a Route

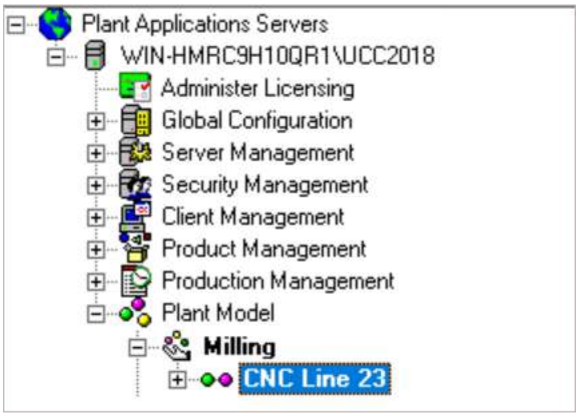
A Production Line is a collection of equipment and Production Units that operate together to produce a product. The Plant Applications Universal Client displays work orders within a Production Unit. To execute a work order on a Production Unit and manufacture products on the Production Line from the Plant Applications Universal Client, you must enable a Production Line to use routes.

Enable a Production Line to Use Routes to Manufacture Products

1. Log in to the Plant Applications Administrator - [Server Manager].
2. In the **Plant Applications Servers** node, select ⊞.
3. In the **Plant Model** node, select⊞ .

   All the Departments for the selected Plant Model appear.
4. In the **<Department_Name>** node, select ⊞, where **<Department_Name>** is the department that includes the Production Line on which you want to enable the route.

   All the lines for the selected Department appear.



5. Right-click the required Production Line, and then select **Route Enable <Production_Line>.**

Note: To manufacture a product on a route-enabled line, you must map the product to all the units along with the virtual unit in the Production Line. For more information, you can refer to the Mapping Products to Units topic.

The selected Production Line is enabled to execute work orders from the Plant Applications Universal Client. By enabling a route on a Production Line, you can configure the required models for each unit, create a Display View, and make the line available for the work order execution. For example, you can create a virtual unit and path for a line.

If there are multiple virtual units or paths configured in a Production Line, an error message appears as described in the following table.

Error Message Name	Error Condition	Corrective Actions
There is more than 1 virtual unit on the line. Please check the configuration and retry	There is more than one virtual unit on the Production Line.	Ensure that you have configured only one virtual unit in the Production Line.

Production event model on virtual unit is other than 118 model. Please check the configuration and retry.	The event model associated with the production event for that Production Line on the virtual unit is not a model 118.	Ensure that the virtual unit on the Production Line contains the production event for the Model 118. You can do one of the following tasks before you run the route-enabled Production Line: • Add a production event for the Model 118 to the virtual unit. • Delete production events for plant models other than Model 118 if added to the virtual unit.
There is more than 1 path configured on the line that contains the virtual unit. Please check the configuration and retry.	Multiple paths are configured on the Production Line that contains the virtual unit.	Ensure that only one path is configured for the Production Line that contains the virtual unit.
There is a path that contains more than 1 virtual units on the path. Please check the configuration and retry.	The path associated with the Production Line contains multiple virtual units.	Ensure that only one virtual unit is configured on the path in the Production Line.
Unit(s) ['+@FailedUnits+' has/have a production(event) model <> 118. It cannot be used for Route configuration.	The units associated with a Production Line does not contain the production event for the Model 118.	Ensure that the virtual unit on the Production Line contains the production event for the Model 118. You can do one of the following tasks before you run the route-enabled Production Line: • Add a production event for the Model 118 to the virtual unit. • Delete production events for plant models other than Model 118 if added to the virtual unit.

Validate a Route-Enabled Production Line

You can verify whether the correct values are specified for each Production Unit and Production Line. You can further validate whether a Production Line is route-enabled by using the Unit Properties and Execution Paths values associated with the Production Line.

Note: To validate the Production Lines, you must have the Plant Applications Administrator role assigned to your user ID.

