GE VERNOVA

PROFICY®SOFTWARE & SERVICES

PROFICY BATCH EXECUTION 5.6

WorkInstruction Manual

Proprietary Notice

The information contained in this publication is believed to be accurate and reliable. However, GE Vernova 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 GE Vernova. Information contained herein is subject to change without notice.

Trademark Notices

"VERNOVA" is a registered trademark of GE Vernova. "GE VERNOVA" is a registered trademark of GE Aerospace exclusively licensed to GE Vernova. The terms "GE" and the GE Monogram are trademarks of GE Aerospace, and are used with permission. All other trademarks are the property of their respective owners.

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

About This Guide

This guide provides an overview of GE's WorkInstruction™ software. It is intended for users interested in understanding the application of WorkInstruction and engineers who are responsible for designing and configuring a Batch Execution and WorkInstruction system.

Reference Documents

Refer to the following manuals for more information on configuring and using the Batch Execution product with WorkInstruction. These manuals are available in electronic form. You can access these manuals by selecting Electronic Books from the Batch Execution program group or from any Help menu.

- System Configuration Manual: for information about Soft Phase Server configuration.
- Phase Programming Manual: for information about configuring the phase logic for the Soft Phase Server.
- PLI Development Manual: for information about configuring the PLI to work with the Soft Phase Server.
- Equipment Configuration Manual: for information about configuring your area model and phase (soft phases and EWI phases).
- Custom Applications manual: for information about configuring VBIS.
- ISA-S88.01 Batch Control Part 1: Models and Terminology – for more information on this standard as defined by the SP88 committee.

Introduction to WorkInstruction

To ensure product quality and consistency, manufacturers in the process industries traditionally applied automation to help eliminate variability. With most manufacturing plants now somewhat automated, the plant operator's function is increasingly seen as a potential area where you can gain benefits from the reduction of variability. This perception is particularly true in industries where operator actions during manufacturing can have significant safety or environmental impact, and the recording and verification of those actions may be a regulatory requirement.

This manual describes GE's solution for reducing operator variability in batch manufacturing: WorkInstruction. WorkInstruction allows for generating, displaying, manipulating, and recording electronic work instructions (EWIs) and signing these instructions with electronic signatures. *EWIs* contain the required operator action, data capture, deviation record (if required), operator verification, and/or confirmation requirements that pertain to a single process step.

Since an EWI is an electronic record kept in lieu of a paper record, it must be signed as dictated by Current Good Manufacturing Practice (cGMP) regulations, with either a single signature, or a double signature. Electronic records and electronic signatures generated by WorkInstruction may adhere to the requirements specified by the Food and Drug Administration's 21 CFR, Part 11 regulation.

While EWIs are essentially an operator interface function, you can use them both within an exclusively manual process, or synchronized with automatic batch processing.

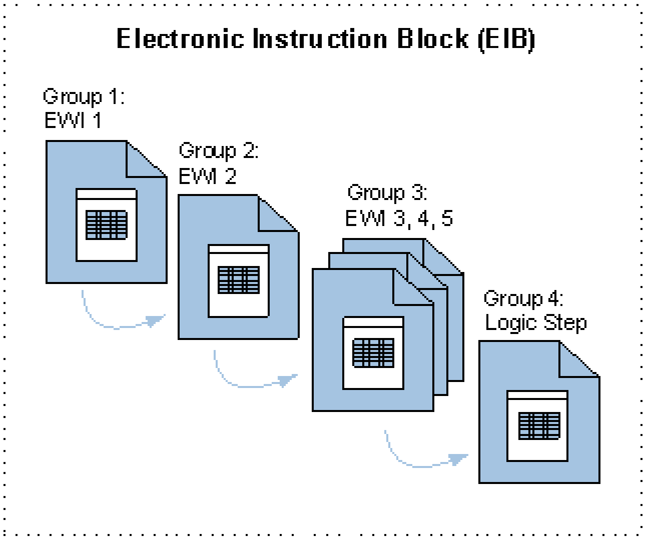
Understanding EWIs, Logic Steps, and EIBs

An electronic work instruction (EWI) contains the following information:

- Instruction text
- Operator data entry requirements
- Operator signing requirements and rights

A Logic Step allows you to perform steps based on conditional logic and looping. Logic Steps may be configured to require an operator action and electronic signing.

An electronic instruction block (*EIB)* is a collection of EWI and Logic Steps that pertain to a single process step, as shown in the following figure. A single process step refers to a step in the within the SFC (Sequential Function Chart).



Overview of an EIB

The operator completes the steps in the order of its group number. If a group contains more than one step, then the operator can execute these steps in any order. For instance, in figure above, after completing and signing EWI 1 and EWI 2, the operator can then complete EWI 3, 4, and 5 in any order.

Once an operator completes and signs EWIs, Logic Steps, and the EIB (the collection of EWIs and Logic Steps), WorkInstruction records the information as part of an electronic record. This information includes:

- Instruction text
- Operator comments
- Data entered, including EGU

- Electronic signatures
- Deviation comments and the actions taken to correct them (manual or automatic)
- Date/time stamp

The sections that follow describe the WorkInstruction components, and explain how operators interact with WorkInstruction and the Batch Execution system.

WorkInstruction Key Features

WorkInstruction provides the following key features:

- Ability to define the work to be carried out, the sequence in which the work occurs, the criteria for data entry, and level of authorization required to sign off on the work instructions.
- Incorporation of electronic signatures and electronic records.
- 21 CFR Part 11 support capabilities.
- Client-server architecture.
- Ability to include multiple instructions per process step.
- Ability to define instruction execution in sequence or in parallel within an EIB.
- Capability to embed Intelligent Verification Objects (IVOs) and field codes within EWI and Logic Steps. This provides the ability to read in real-time data from the Batch Execution Server or a SQL database.
- Ability to include hyperlinks to pictures and documents within EWIs.
- Capability to include instructions originating from the Soft Phase Server or the control system.
- Single ISA S88.01 recipe that controls both the automated and the manual process of manufacturing.
- Common batch record for both automatically collected data and operator actions.
- Ability to use the latest version of an EIB instead of a specified version.
- Ability to skip one or more EWI or Logic Steps.
- Ability to re-execute an EWI Step.
- Ability to repeat the execution of EWI and Logic Steps with the use of Logic Step looping.

Components of WorkInstruction

The following table describes the key components of the WorkInstruction product and how each is used.

WorkInstruction Components	
Component	**Description**
WorkInstruction EIB Server	Application that controls the execution of an EIB. Located on the same computer as the Batch Execution Server. The EWI ActiveX control runs as a client of the EIB Server. The EIB Server is a client of VBIS.
WorkInstruction Editor	Authoring tool used to create and define EIBs, Logic Steps, and EWIs. Usually located on the batch development computer. Can be launched independently or from the Batch Execution Recipe Editor. Saves the EIBs and EWIs as XML files and stores them in Microsoft® Visual SourceSafe®.
WorkInstruction EWI ActiveX control	Display component used by operators to view and execute EIBs, Logic Steps, and EWIs. Usually located on an iFIX node or web server.
Batch Execution Soft Phase Server	OPC Server, running under Windows, which acts as a PC-based controller between the Batch Execution Server and soft phases. Provides the state machine and PLI for each soft phase in the system. Used with the WorkInstruction product to send electronic work instructions to the operator (7001 requests). Resides on the same computer as the Batch Execution Server.
Microsoft® Visual SourceSafe®	The software application used for version control and managing of the EIB files (stored as XML files). The Microsoft Visual SourceSafe database typically resides on a separate computer.
WorkInstruction Database Tables	Set of tables that capture the results of the execution of an EIB, the electronic batch record. WorkInstruction supports SQL Server and Oracle. These tables are part of the standard Batch Execution Archiver table set.
WorkInstruction Demo	A demo project that demonstrates the features of WorkInstruction.