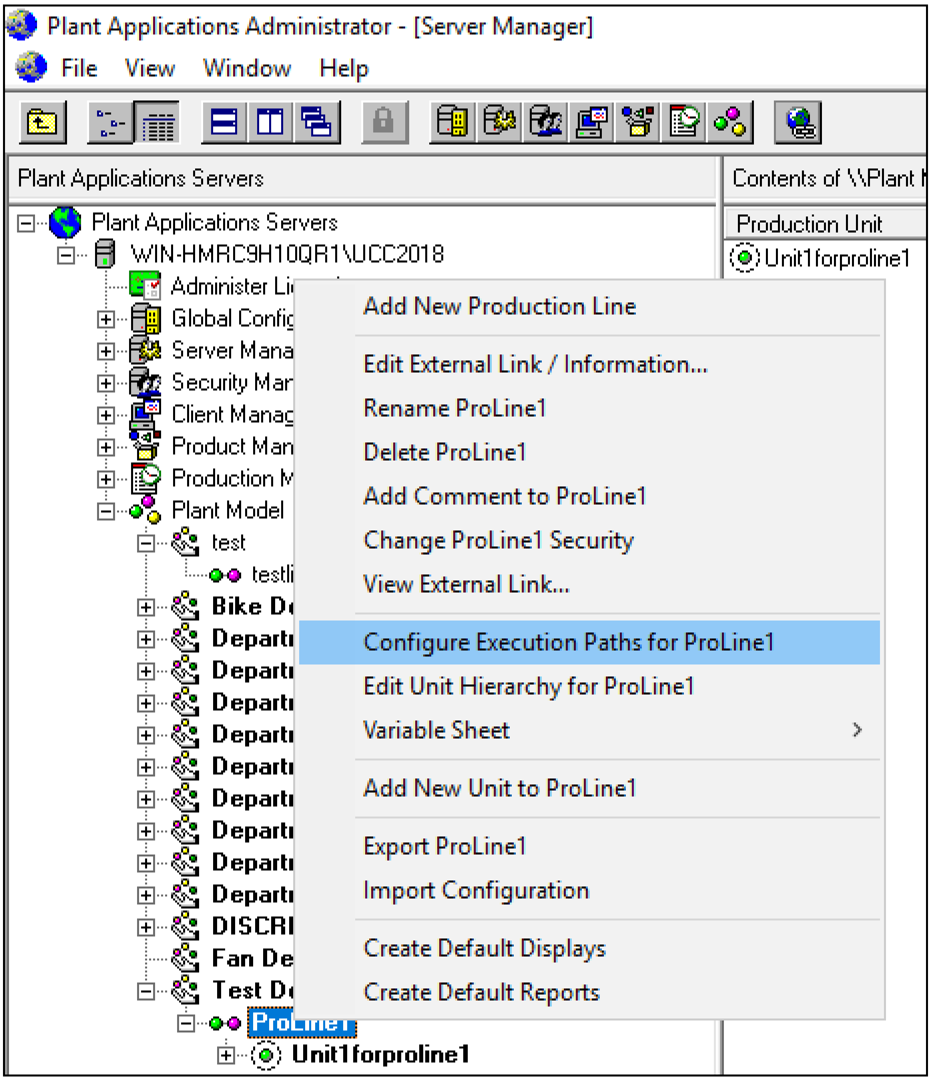
1. Log in to the Plant Applications Administrator - [Server Manager].
2. In the **Plant Applications Servers** node, select .
3. In the **Plant Model** node, select .
4. All the Departments for the selected Plant Model appear.
5. In the **Plant Model**, right-click a virtual Production Unit, and then select **Edit <Production Department: Production Line>: Properties**.
6. In the **<Unit Properties Configuration – [Production Unit / Production Line]>** window, ensure that the value of the Extended Information field is BATCH.

7. Right-click the unit and select **Configure Events on <Production Unit: Production Line>.** Ensure that the following properties are configured for the selected Production Unit:
   - The production event for the Model 118 is included in the virtual Production Unit.
   - The **Area**, **Cell**, and **Unit** properties for the virtual Production Unit are configured.
   - The Waste event for the Model 304 is included in the virtual Production Unit.
8. Optional: To verify whether the Model 49000 is active, run the following query on the service-oriented architecture (SOA) database, and then enter the value in the <ED_Model_Id> field as 49000:

   ```
   Select * in the <event_configuration> window in the SOA database.
   ```

The record associated with the model 49000 appears and the value in the <is_active> field is 1.

9. Right-click the Production Line, and then select **Configure Execution Paths for <production_line>**.



The **Execution Path Configuration for <Production_Line>** window appears.

10. In the Execution Path Configuration for <Production_Line> window, ensure that the Production and Schedule boxes are set to False.
11. Select the **Path Units** tab, and then select the virtual unit in the **Execution Path Units** section.

12. Ensure that you have configured the following properties for the virtual unit:
    - The <is_production_point> production point value is set to True**.**
    - The <is_schedule_point> schedule point value is set to True**.**
    - The <unitorder> unit order value is set to 1.
    - The Valid Transition property for a pending Path Item contains the Planning and Next transition options.
    - For an order in the Active status, the value of the How Many property is set to 1000.
    - On successful execution, the <IsRouteEnable> UDP is set to True.

    Note: If there is more than one virtual unit on a Production Line, you cannot enable the Production Line using a route. In this case, you must manually select the correct virtual unit and remove the other virtual units.

13. In the **Client Management** node, select .
14. In the **Display** node, select , and then select the display group.
    The Schedule View for the route-enabled path appears.

    Note: If the Schedule View does not exist, configure the View according to the required configuration, where the display path is the path in the line that contains a single virtual unit, and then set the security group for the View to Administrator. For more information, refer to the Creating and Configuring a Scheduled View Display topic.

Result Sets

For detailed information on result sets, refer to *Plant Applications Result Sets* help file.

Result Sets Overview

Often, we need to use custom SQL stored procedures to add or modify data in the Plant Applications database. When this need arises, we **do not** directly use Insert, Update, or Delete statements within the stored procedure. Instead, we rely on result sets.

Result sets allow us to use the Plant Applications messaging system to safely update the Plant Applications database and notify the Plant Applications clients of the changes made.

Result Sets and the Messaging System

There are several different types of result sets and each has its own distinct format. Within the custom stored procedure, a result set is simply a **Select** statement followed by a list of parameters. The number and type of parameters depend on the result set type.

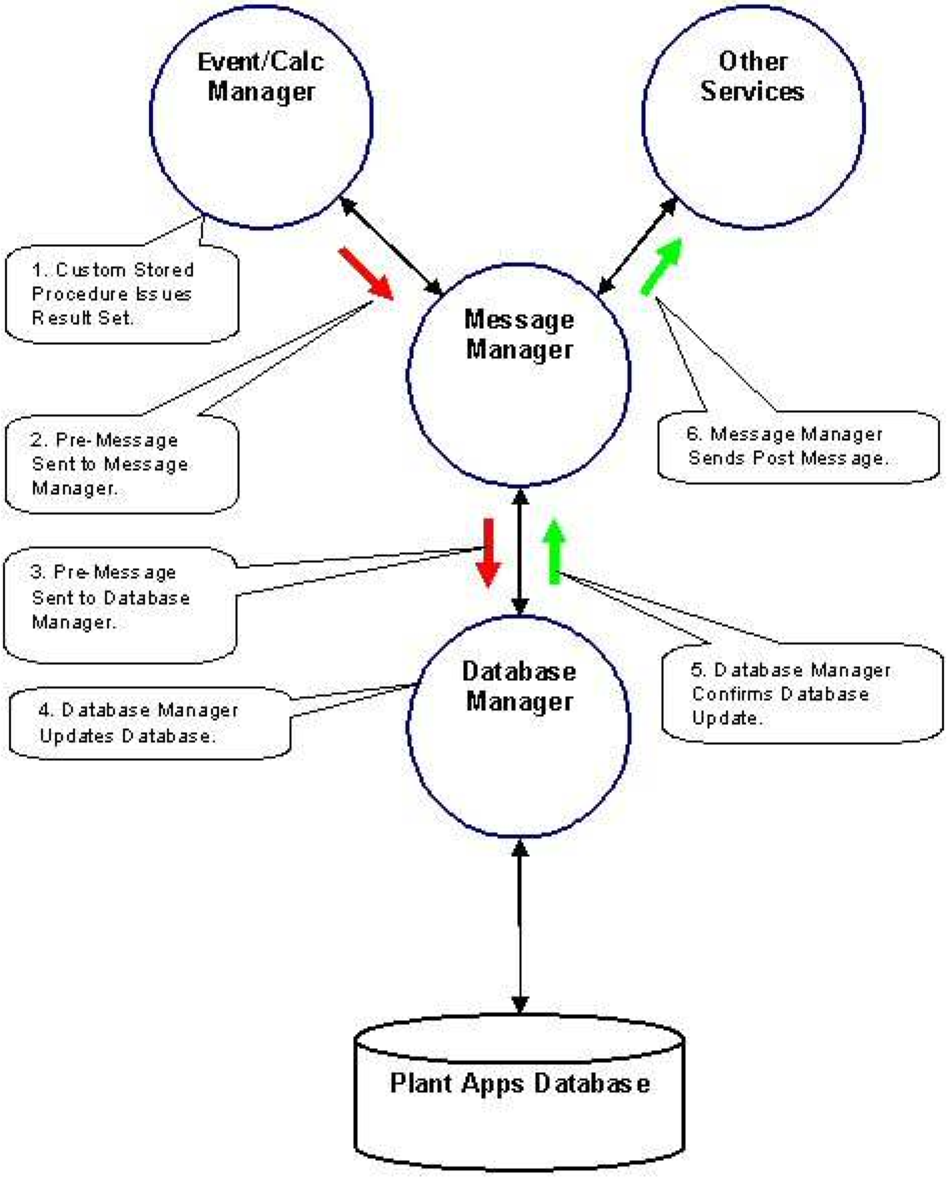
The first value after the **Select** statement must be an integer which is an identifier indicating which type of result set (message) is being issued.

Result sets are not issued to the messaging system until the execution of the custom stored procedure is complete. Also, result sets do not have output parameters.

With respect to the Router service (this was called the Message Bus in earlier releases) and result sets, there are two message types. These are a **pre-message** and a **post-message**. A **pre-message** is an indication for the Event Manager to request the Database Manager to update the database while a **post-message** indicates that the database has been updated and the other services must be notified of the update.

Most result sets will be used to modify the Plant Applications database. For the result sets that do update the database, you will need to set the value of the pre/post parameter accordingly.

The following diagram shows how a result set message is processed.


Event/Calc Manager
Other Services
1. Custom Stored Procedure Issues Result Set.
2. Pre-Message Sent to Message Manager.
Message Manager
6. Message Manager Sends Post Message.
3. Pre-Message Sent to Database Manager.
4. Database Manager Updates Database.
Database Manager
5. Database Manager Confirms Database Update.
Plant Apps Database

Maintenance and Troubleshooting

For information on data purge, refer to *Plant Applications Data Purge Utility* help file.

For information on maintenance, refer to *Plant Applications Maintenance* help file.

Import/Export Tool

You can import or export Plant Applications configurations, such as users, events, lines, and variables. For example, you can export site users from one server and import them into Plant Applications on a different server. The import/export tool can save you from having to manually enter similar objects onto a different server. Another way to use the import/export tool is to create templates. For example, you can export variables, change the variable data in the worksheet, and then import the changed variable data back into Plant Applications.

You must have Administrator access to the Administrator security group to import or export objects.

Exporting Configurations

When you select a configuration to export, Plant Applications automatically launches Excel and creates the necessary worksheets within the Excel workbook. After the workbook has been created, you can edit the configuration information within the workbook. Then, you can import the data into Plant Applications either on the same server or on a different server.

You must have **Administrator access** to the **Administrator security group** to export configurations.

Before you begin the export, you must close all instances of Excel, unless you want the information added to the open Excel workbook.

To export objects:

1. Log in to the Plant Applications Administrator program.
2. Right-click on what you want to export and select **Export <*object name*>**. The **Configuration Export** dialog box appears
3. Select what you want to export and click **OK**.

NOTE: *The options in the Export Configuration dialog box will vary, depending on the configuration you are exporting.*

4. Excel will start and your workbook will be created.
5. To export additional configurations, repeat steps 2 and 3.

Importing Configurations

You can import your Plant Applications configurations, either onto the same server or onto a different server. Importing can help minimize or eliminate tedious data entry of information that may already exist in Plant Applications on another server. Additionally, you can create and export templates, such as variables, edit the worksheets and then import the edited configuration into Plant Applications.

NOTE: *When importing multiple user-define events from Excel, an error occurs when UDEs have the same description as exported from the Plant Applications Administrator. Refer to Changing the Description to Import Multiple UDEs from Excel.*

If your workbook contains multiple worksheets, Plant Applications will import everything that you've selected on each worksheet, in the correct order.

If your configurations are contained in separate workbooks, you must import the configurations is the correct order. See Import Order for more information.

Keep in mind that you can change the information in your worksheet, and the changed information will be imported.