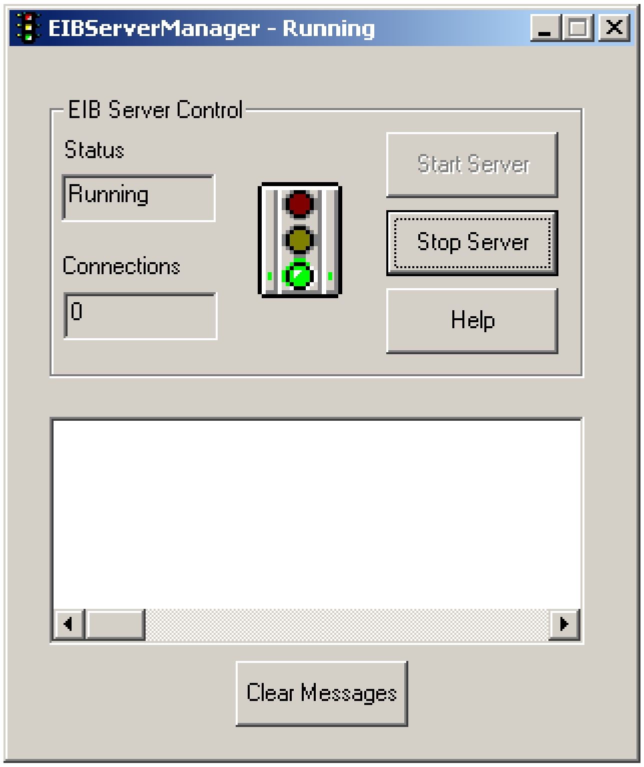
The sections that follow outline the individual features of the WorkInstruction components listed in the table.

WorkInstruction EIB Server

The WorkInstruction EIB Server provides the following features:

- Controls the execution of EIBs independently from the Batch Execution Server execution of the recipe procedures.
- Interfaces with VBIS to receive EIBs to process and send events to the electronic batch record.
- Provides for distributed use of work instructions to one or more EWI ActiveX control clients.
- Interfaces with Windows security to provide electronic signatures for 21 CFR Part 11.
- Provides warm restart ability to allow you to return the EIB Server to its last known state to recover data after an unexpected loss of the EIB Server.

To start the EIB Server, WorkInstruction comes with a separate application called the EIB Server Manager. This application is included in the Batch Execution folder with the other Batch Execution applications. The following figure shows the EIB Server Manager.



EIBServerManager - Running

Status: Running

Connections: 0

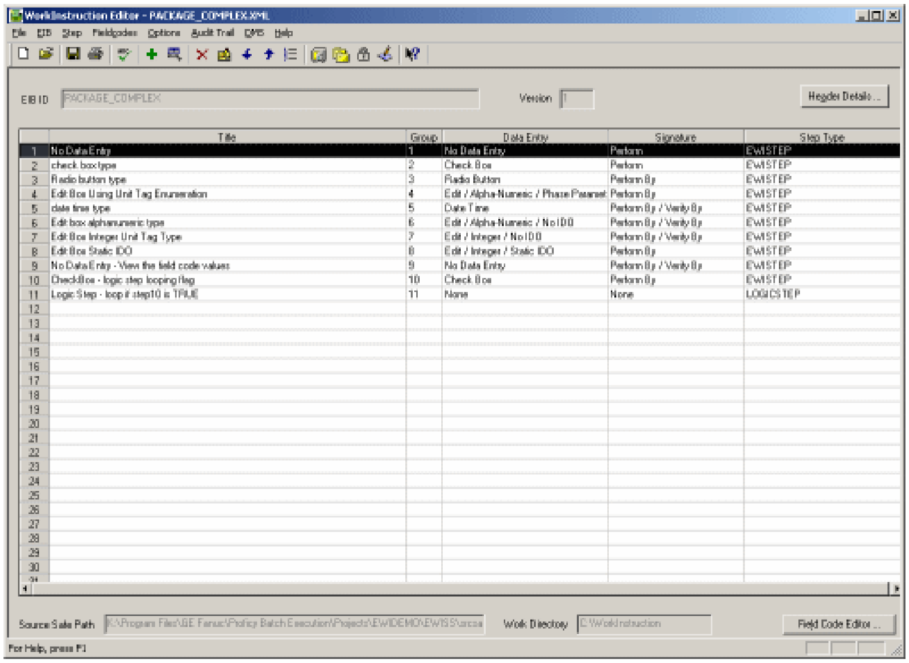
EIB Server Manager

WorkInstruction Editor

The WorkInstruction Editor provides the following features:

- Ability to define electronic work instructions and logic steps that represent the plant work flow.
- Storage of electronic work instruction blocks (EIBs) in XML format using an XML schema to validate a document.
- Capability to control documents with the Document Management System (DMS) for full version control through Microsoft Visual SourceSafe.
- Ability to enable an audit trail that can be viewed in the Batch Execution Audit Reporter.
- Default signature template configuration. By defining the defaults in this template, the signature requirement section automatically populates with these defaults when you create a new EWI Step, Logic Step, or EIB.
- Field Code Editor and Field Code support throughout an EIB.
- Intelligent Verification Objects (IVOs) for robust deviation detection.
- Ability to read and write data from and to Batch Execution phase parameters and unit tags.
- Advanced sequencing of steps using Logic Steps.
- Support for step verification.
- Ability to regroup steps, and automatically arrange and re-arrange group numbers.
- Ability to move steps up and down in the sequence.

The following figure displays the WorkInstruction Editor application.


WorkInstruction Editor - PACKAGE_COMPLEX.XML
File EIB Step Fieldcodes Options Audit Trail DMS Help
Header Details ...
Field Code Editor ...
For Help, press F1

EIB ID: PACKAGE_COMPLEX

Version: 1

	Title	Group	Data Entry	Signature	Step Type
1	No Data Entry	1	No Data Entry	Perform	EWISTEP
2	check box type	2	Check Box	Perform	EWISTEP
3	Radio button type	3	Radio Button	Perform By	EWISTEP
4	Edit Box Using Unit Tag Enumeration	4	Edit / Alpha-Numeric / Phase Paramet	Perform By	EWISTEP
5	date time type	5	Date Time	Perform By / Verify By	EWISTEP
6	Edit box alphanumeric type	6	Edit / Alpha-Numeric / No IDO	Perform By / Verify By	EWISTEP
7	Edit Box Integer Unit Tag Type	7	Edit / Integer / No IDO	Perform By / Verify By	EWISTEP
8	Edit Box Static IDO	8	Edit / Integer / Static IDO	Perform By	EWISTEP
9	No Data Entry - View the field code values	9	No Data Entry	Perform By / Verify By	EWISTEP
10	CheckBox - logic step looping flag	10	Check Box	Perform By	EWISTEP
11	Logic Step - loop if step10 is TRUE	11	None	None	LOGICSTEP
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					

Source Safe Path: K:\Program Files\GE Fanuc\Proficy Batch Execution\Projects\EWIDEMO\EWISS\srcsa

Work Directory: C:\WorkInstruction

WorkInstruction Editor Application

WorkInstruction EWI ActiveX Control

The EWI ActiveX control provides the following features:

- Ability to run in an OLE compliant container, such as Microsoft Explorer or the Proficy iFIX WorkSpace.
- Capability for operator data entry, instruction signing, and deviation or comment entry.
- Logging of all data, signatures, deviations, and comments through the EIB Server.
- Ability to customize the look of the EWI ActiveX display through property pages and methods and properties exposed for scripting.

The operator views EWIs and Logic Steps from the EWI ActiveX control, and not from the Batch Execution Client.

Refer to Configuring the EWI ActiveX Control for more details about the EWI ActiveX control. This section also includes a sample graphic of the EWI ActiveX control.

Batch Execution Soft Phase Server

The Soft Phase Server provides a client/server solution for developing soft phases. It can support multiple clients connected simultaneously. The Soft Phase Server also provides a single server for all soft phases in the system, eliminating the need to develop and deploy separate applications for each soft phase. The WorkInstruction demo project uses the Soft Phase Server in OPC Simulation Mode.

For more information on the Soft Phase Server, refer to the Configuring the Soft Phase Server section in the System Configuration Manual.

Microsoft Visual SourceSafe

With WorkInstruction you use the Microsoft Visual SourceSafe version control system to manage the individual EIBs in your work flow. With the Document Management System (DMS) in the WorkInstruction Editor, you can directly connect to Microsoft Visual SourceSafe.

WorkInstruction stores EIBs as XML documents through the WorkInstruction Editor. You can control which EIBs are under development, which EIBs are in production, and which EIBs are obsolete from your production process in Microsoft Visual SourceSafe. The most current version of the EIB displays in the WorkInstruction Production project. You can view the history of an EIB or add comments when you check in or check out a file.

The WorkInstruction Editor provides an audit trail (if you enable it) to track the SourceSafe actions on the local computer. These file actions include adding, checking in, checking out, copying to production project, and moving to obsolete project, and so on. When you enable the Use Signature Verification for DMS Commands option in the Database Info for Audit Trail dialog box, the user must enter a Windows user name and password to perform any of these actions. That user name and password is then recorded in the audit trail.

In addition to requiring a valid SourceSafe user name and password, the user must also enter a Windows password for an additional authority check, when the audit trail is enabled. By performing this extra authority check, WorkInstruction supports the 21 CFR Part 11 "write-to-file" requirements.