You must have **Administrator access** to the **Administrator security group** to import or export objects.

To import configurations:

1. Open the Excel workbook that contains the exported configurations.
2. In the workbook, under the **Selected** column, place an **X** in the cell that corresponds to what you want to import.
3. Log in to the Plant Applications Administrator program.
4. Right-click on what you want to import and select **Import Configuration**.

Keep in mind that everything that you've selected on your workbook will be imported, regardless of where you've clicked. For example, if you select Import Configuration from Administer Alarms and have variables selected in your workbook, then variables (and everything else that is selected) will be imported.

5. Click **OK** to complete the import.
6. Refresh the server.

If the import fails, switch to your workbook. Under **Returned Messages**, a reason for the failed import is displayed.

Import Order

If your Plant Applications configurations are in the same workbook, your configurations will be imported in the correct order. If your Plant Applications configurations are in separate workbooks, they must be imported in the correct order. This does not mean that all configurations have to be imported; it simply means that certain configurations must be imported before other configurations. For example, if you are importing a production line, and you have security assigned to the production line, you must import them in the following order:

1. The users who are members of the security group assigned to the production line
2. The security group assigned to the production line
3. The production line

If you try to import the production line without the other information, the import will fail. If you don't want to import the security groups and site users, delete the information in the **Security User Groups** column before you import.

IMPORTANT: *Keep in mind that if you import the changed information back into the same Plant Applications server, the production lines will no longer have a security group assigned to them.*

Import Order

- Data Types
- Event Reasons
- Production Status
- Events

- Event Subtypes
- Users
- Security Groups
- Production Lines
- Production Units
- Production Groups
- Variables
- Variable Calculations
- Product Family
- Product Groups
- Products To Units
- Var Specs
- Product Properties
- Characteristics
- Characteristic Groups
- Spec Variables
- Central Specs
- Product Characteristics
- Reason Trees
- Reason Category
- Crew Schedule
- Alarm Templates
- Alarm Template Data
- Displays
- Display Variables
- Display Options
- Unit Locations
- Schedule Statuses
- Process Order
- Process Order Sequence
- Process Order Pattern
- Event Reason Tree
- Timed Event Fault
- Waste Event Fault

What Can Be Imported or Exported?

You can import or export most of your Plant Applications configurations.

You must have **Administrator access** to the **Administrator security group** to import or export configurations.

Global Configuration	**Product Management**
Schedule Statuses	Product Families
Data Types	Product family
Individual data source types	Product Properties
Reasons	Product property
Trees	Product Groups
Subscription Groups	Product group
Subscriptions	**Production Management**
Engineering Units	Customers
Engineering Conversions	Customer Orders
Production Statuses	Customer Shipments
Alarms	Production Schedules
Events	Crew Schedule
Language Prompts	Non-Productive Time
Server Management	**Plant Model**
E-Mail Engine Configurations	Plant Model
Historian Connections	Departments
Security Management	Production Lines
Site users	Production Units
Security Groups	Variable Groups
Security Roles	
Client Management	
Displays	
Display Groups	
Individual displays (export only)	

MESCore Service Provider Documentation

MESCore Service Provider Interface Methods Help provides API-level information about methods for third-party developers and application engineers. The help is available in the Documentation folder that is linked from the Installation Menu and is also available from the Start menu after installing Plant Application Clients.

.NET SDK

Plant Applications SDKs

The Plant Applications SDK is licensed separately and by concurrent users for creating custom displays and interfaces. A legacy SDK supports VB6 or C++ applications. Plant Applications 5.0 and 6.0 supported a new .NET SDK for C# or VB.NET applications. Plant Applications 6.1 adds a COM wrapper to use with the .NET SDK to continue support for the legacy VB6 or C++ applications written for the legacy SDK.

The .NET SDK is installed from the Plant Applications Installation Menu as described in the *Getting Started Guide*. The COM wrapper is installed with the .NET SDK.

> ***Note:*** *The COM wrapper does not support BigInt (long integers) in the Tests table because Visual Basic and COM do not support 64-bit.*