Refer to the Configuring Microsoft Visual SourceSafe section for more information and a picture of the WorkInstruction Document Management System dialog box.

WorkInstruction Database Tables

The WorkInstruction tables store the electronic batch record generated during the execution of an EIB, EWI, or Logic Step. The tables are designed to allow for data mining and report creation. All signatures are clearly denoted and time stamped accordingly. In the case of operator data entry, multiple values are stored to accurately reconstruct the choices and selections the operator made.

Refer to the Understanding the WorkInstruction Data Model section for more details about the WorkInstruction database tables.

WorkInstruction Demo Project

WorkInstruction includes a demo project, EWIDemo.wkb, which demonstrates the product features. The EWIDemo.wkb project is a simple project for the Batch Execution WorkSpace, with the area model and recipes already configured. It includes two sample EIBs and a Microsoft SourceSafe database that is installed along with Batch Execution and WorkInstruction products.

The EWIDemo project is located in the Program Files\Proficy\Proficy Batch Execution\projects folder. The SourceSafe database is located in the Program Files\Proficy\Proficy Batch Execution\projects\EWIDemo\EWISS folder. There are a few steps that you must perform for the EWIDemo project to run correctly.

▶To configure the EWIDemo project:

1. Start the Batch Execution WorkSpace.
2. Open the EWIDemo project.
3. On the File menu, click Save. This step makes the EWIDemo the most recent project for the Batch Execution Server to use when it starts up.
4. Create two Windows security groups: Operator and Supervisor. Add at least one user to each group. For detailed instructions on how to do this, see the Configuring Windows Security section.

For more information on setup options, refer to the Readme.txt file in the Program Files\Proficy\Proficy Batch Execution\projects\EWIDemo\recipes folder. Once you configure the EWIDemo project, you can run the samples.

▶To run the samples in the EWIDemo project:

1. Open the Batch Execution Server Manager.
2. Click the Start Server button. The project uses the Soft Phase Server as the simulator. A prompt appears requesting a warm or cold soft phase boot.
3. Click Cold. A prompt appears requesting a warm or cold Batch Execution Server boot.
4. Click Cold.
5. In the Soft Phase Server application menu, select Open from the File menu. The Open dialog box appears.
6. Select the EWIdemo.sim file and click OK.

   NOTE: *You can preconfigure the EWIDemo.sim to load when starting the Soft Phase Server. To do this, enter the full path to this file in the Batch Execution WorkSpace in the Batch*

Execution Configuration dialog box on the Restart Control tab, in the Database Configurations area, in the Soft Phase Server field.

7. Start the EIB Server. A prompt appears requesting a warm or cold boot.
8. Click Cold.
9. Start the Batch Execution Client or BatchList ActiveX control.

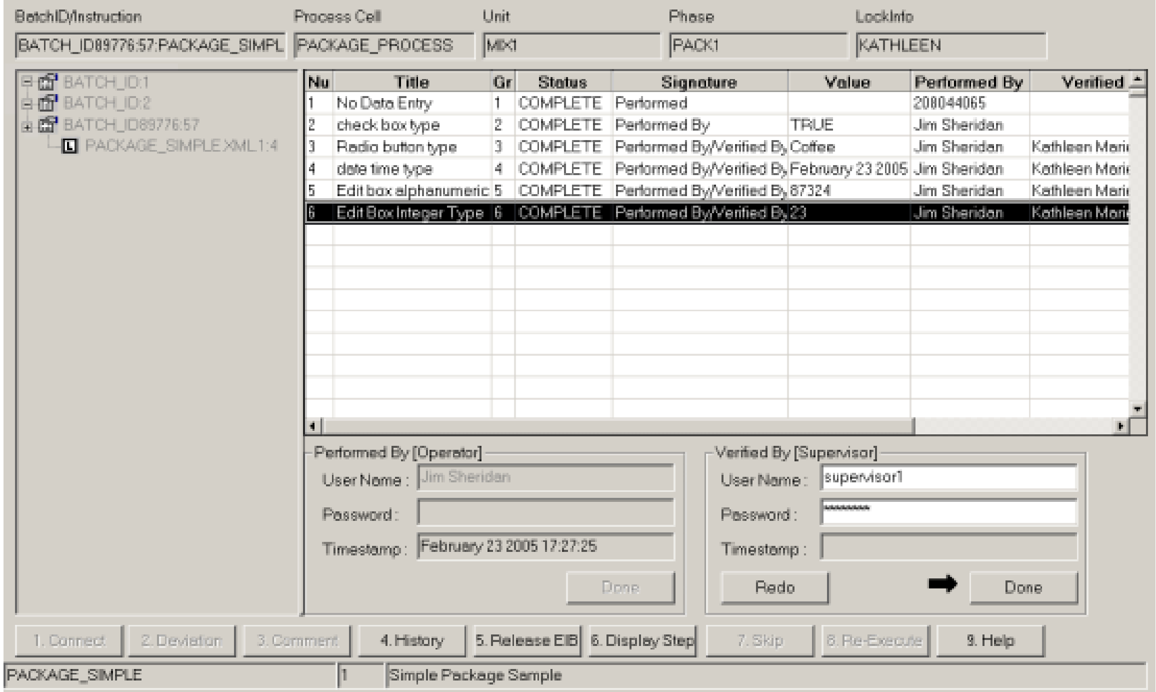
   NOTE: *Batch Execution includes a Microsoft Visual Basic demo that contains the WorkInstruction and Batch Execution ActiveX controls. You can find this demo in this location: c:\Program Files\Proficy\Proficy Batch Execution\samples\controls\VisualBasic\IntellutionBatchControls.exe. Batch Execution also includes an HTML sample in this location: c:\Program Files\Proficy\Proficy Batch Execution\samples\controls\HTML\index.htm. If you use the HTML sample, make sure that you run the Setup.bat file located in this folder first.*

10. In the Batch Execution Client or BatchList ActiveX control, schedule a recipe to run. Choose Package_Simple or Package_Complex.
11. Start the batch.
12. Use the EWI ActiveX control, EWIX.ocx, to process the EIB prompt and its electronic work instructions.

    For example, for purposes of this demo, you can run the EWI ActiveX control in the ActiveX Control Test Container application which Microsoft® Visual Studio® provides.

Example of EWIDemo

The following figure shows an example of the EWI ActiveX control, with the demo project running.



BatchID/Instruction: BATCH_ID89776:57:PACKAGE_SIMPL
Process Cell: PACKAGE_PROCESS
Unit: MIX1
Phase: PACK1
LockInfo: KATHLEEN

BATCH_ID:1
BATCH_ID:2
BATCH_ID89776:57
PACKAGE_SIMPLE.XML:1:4

Nu	Title	Gr	Status	Signature	Value	Performed By	Verified
1	No Data Entry	1	COMPLETE	Performed		208044065	
2	check box type	2	COMPLETE	Performed By	TRUE	Jim Sheridan	
3	Radio button type	3	COMPLETE	Performed By/Verified By	Coffee	Jim Sheridan	Kathleen Mari
4	date time type	4	COMPLETE	Performed By/Verified By	February 23 2005	Jim Sheridan	Kathleen Mari
5	Edit box alphanumeric	5	COMPLETE	Performed By/Verified By	87324	Jim Sheridan	Kathleen Mari
6	Edit Box Integer Type	6	COMPLETE	Performed By/Verified By	23	Jim Sheridan	Kathleen Mari

Performed By [Operator]
User Name : Jim Sheridan
Password :
Timestamp : February 23 2005 17:27:25

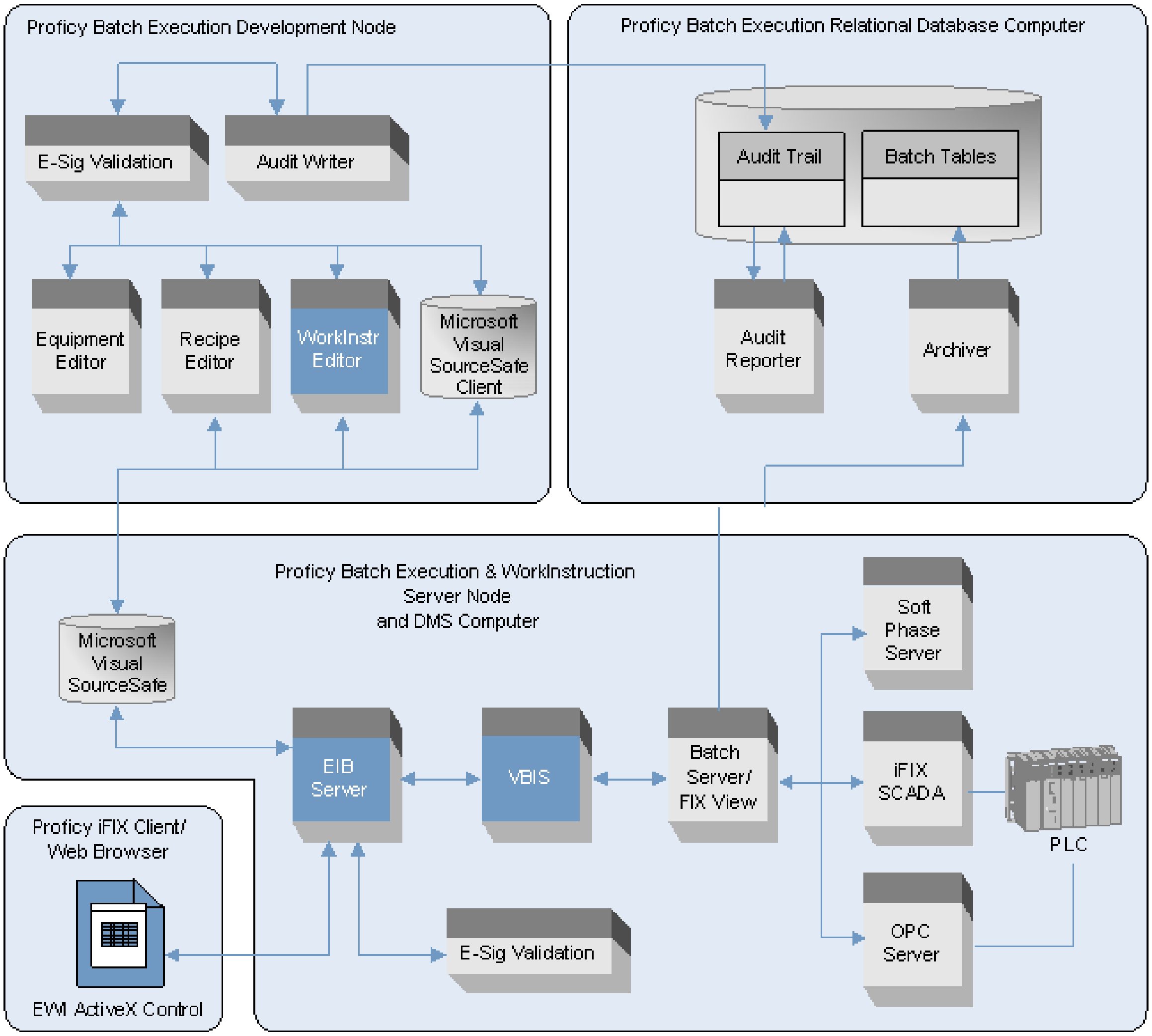
Verified By [Supervisor]
User Name : supervisor1
Password : ********
Timestamp :

PACKAGE_SIMPLE | 1 | Simple Package Sample

WorkInstruction Demo

Understanding How WorkInstruction Works With Batch Execution

The following figure describes how the components of WorkInstruction typically function with the Batch Execution components. It also displays where the computers, such as the server, database, development, and client computers, typically run each component.



Proficy Batch Execution Development Node
E-Sig Validation
Audit Writer
Equipment Editor
Recipe Editor
WorkInstr Editor
Microsoft Visual SourceSafe Client
Proficy Batch Execution Relational Database Computer
Audit Trail
Batch Tables
Audit Reporter
Archiver
Proficy Batch Execution & WorkInstruction Server Node and DMS Computer
Microsoft Visual SourceSafe
Soft Phase Server
EIB Server
VBIS
Batch Server/ FIX View
iFIX SCADA
PLC
OPC Server
E-Sig Validation
Proficy iFIX Client/ Web Browser
EWI ActiveX Control

Overview of Batch Execution and WorkInstruction Processes

Getting Started

The sections that follow describe the information that you need to know to install WorkInstruction. It includes the following topics:

- Software Requirements
- Deploying WorkInstruction
- Overview of WorkInstruction Setup
- Configuring Windows Security
- Configuring Microsoft Visual SourceSafe
- Defining Batch Execution Project Settings for WorkInstruction
- Building Electronic Work Instructions in the WorkInstruction Editor
- Configuring Phases for EWIs
- Configuring iFIX with WorkInstruction

Software Requirements

The following software is necessary to use the WorkInstruction product:

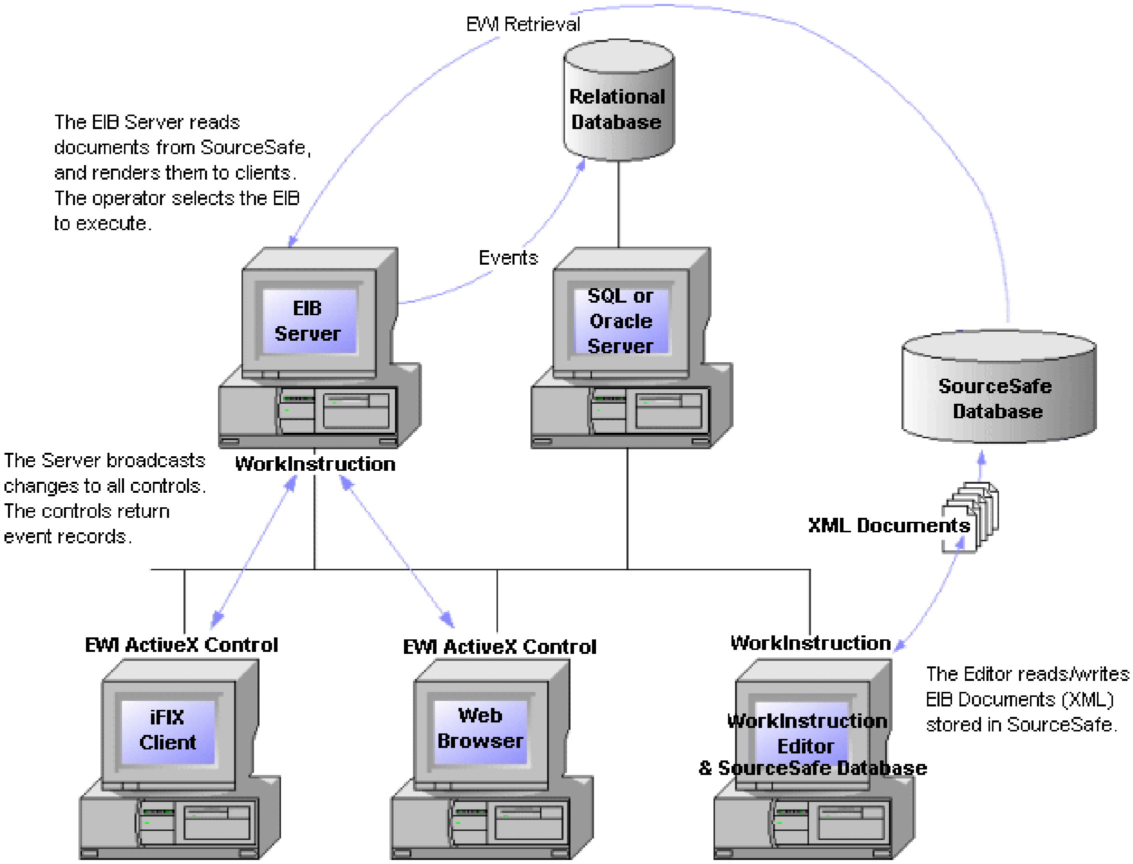
- Proficy Batch Execution
- Microsoft Visual SourceSafe 6.0 or 2005

IMPORTANT: *WorkInstruction and Microsoft Visual SourceSafe must be installed on the same computer. A remote connection to Visual SourceSafe is not supported with the WorkInstruction product.*

For Batch Execution Client and Server requirements, see Getting Started in the System Configuration Manual.

Deploying WorkInstruction

For the WorkInstruction product, the Batch Execution Server computer also runs the WorkInstruction EIB Server. You will likely designate a different computer for the relational database, as in a Batch Execution setup without WorkInstruction. The WorkInstruction EWI ActiveX control runs on the client computers, while the WorkInstruction Editor application usually runs on a development computer. The following figure shows a typical configuration of WorkInstruction.