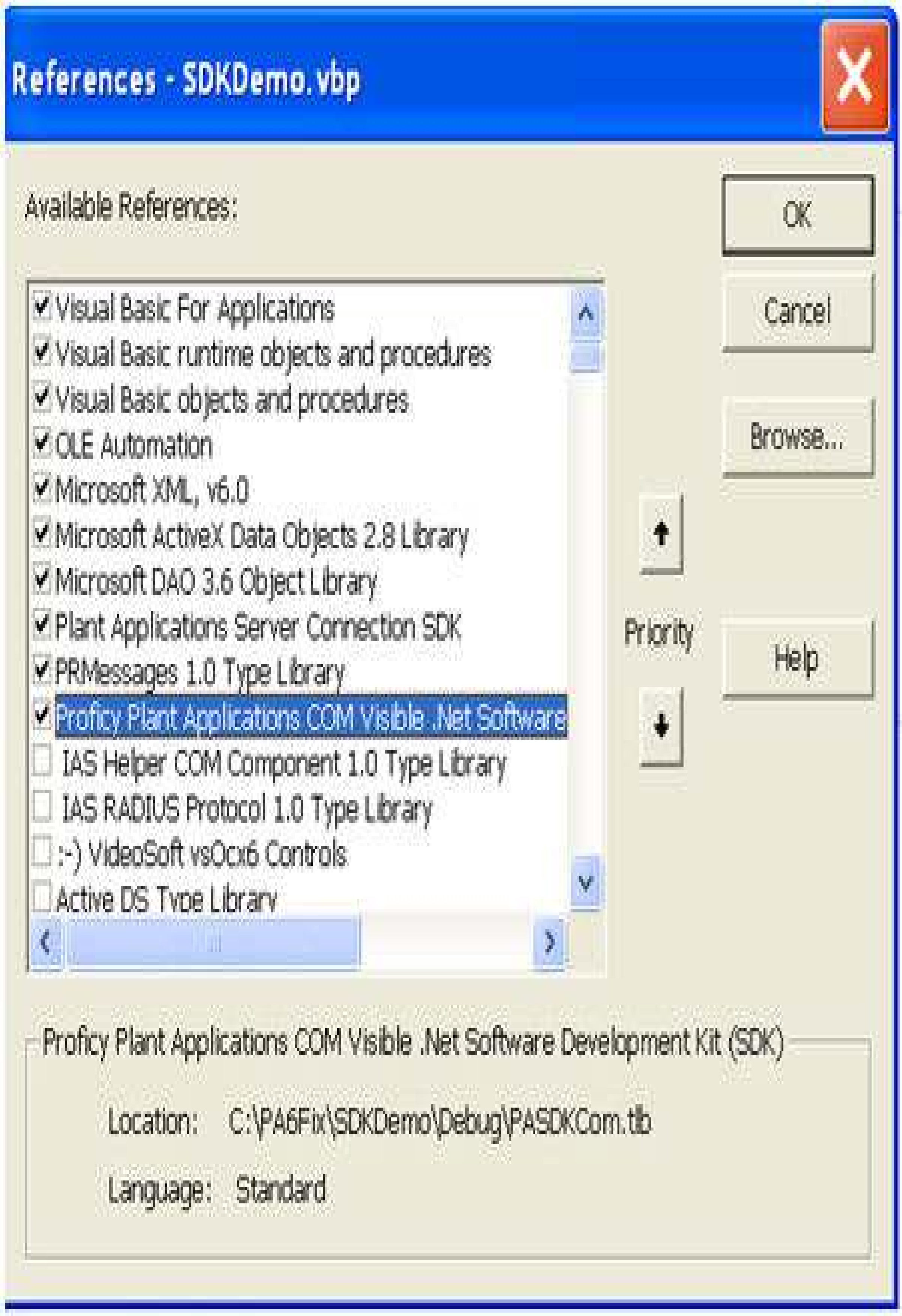
Help for the SDKs is installed with the SDK. The following examples are provided here:

- Using the COM wrapper in the .NET SDK with legacy VB6 or C++ applications
- Using the .NET SDK with C# or VB.NET applications

Using the .NET SDK COM Wrapper with Legacy C++ and VB6 Applications

The examples given below describe using the Plant Applications DCOM Server to connect to latest Plant Applications SDK, which is the .NET SDK. The intended audience is for VB6 users who want to use the latest SDK with existing applications written for the .COM SDK.

Before using the COM Wrapper, add the following reference:



References - SDKDemo.vbp

Available References:

- ☑ Visual Basic For Applications
- ☑ Visual Basic runtime objects and procedures
- ☑ Visual Basic objects and procedures
- ☑ OLE Automation
- ☑ Microsoft XML, v6.0
- ☑ Microsoft ActiveX Data Objects 2.8 Library
- ☑ Microsoft DAO 3.6 Object Library
- ☑ Plant Applications Server Connection SDK
- ☑ PRMessages 1.0 Type Library
- ☑ Proficy Plant Applications COM Visible .Net Software
- □ IAS Helper COM Component 1.0 Type Library
- □ IAS RADIUS Protocol 1.0 Type Library
- □ :-) VideoSoft vsOcx6 Controls
- □ Active DS Type Library

Proficy Plant Applications COM Visible .Net Software Development Kit (SDK)

Location: C:\PA6Fix\SDKDemo\Debug\PASDKCom.tlb

Language: Standard

Connect to DCOM Server

```
//The code shows connecting to a DCOM server.  If the DCOM is remote and a DCOM security setting is involved, the user needs to establish the proper settings.

//Input Parameter:

//ServerHost: DCOM server host machine name, can be IP, data type: string

//ClientAppName:  Application name, data type: string

//EnableDetailedLog: Log detailed information or not, data type: Boolean

//No return value

Dim PAComSvr As PACOMServer

Dim Token as string

Dim ConnectionStatus as boolean

PAComSvr.Init  ServerHost, ClientAppName, EnableDetailedLog

Token = PAComSvr.Login_2(User,UserPassword)

If Token <> "" Then

        lblLoading.Caption = "Successfully connect to Proficy DCOM Server … "

        ConnectionStatus = True

        …Do further steps

Else

        ConnectionStatus = False

        MsgBox "Failed to get current user token."

End If
```

Disconnect to DCOM Server

```
// This subroutine shows how to disconnect to a DCOM Server.  The DCOM Server should be disconnected before exiting the program.

    PAComSvr.Logout

   ConnectionStatus = False
```

Query SDK Objects Collection

```
//The following code illustrates how to query all production lines.

//Input Parameters:

//Token: Returned value of function: PAComSvr.Login_2

// RowCountLimit: returned object maximum number

//Return values: Collection of object: PACOMProductionLine

Dim paLineQry As PACOMProductionLineQuery

Set paLineQry = New PACOMProductionLineQuery

paLineQry.Init PAComSvr
```

```
Dim allLines() As PACOMProductionLine

allLines = paLineQry.RunQuery(Token, RowCountLimit)

If UBound(allLines) < 0 Then

     MsgBox "No production line find "

Else

     For Each Line In allLines

         ShowLine(Line.Text, Line.Id)

     Next

End If
```