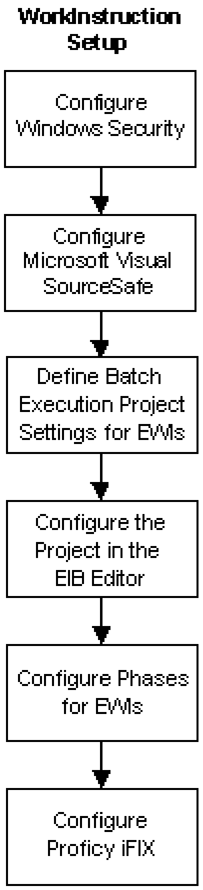

EWI Retrieval
Relational Database
The EIB Server reads documents from SourceSafe, and renders them to clients. The operator selects the EIB to execute.
Events
EIB Server
SQL or Oracle Server
SourceSafe Database
WorkInstruction
The Server broadcasts changes to all controls. The controls return event records.
XML Documents
EWI ActiveX Control
EWI ActiveX Control
WorkInstruction
The Editor reads/writes EIB Documents (XML) stored in SourceSafe.
iFIX Client
Web Browser
WorkInstruction Editor & SourceSafe Database

Sample Configuration

The only requirement for your computer setup is that the EIB Server and Batch Execution Server reside on the same node. If you are using the Soft Phase Server, it should also reside on that node.

Overview of WorkInstruction Setup

You configure WorkInstruction by using Windows security, Microsoft Visual SourceSafe, Proficy Batch Execution WorkSpace, WorkInstruction Editor, Proficy iFIX WorkSpace, and the Batch Execution Soft Phase Server or PLI. The following figure shows an overview of the steps needed to configure WorkInstruction with iFIX.


WorkInstruction Setup
Configure Windows Security
Configure Microsoft Visual SourceSafe
Define Batch Execution Project Settings for EWIs
Configure the Project in the EIB Editor
Configure Phases for EWIs
Configure Proficy iFIX

Overview of Setup Steps

The sections that follow describe these steps, in the order listed in the figure above.

After you configure your system with the steps listed in these sections, you create instances of the EWI ActiveX control on the operator computers, so that they can view the EWIs. Refer to the Configuring the EWI ActiveX Control section for details.

Configuring Windows Security

To configure security for WorkInstruction, you can use existing Windows security groups and user accounts or define new Windows groups and user accounts. You assign security group checks to each EWI Step, Logic Step, and EIB when authoring an EIB. During execution, only users who are members of the assigned groups can electronically sign the EWI Step, Logic Step, or EIB.

For example, create two groups, one named "Supervisor," and the other "Operator" to distinguish between personnel who can electronically sign a "Performed By" signature requirement, versus someone who can sign a "Verified By" signature requirement. Likewise, you can create groups to represent the production lines in your plant, so that operators working on one particular line cannot sign an EWI executing on another production line. You need at least one group. The same user cannot sign both the Performed By and Verified By signature requirements, yet the same user may belong to both groups. The Performed By and Verified By signatures are compared to the Windows full user names of the users.

The WorkInstruction security service supports both Local and Domain groups. The WorkInstruction security service resides on the same node as the EIB Server. Upon receiving a request to validate a signature, it checks the local computer for the specified group. If the group is not found, it then checks for a Domain level group to verify the user name and password.

You can choose to configure your WorkInstruction groups and user names on the EIB Server computer or as part of your plant's overall Domain Security Configuration.

The following instructions lead you through defining two local groups, one called "Operator" and one called "Supervisor." See the following section for more details: Adding Operators and Supervisors.

Again, you are free to define the groups' names or reuse existing groups already present in your plant security configuration.

Adding Operators and Supervisors

The following steps first describe how to add the user names of the all operators and supervisors. Then, the steps outline how to add the actual Operator and Supervisor groups and assign users to each group.

▶To add operators and supervisors:

1. Log in to Windows as an administrator, if you are not already.
2. Click the Start button, and point to Settings, Control Panel, Administrative Tools, and then Computer Management. The Component Services Microsoft® Management Console (MMC) snap-in appears.
3. In the System Tools folder, double-click the Local Users and Groups item.
4. Right-click the Users folder and select New User from the pop-up menu. The New User dialog box appears.
5. Enter information in the User Name, Full Name, Password, and Confirm Password fields.
6. Click Create.
7. Repeat steps 5-6 for each individual user that you want to add.
8. Click Close.
9. Right-click the Groups folder and select New Group from the pop-up menu. The New Group dialog box appears.
10. Enter *Operator* in the Group Name field.

    NOTE: *You do not have to use the group name Operator. You can use the group specific to your work flow. For example, you might enter PackingLineOperator, instead of Operator.*

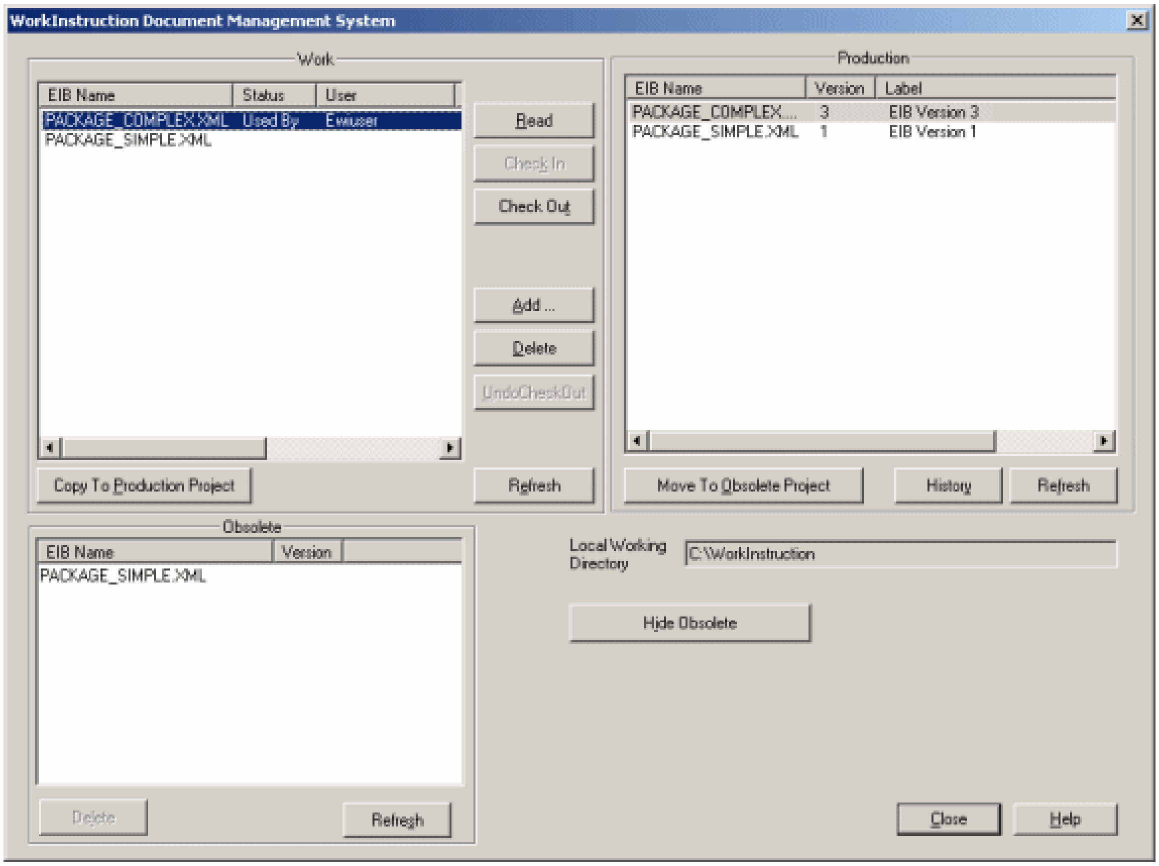
11. Click Add. The Select Users dialog box appears.
12. Enter a user name that you want to add to the Operator group, and then click OK.
13. Repeat steps 11-12 for each user you want to add to the Operator group.
14. After you finish adding users, click Create.
15. Repeat steps 9-14, but this time use *Supervisor* as the group name.

    NOTE: *You do not have to use the group name Supervisor. You can use the name of the supervisory group specific to your work flow. For example, you might enter ShiftManager, instead of Supervisor.*

Configuring Microsoft Visual SourceSafe

With WorkInstruction, you use the Microsoft Visual SourceSafe version control system to manage the individual EIBs in your work flow. With Microsoft Visual SourceSafe you control which EIBs are under development, which EIBs are in production, and which EIBs are obsolete from your production process. Through the WorkInstruction Editor you can access the Document Management System (DMS), which directly connects to Microsoft Visual SourceSafe, after you configure your system.

The following figure shows what the WorkInstruction Document Management System looks like from the WorkInstruction Editor.



EIB Name	Status	User
PACKAGE_COMPLEX.XML	Used By	Ewiuser
PACKAGE_SIMPLE.XML		

EIB Name	Version	Label
PACKAGE_COMPLEX....	3	EIB Version 3
PACKAGE_SIMPLE.XML	1	EIB Version 1

EIB Name	Version
PACKAGE_SIMPLE.XML	

Local Working Directory: C:\WorkInstruction

WorkInstruction Document Management System Dialog Box

Using SourceSafe with Remote Systems

SourceSafe can be used to provide version control in situations where WorkInstruction Editor is on remote systems. However, to do so successfully, you must install a SourceSafe client on every remote computer using WorkInstruction. And, when configuring the SourceSafe path, you must use a UNC pathing.

Working with Microsoft Visual SourceSafe

Once you have installed Microsoft Visual SourceSafe, you need to perform the following steps to use Visual SourceSafe with WorkInstruction:

1. Create a database.
2. Add Microsoft Visual SourceSafe users to the database.
3. Create project folders.
4. Configure Microsoft Visual SourceSafe server settings.
5. Install Microsoft Visual SourceSafe Clients (optional).

The following sections describe these steps in more detail.

Creating a Microsoft Visual SourceSafe Database

▶To create a Microsoft Visual SourceSafe database:

1. Start the Visual SourceSafe Administration application.
2. Create the database:
   - In version 6.0, on the Tools menu, click Create Database.
   - In version 2005, on the File menu, click New Database.
3. In the dialog box or wizard that appears, enter the location where the database files shall reside, and follow the instructions on your screen.

Creating Microsoft Visual SourceSafe Users

Refer to the Microsoft Visual SourceSafe online Help for instructions on adding and configuring user accounts.

Adding Projects to Microsoft Visual SourceSafe

After you create a database in Microsoft Visual SourceSafe, you need to create the following projects in SourceSafe, for use with WorkInstruction:

- EWIWork
- EWIProduction
- EWIObsolete

▶To add projects:

1. In Microsoft Visual SourceSafe, on the File menu, click Create Project. The Create Project dialog box appears.
2. Enter EWIWork and then click OK.
3. Repeat steps 1-2 for EWIProduction.
4. Repeat steps 1-2 for EWIObsolete.

Configuring Microsoft Visual SourceSafe Server Settings

You must disable automatic user login in Microsoft Visual SourceSafe Server. To configure the server settings, perform the following steps from the Administration tool.

▶To disable automatic user login in Microsoft Visual SourceSafe:

1. In the Visual SourceSafe Administration tool, on the Tools menu, click Options. The Source Safe Options dialog box opens.
2. From the General tab, clear the Use network name for automatic user login check box.
3. Click OK.

Adding Microsoft Visual SourceSafe Clients

If you plan to use SourceSafe with WorkInstruction on remote computers, in addition to installing the SourceSafe server, you must install SourceSafe clients on each of the remote computers. Refer to Microsoft's online help for installation instructions and licensing requirements.

If SourceSafe is Installed on a Separate Computer

The user account you create during the Batch install must also exist on the Microsoft Visual SourceSafe Server computer. If you do not install Batch Execution on the same computer as SourceSafe, then you need to create this user on the SourceSafe computer. If you do not specify an account name during install, the BatchExecutive account is used by default.

▶To add the batch user account:

1. Log in to Windows as an administrator, if you are not already.
2. Click the Start button, and point to Settings, Control Panel, Administrative Tools, and then Computer Management. The Component Services Microsoft® Management Console (MMC) snap-in appears.
3. In the System Tools folder, double-click the Local Users and Groups item. Click the Start button, and point to Settings, Control Panel, and then Users Accounts. The User Accounts dialog box appears.
4. Right-click the Users folder and select New User from the pop-up menu. The new user dialog box appears.
5. Enter the user name of the batch account that you created during install (BatchExecutive if none was specified), including the full name and the user's password. Select the Password Never expires check box.
6. Click Create.
7. Click Close.
8. Right-click the Groups folder and select New Group from the pop-up menu. The New Group dialog box appears.
9. In the Group Name field, enter a name.

   NOTE: *You cannot enter the same name as the user. For example, for the default user account of, BatchExecutive, you cannot enter BatchExecutive here.*

10. Click Add. The Select Users dialog box appears.
11. Enter the batch account as the user name that you want to add to your group, and then click OK.
12. Click Create.
13. Click Close.

Defining Batch Execution Project Settings for WorkInstruction

You need to configure your Batch Execution project to enable the Recipe Editor to locate Microsoft Visual SourceSafe. If you do not define the location of the SourceSafe database, you will not be able to assign EIBs to phases in the Recipe Editor.

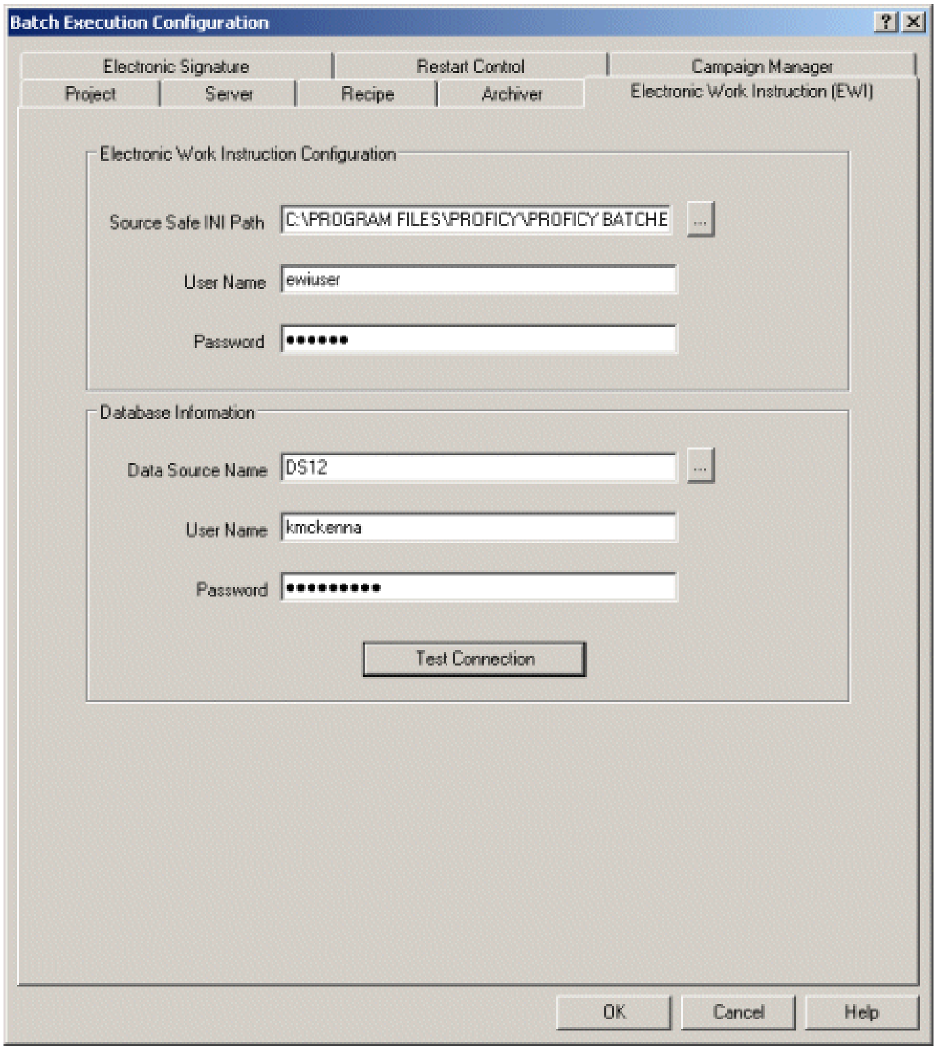
▶To define the location of the SourceSafe database from the WorkSpace:

1. From the Batch Execution WorkSpace, double-click the Batch Execution Configuration icon in the WorkSpace tree. The Batch Execution Configuration dialog box appears.
2. From the Project tab, enter the location where you want to store the EIBs that get pulled from Microsoft Visual SourceSafe in the EIB Directory Store edit box.
3. Click the Electronic Work Instruction (EWI) tab.
4. Enter the path to the Microsoft Visual SourceSafe database .INI file. An example of a path is as follows: C:\Program Files\Proficy\Proficy Batch Execution\Projects\EWIDEMO\EWISS\srcsafe.ini.
5. Enter a valid SourceSafe user name and password. The EIB Server uses this account to read the EIB from SourceSafe.

   NOTE: *If you are using a remote computer with a SourceSafe client installed, use UNC pathing.*

6. Optionally, if you want to use dynamic SQL IDOs, enter the name of an ODBC data source DSN, and a user name and a password.
7. Click OK.