Use Filter to Query an SDK Object from Objects Collection

```
//The following code illustrates how to get all units under a particular line.

//Input parameters:

// PACOMProductionUnitPropertyEnum_ProductionLine :Object property enum

// PACOMQueryOperator_Equals: PAComQueryOperator

//LineName: Selected Production Line Name

Dim paUnitQry As PACOMProductionUnitQuery

Set paUnitQry = New PACOMProductionUnitQuery

paUnitQry.Init PAComSvr

paUnitQry.AddFilter_2 PACOMProductionUnitPropertyEnum_ProductionLine,
PACOMQueryOperator_Equals, LineName

//The operator can use like, GreaterThan, LessThan, Like etc. for example:
//PACOMQueryOperator_Like, "%o%"

// PACOMQueryOperator_Like, "???%"

// PACOMQueryOperator_Like, "%s"

// PACOMQueryOperator_LessThan, EndTime

Dim allUnits() As PACOMProductionUnit

allUnits = paUnitQry.RunQuery_2(token)

If UBound(allUnits) < 0 Then

       MsgBox "No production units find."

Else

   For Each unit In allUnits

       ShowUnit(unit.Text, unit.Id)

   Next

End If
```

Add an SDK Object

```
//The following code illustrates how to add a variable.
//Input Parameters:
//Token: Returned value of function: PAComSvr.Login_2
// tranType: TransType enum, 1 Add;2 Update; 3 Delete
//Return values: object of PACOMQueryReport
Dim tranType As PACOMNullableInt32
Set tranType = New PACOMNullableInt32
tranType.Value = 1
Dim Rpt As PACOMQueryReport
Set paVarEvent = New PACOMVariableResultEvent
Dim VarName As String
paVarEvent.SetServer PAComSvr
paVarEvent.ResultOn = Now
paVarEvent.Value = “abc”
paVarEvent.TransactionType = tranType
paVarEvent.CommentText = “comment …”
paVarEvent.TransNum = 0
paVarEvent.WriteDirect = true // false:go through message server
Set Rpt = paVarEvent.Add_2(token)
If Rpt.NoProblems Then
   MsgBox "VRE Add - [SUCCESS]"
Else
   MsgBox "VRE Add - [FAIL] " + Rpt.GetReportErrorMsg
End If
```

Delete an SDK Object

```
//The following code illustrates how to delete variables.
//Input Parameters:
//Token: Returned value of function: PAComSvr.Login_2
// tranType: TransType enum, 1 Add;2 Update; 3 Delete
//Query result return values: object collection of PACOMVariableResultEvent
//Delete result return result: object of PACOMQueryReport
 Dim tranType As PACOMNullableInt32
 Dim Rpt As PACOMQueryReport
 Set tranType = New PACOMNullableInt32
 tranType.Value = 3
 Dim paVarResult() As PACOMVariableResultEvent
 Dim VarId As Integer
 Dim VarName As String
 Dim paVarResultQry As PACOMVariableResultEventQuery
 Set paVarResultQry = New PACOMVariableResultEventQuery
 paVarResultQry.Init PAComSvr
     paVarResultQry.AddFilter_3 PACOMVariableResultEventPropertyEnum_Variable,
     PACOMQueryOperator_Like, %VarName%
 paVarResult = paVarResultQry.RunQuery_2(Token)
 If UBound(paVarResult) < 0 Then
   MsgBox "No variable result record find "
 Else
  For Each var1 In paVarResult
    var1.TransactionType = tranType
    Set Rpt = var1.Delete_2(token)
   If Rpt.NoProblems Then
      MsgBox "VRE Delete - [SUCCESS]"
   Else
     MsgBox "VRE Delete - [FAIL] " + Rpt.GetReportErrorMsg
   End If
  Next
End If
```

Update an SDK Object with ESignature

```
//The following code updates a waste event with a newly created ESignature.
Dim WasteEvent As PACOMWasteEvent
Set WasteEvent =new PACOMWasteEvent
Dim ESig As PACOMESignature
Set ESig = New PACOMESignature
ESig.OperatorId = PAComSvr.AuthenticateLogOn(ESigUserName, EsigUserPassword)
ESig.ApproverId = PAComSvr.AuthenticateLogOn(ApproverName, ApproverPassword)
ESig.Add_2 (Token)
If (ESig.Id.HasValue = False) Then
    MsgBox "ESignature could not be created"
    Exit Sub
End If
WasteEvent.ESignatureId = ESig.Id.Value
WasteEvent.Update_2 (Token)
```

Subscribe an Event

```
//The following code subscribes a variable event.
// The parameters to the subscription method depend on the type of message you are
// subscribing to.  Most require the line and the unit. A few also require the variable.
//Message types related to production paths require the path code.
Dim WithEvents DataCallBack As PACOMSubscription
Dim VarResultSub As New PACOMVariableResultEvent
If DataCallBack Is Nothing Then
Set VarResultSub = New PACOMVariableResultEvent
VarResultSub. VariableId = VarId
Set DataCallBack = VarResultSub.Subscribe(PAComSvr)
End If
```

Unsubscribe an Event

```
//The following code unsubscribes a variable event.
//The Input parameter is the object instance returned when call subscribe
If Not VarResultSub Is Nothing Then
   VarResultSub.Unsubscribe (DataCallBack)
   VarResultSub.UnsubscribeAll (PAComSvr)
End If
```