The following figure displays an example of the Batch Execution Configuration dialog box with user-supplied information for Microsoft Visual SourceSafe for the EWI demo project.



Electronic Work Instruction Configuration

Source Safe INI Path: C:\PROGRAM FILES\PROFICY\PROFICY BATCHE

User Name: ewiuser

Password: ••••••

Database Information

Data Source Name: DS12

User Name: kmckenna

Password: ••••••••••

Batch Execution Configuration Dialog Box

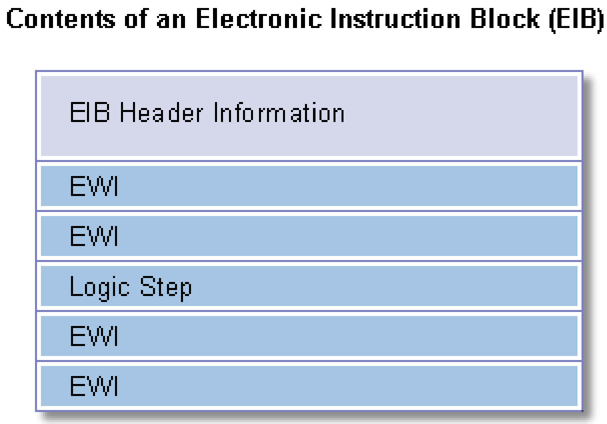
The EWI Demo project includes a SourceSafe database with two sample EIBs that you can check in and out of the SourceSafe. The user name for this database is *ewiuser* and the password is *ewiuser*.

NOTE: *The EIBs in the demo project are configured to use the Performed By group name Operator, and the Verified By group name of Supervisor.*

Building Electronic Work Instructions in the WorkInstruction Editor

When you are ready to define your electronic work instructions (EWIs), you use the WorkInstruction Editor tool. Through the WorkInstruction Editor you can create and modify collections of EWIs and move them from working projects, directly into production.

WorkInstruction uses the term electronic instruction block (EIB) to refer to a collection of EWI or Logic Steps that you execute in sequence or in parallel, along with the EIB header information. The following figure describes an example of an individual EIB.



Example of EIB Contents

You define both the EIB header information and the EWI and Logic Steps for each EIB within the WorkInstruction Editor. Each EIB may contain up to 500 work instructions.

The following sections describe the EIB header information and EWIs in detail. It also describes the signature requirements and security groups within an EIB and an EWI and how to step up the EIB and EWIs in the WorkInstruction Editor.

What Does it Mean to Run EWIs in Parallel?

Two EWIs are said to run in parallel when they possess the same group number within the EIB step list. Even though you can only execute one EWI at a time, it is said that EWIs run in parallel if the operator can choose which EWI to perform. You can have as many EWIs as you want with the same group number. EWIs are sequential if they are in different groups. Consider the following example:

EWI # Description Group

First EWI 1
Second EWI 2
Third EWI 3
Fourth EWI 4
Fifth EWI 4

In the example above, First EWI will first execute first. Since First EWI is alone in the group (group 1), the operator must perform that one first. When complete, Second EWI AND Third EWI become active. Since they both are within the same group (group 2) they can be performed in any order. Second EWI and Third EWI are said to be running in parallel. When both Second EWI AND Third EWI are complete (in either order) then Fourth EWI is active and must be complete and so on.

EIB Header Information

The EIB header information describes the context of the work instructions, who created, reviewed, and authored the work instructions, along with the appropriate level of security to acquire and execute the EIB in production. The characters that you can enter for all fields contained in the EIB header information include A-Z, 0-9, and the underscore character.

The following fields describe the EIB header information:

EIB ID – the internal ID number of the EIB. This is a read-only field.

EIB Version – the version number of the EIB. This is a read-only field.

Document Control Number – the document control number for the EIB. This reference number is optional; it is an internal number specific to your plant's operational processes. This field is only provided for informational purposes. You can enter up to 30 characters.

EIB Title – the name of the EIB. You can enter up to 80 characters.

Author (First Name) – the first name of the EIB author. You can enter up to 30 characters.

Author (Middle Initial) – the middle initial of the EIB author. You can enter up to 3 characters.

Author (Last Name) – the last name of the EIB author. You can enter up to 30 characters.

Verified By (First Name) – the first name of the user required to verify completion of the EIB. You can enter up to 30 characters.

Verified By (Middle Initial) – the middle initial of the user required to verify completion of the EIB. You can enter up to 3 characters.

Verified By (Last Name) – the last name of the user required to verify completion of the EIB. You can enter up to 30 characters.

Approved By (First Name) – the first name of the user required to approve completion of the EIB. You can enter up to 30 characters.

Approved By (Middle Initial) – the middle initial of the user required to approve completion of the EIB. You can enter up to 3 characters.

Approved By (Last Name) – the last name of the user required to approve completion of the EIB. You can enter up to 30 characters.

Document Control Status – the document control status, if required. For example: working, released, or obsolete. This optional field is for informational purposes only. You can enter up to 30 characters.

Effective From Date – the beginning date of the EIB.

Effective Until Date – the end date of the EIB.

Signature Requirements – describes whether the user must enter a Performed By or Verified Group user name.

Performed By Group – the Windows group name to which the operators belong. You can enter up to 30 characters. During execution of an EIB, only users who are members of this group can electronically sign the EIB as "Performed By" and enter operator comments. This field is required.

Verified By Group – the Windows group name to which the supervisors belong. You can enter up to 30 characters. During execution of an EIB, only users who are members of this group can electronically sign the EIB as "Verified By" and enter operator comments. This field is required.

Release Override – if selected, described the Windows group name to which the user performing the release override must belong.

Default Name for Unchecked Properties – the Windows computer name or login name that is captured by default if there is no performed by or verified group defined.

Acquire – if selected, describes the Windows group name to which the user performing the acquire must belong. If this option is *not* selected, the default name (computer name or login name) is captured, and the user is not prompted to enter a signature.

Release – if selected, describes the Windows group name to which the user performing the release must belong. If this option is *not* selected, the default name (computer name or login name) is captured, and the user is not prompted to enter a signature.

Deviation – if selected, describes the Windows group name to which the user performing the deviation must belong. If this option is *not* selected, the default name (computer name or login name) is captured, and the user is not prompted to enter a signature.

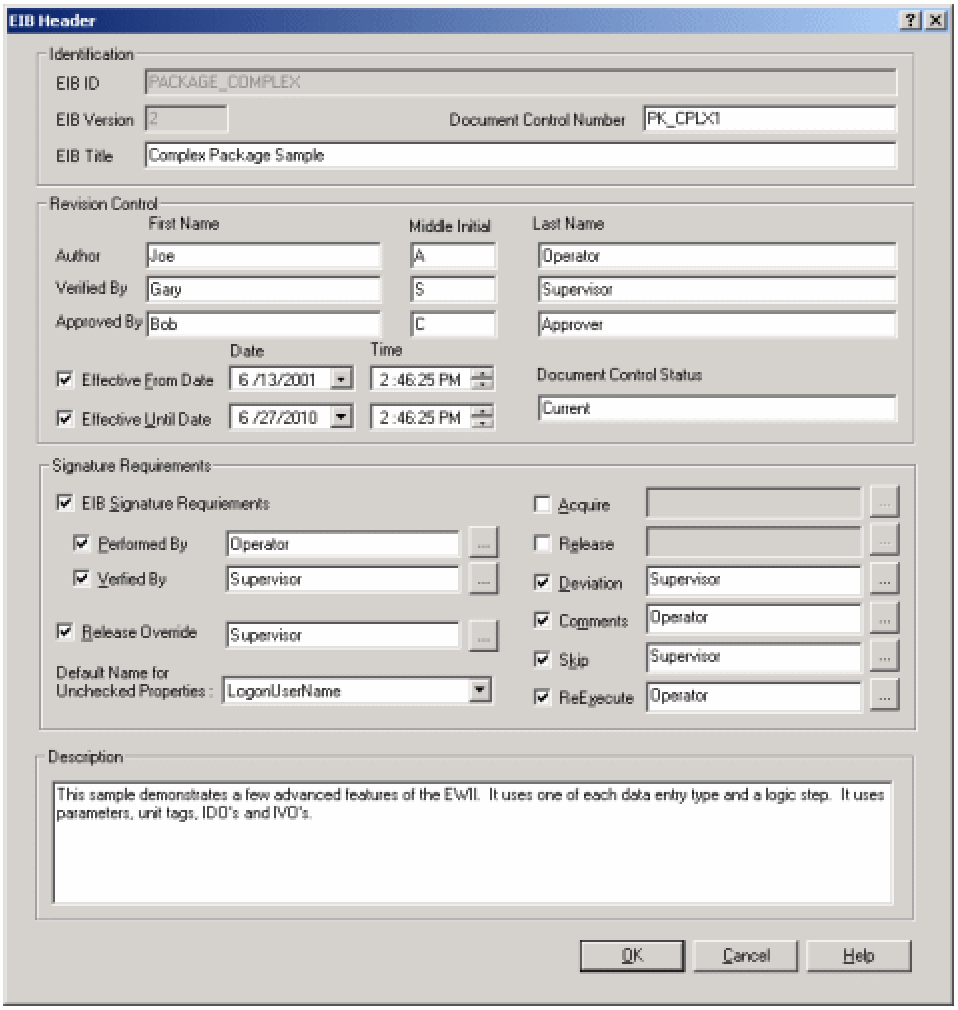
Comments – if selected, describes the Windows group name to which the user adding the comment must belong. If this option is *not* selected, the default name (computer name or login name) is captured, and the user is not prompted to enter a signature.