Subscribed Variable Event Fired

```
//The following code handles a variable change event.
//This event handler fires when a new message is received.
//Because you are subscribed to only the Variable Result messages for this variable
//on this line and unit, it could be the one you just sent.
Private Sub DataCallBack_PACOMMessageReceived(ByVal Msg As PASDKCOM.PACOMMessageInterface)
```

```
    Dim Pending As String

    If Not Msg.VariableResultEvent Is Nothing And Msg.MsgType =
PACOMMessageTypes_VariableResultEvent Then

       If (Msg.VariableResultEvent.Pending) Then

         ShowVariableStatus = "Pending[ TRUE] "

       Else

         ShowVariableStatus = "Pending[FALSE] "

       End If

       If Msg.VariableResultEvent.TransactionType.Value = 3 Then

         ShowVariableOperation = "Delete Variable Result:" + CStr(Msg.VariableResultEvent.Variable

         Exit Sub

       End If

       If Msg.VariableResultEvent.TransactionType.Value = 1 Then

         ShowVariableOperation = "Add Variable Result:" + CStr(Msg.VariableResultEvent.Variable)

         ShowVariableDetails="VariableId = " + CStr(Msg.VariableResultEvent.VariableId.Value)

         ShowVariableValue =  Msg.VariableResultEvent.Value

       End If

       If (Msg.VariableResultEvent.Pending) Then

         Exit Sub

       Else

         ShowLineName ="LineName = " + Msg.VariableResultEvent.ProductionLine

        ShowUnitName = "UnitName = " + Msg.VariableResultEvent.ProductionUnit

        ShowVariableName = "VariableName = " + Msg.VariableResultEvent.Variable

        ShowProductCode = "ProductCode = " + Msg.VariableResultEvent.ProductCode

        ShowEventName "EventName = " + Msg.VariableResultEvent.EventName

        ShowVarSpec = "LRL = " + Msg.VariableResultEvent.LRL

        ShowVarSpec == "LWL = " + Msg.VariableResultEvent.LWL

        ShowVarSpec ="LUL = " + Msg.VariableResultEvent.LUL

        ShowVarTarget ="Target = " + Msg.VariableResultEvent.TGT

        ShowVarSpec = "UEL = " + Msg.VariableResultEvent.LEL

        ShowVarLink = "URL = " + Msg.VariableResultEvent.url

        ShowVarSpec = "UWL = " + Msg.VariableResultEvent.LWL

        ShowVarSpec = "UUL = " + Msg.VariableResultEvent.LUL

      End If
```

```
  End If
```

Using the .NET SDK with C# or VB.NET Applications

This topic provides examples for .NET-based users to use the latest Plant Applications .NET SDK. You must first add the reference for the PASDK DLL, which has the following name spaces:

- Proficy.PlantApplications.SDK.Objects
- Proficy.PlantApplications.SDK.Queries
- Proficy.PlantApplications.SDK.Database
- Proficy.PlantApplications.SDK.Messaging
- Proficy.PlantApplications.SDK.Authentication
- Proficy.PlantApplications.SDK.Common

Query SDK Objects Collection

```
//Following codes illustrate how to query production lines

Server TheServer = new Server(System.Environment.MachineName, "AppName");

CString MyToken = TheServer.Login(User,Password);

PAProductionLineQuery line = new PAProductionLineQuery(TheServer);

List<PAProductionLine> lines;

lines = line.RunQuery(MyToken,LineCountLimit);

if(lines!=null)

{

 for(int ii=0;ii<lines.Count;ii++)

  ShowLineProperties(lines[ii].Id,lines[ii].ProductionLine);

}
```

Using Complex Filter to Query SDK Objects

```
//Following code illustrate how to get all units which the name contain character ‘m’

PAProductionUnitQuery unit = new PAProductionUnitQuery(TheServer);

List<PAProductionUnit> units;

unit.AddFilter(PAProductionUnit.Property.ProductionLine, PADataObjectQuery.Operator.Equals, Line);

unit.AddAnd();

unit.AddFilter(PAProductionUnit.Property.ProductionUnit, PADataObjectQuery.Operator.Like, “%m%”;

units = unit.RunQuery(MyToken);

if (units != null)

{

 for(int ii=0;ii<lines.Count;ii++)
```

```
  ShowLineProperties(lines[ii].Id,lines[ii].ProductionLine);
}
```

Add an SDK Object

```
//Following codes illustrate how to add a variable using ESignature
PAVariableResultEvent VarEvent=new PAVariableResultEvent() ;
Authenticator ESig = new Authenticator();
ESig.OperatorName = "comxclient";
ESig.OperatorPassword = "comx";
ESig.ApproverName = "comxclient";
ESig.ApproverPassword = "comx";
VarEvent.ResultOn=DateTime.Now;
VarEvent.Value="abc";
VarEvent.CommentText="This is a test";
VarEvent.WriteDirect=false;
QueryReport Rpt=VarEvent.Add(MyToken,ESig,ESig);
if (!Rpt.NoProblems)
  LogEvent("ERROR Adding VarResultEvent:" Rpt.GetReportErrorMsg());
```

Delete an SDK Object

```
//Following codes will delete a BOM
PABOM BOMInstance = new PABOM(TheServer);
BOMInstance.Id = Id;
BOMInstance.BOM = BOMName;
QueryReport rpt = BOMInstance.Delete(MyToken);
if (rpt.HasErrors)
{
LogEvent("ERROR Deleting a BOM:" Rpt.GetReportErrorMsg());
}
```

Update an SDK Object

```
//Following codes will update a BOM
PABOM BOMInstance = new PABOM(TheServer);
BOMInstance.Id = Id; //for query purpose
BOMInstance.BOM = NewBOMName;
BOMInstance.UpdateProperties.Add(PABOM.Property.BOM);
BOMInstance.BOMFamily = NewBOMFamily;

BOMInstance.UpdateProperties.Add(PABOM.Property.BOMFamily);

BOMInstance.BOMFamilyId = NewBOMFamilyId;

BOMInstance.UpdateProperties.Add(PABOM.Property.BOMFamilyId);

BOMInstance.CommentId NewCommentId;

BOMInstance.UpdateProperties.Add(PABOM.Property.CommentId);

BOMInstance.Update(MyToken)
```

Subscribe an event

```
//Following codes will subscribe a product change event for the specified unit

//declare a delegate

public delegate void MessageReceivedDelegate(IGateway Connection,
MessagingMsgReceivedEventArgs EventArgs);

Subscription ProdChangeSubs = PAProductChangeEvent.Subscribe(TheServer, unitId);

ProdChangeSubs.MessageReceived += new
MessageReceivedDelegate(ProdChangeSubs_MessageReceived);
```

Unsubscribe an event

```
//Following codes will unsubscribe a product change event that subscribed above codes

TheServer.Messaging.Unsubscribe(ProdChangeSubs);
```

Process the incoming message

```
void MessageReceived(IGateway Connection, MessagingMsgReceivedEventArgs EventArgs)

{

if (EventArgs.EventObject.Pending)

return;

if (EventArgs.EventObject.GetType() == typeof(PAVariableResultEvent))

{

PAVariableResultEvent Obj = (PAVariableResultEvent)EventArgs.EventObject;
HandleIncomingVariable((int)Obj.UserId,Convert.ToInt32(Obj.Value));

}

else if (EventArgs.EventObject.GetType() == typeof(TopicMsg))

{

int row, col, val;

TopicMsg Obj = (TopicMsg)(EventArgs.EventObject);

for (row = 0; row < Obj.TopicValue.Rows.Count; row++)

for (col = 0; col < Obj.TopicValue.Columns.Count; col++)
{
if (Obj.TopicValue.Rows[row][col].ToString() == "")
val = 100;
else
val = Convert.ToInt32(Obj.TopicValue.Rows[row][col]);
…
}
}
else if (EventArgs.EventObject.GetType() == typeof(PAProductionEvent))


{
PAProductionEvent Obj = (PAProductionEvent)EventArgs.EventObject;
HandleProductionEvent((int)Obj.UserId, (Obj.AppliedProductId.HasValue ?
(int)Obj.AppliedProductId : (int)0));
}
else if (EventArgs.EventObject.GetType() == typeof(PAWasteEvent))
{
PAWasteEvent Obj = (PAWasteEvent)EventArgs.EventObject;
HandleWasteEvent((int)Obj.UserId,(int)Obj.WasteTypeId);
}
else if (EventArgs.EventObject.GetType() == typeof(PAProductChangeEvent))
{
PAProductChangeEvent Obj = (PAProductChangeEvent)EventArgs.EventObject;
HandleProductChangeEvent((int)Obj.UserId,(int)Obj.ProductId);
}
else if (EventArgs.EventObject.GetType() == typeof(PADowntimeEvent))
{
PADowntimeEvent Obj = (PADowntimeEvent)EventArgs.EventObject;
HandleDownTimeEvent((int)Obj.UserId,(int)Obj.DowntimeFaultId);
}
else if (EventArgs.EventObject.GetType() == typeof(PAUserDefinedEvent))
{
PAUserDefinedEvent Obj = (PAUserDefinedEvent)EventArgs.EventObject;
HandleUserDefinedEvent((int)Obj.UserId,(int)Obj.EventSubTypeId);
}
```

```
else if (EventArgs.EventObject.GetType() == typeof(PAGenealogyEvent))
{
PAGenealogyEvent Obj = (PAGenealogyEvent)EventArgs.EventObject;
HandleGenealogyEvent((int)Obj.UserId,(int)Obj.DimensionX);
}


else if (EventArgs.EventObject.GetType() == typeof(PAPathInputEvent))
{
PAPathInputEvent Obj = (PAPathInputEvent)EventArgs.EventObject; HandlePathInputEvent=
((int)Obj.UserId,(int)Obj.DimensionY);
}
else if (EventArgs.EventObject.GetType() == typeof(PAProductionPlanEvent))
{
PAProductionPlanEvent Obj = (PAProductionPlanEvent)EventArgs.EventObject;
HandleProductionPlanEvent((int)Obj.ProductId);
}
else if (EventArgs.EventObject.GetType() == typeof(PAProductionSetupEvent))
{
PAProductionSetupEvent Obj = (PAProductionSetupEvent)EventArgs.EventObject;
HandleProductionSetupEvent((int)Obj.DimensionA);
}
else if (EventArgs.EventObject.GetType() == typeof(PAProductionPlanStartEvent))
{
PAProductionPlanStartEvent Obj = (PAProductionPlanStartEvent)EventArgs.EventObject;
HandleProductionPlanStartEvent((int)Obj.UserId,(int)Obj.ProductionUnitId);
}
else if (EventArgs.EventObject.GetType() == typeof(PAProductionStatsEvent))
{
PAProductionStatsEvent Obj = (PAProductionStatsEvent)EventArgs.EventObject;
HandleProductionStatsEvent((int)Obj.ActualBadQuantity);
}
```

Glossary

Material Definition: Contains properties defining a specific material (equivalent to a materials data sheet), but is not the actual material itself in the Proficy S95 data model. It may have material classes associated with it or can stand alone. Properties are passed to the definition.

Material Model: Defines all created or consumed materials in the Proficy S95 data model. Materials are components that make something else and can be a finished product, intermediate product, or raw material--all of which can be tracked individually or used up as part of production. In a Plant Applications database, materials do not have a separate definition and are never considered as separate from a product. In a UMDB, material definitions created in the Proficy S95 data model are linked to the Plant Applications data model to associate materials with specifications and quality functions.

Material Property: Used in the Proficy S95 data model to describe Material Definitions and Classes. In a UMDB, Material Properties are linked to variables for representation in the Plant Applications data model. Because Material Properties are not associated with a unit in the Plant Applications data model, they are considered "unit-less" variables and listed under the S95 Department category in the Plant Model.

Material Specification: Contains Material Definitions and classes in the Proficy S95 data model. It is associated with the Production Model and is not part of the Material Model.

Unified Manufacturing Database (UMDB): A combined database with Plant Applications and S95 ("SOA") records.