Skip – if selected, describes the Windows group name to which the user performing the skip function must belong. If this option is *not* selected, the default name (computer name or login name) is captured, and the user is not prompted to enter a signature.

ReExecute – if selected, describes the Windows group name to which the user performing the re-execution must belong. If this option is *not* selected, the default name (computer name or login name) is captured, and the user is not prompted to enter a signature.

Description – a description for the EIB. You can enter up to 1024 characters.

You can define all of the EIB header information fields in the WorkInstruction Editor from the Header dialog box, as shown in the following figure.



Identification

EIB ID: PACKAGE_COMPLEX

EIB Version: 2

Document Control Number: PK_CPLX1

EIB Title: Complex Package Sample

Revision Control

	First Name	Middle Initial	Last Name
Author	Joe	A	Operator
Verified By	Gary	S	Supervisor
Approved By	Bob	C	Approver

	Date	Time
☑ Effective From Date	6 /13/2001	2 :46:25 PM
☑ Effective Until Date	6 /27/2010	2 :46:25 PM

Document Control Status: Current

Signature Requirements

☑ EIB Signature Requirements

☑ Performed By: Operator

☑ Verified By: Supervisor

☑ Release Override: Supervisor

Default Name for Unchecked Properties : LogonUserName

☐ Acquire

☐ Release

☑ Deviation: Supervisor

☑ Comments: Operator

☑ Skip: Supervisor

☑ ReExecute: Operator

Description

This sample demonstrates a few advanced features of the EWI. It uses one of each data entry type and a logic step. It uses parameters, unit tags, IDO's and IVO's.

WorkInstruction Editor Application

EWI Information

Each EWI consists of three parts: the instruction(s) to be carried out, the data entry required by the operator, and the appropriate level of security needed to execute the EWI.

The following fields describe the contents of an EWI:

Label (in all data entry sheets) – the label that you want to appear for the data entry group box. You can enter up to 30 characters, including A-Z, 0-9, a colon (:) character, and the underscore character.

Instructional Text – the instructional text that you want to display with the EWI. You can enter up to 4096 characters, including A-Z, 0-9, word spaces, and the underscore character.

Field Codes – use this drop down-list to add field codes to instructional text. Click the Add button to open the Field Code Editor.

Data Entry – select the type of data entry from the drop-down list that you want to allow the operator to enter. Click Details to open more specific data entry requirements. For instance, if Edit Box is selected from the drop-down list, the Edit Page dialog box displays. From here you can select the data type (integer, real, or alphanumeric), the IDO, and/or IVO.

Title – the title for the EWI. You can enter up to 80 characters, including A-Z, 0-9, and the underscore character.

Run Task – the task that you want to run for this individual EWI. You can enter up to 250 characters, including A-Z, 0-9, and the underscore character. This field is optional. If Run Task is selected, enter a label in the Runtime Caption field. This task displays in the EWI at run time as a command button.

Signature Requirements – describes whether the user must enter a "Performed By" or "Verified By" group user name.

Performed By Group – the Windows group name to which the operators belong. You can enter up to 30 characters. During execution of an EWI, only users who are members of this group can electronically sign the EWI as "Performed By" and enter operator comments. This field is required.

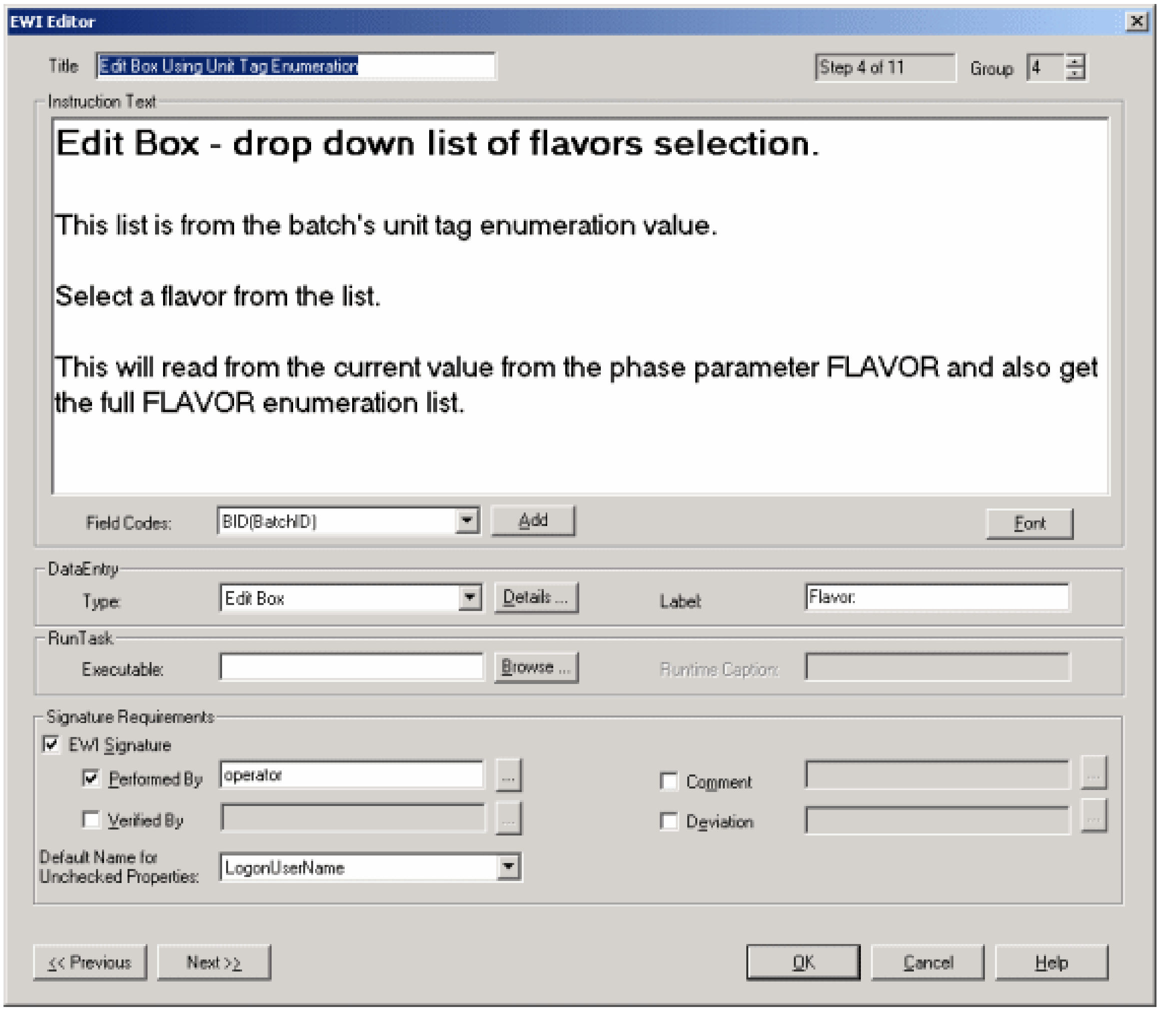
Verified By Group – the Windows group name to which the supervisors belong. You can enter up to 30 characters. During execution of an EWI, only users who are members of this group can electronically sign the EWI as "Verified By" and enter operator comments. This field is required.

Default Name for Unchecked Properties – the Windows computer name or login name that is captured by default if there is no performed by or verified group defined.

Deviation – if selected, describes the Windows group name to which the user performing the deviation must belong. If this option is *not* selected, the default name (computer name or login name) is captured, and the user is not prompted to enter a signature.

Comments – if selected, describes the Windows group name to which the user adding the comment must belong. If this option is *not* selected, the default name (computer name or login name) is captured, and the user is not prompted to enter a signature.

You can define each EWI in the WorkInstruction Editor from the EWI Editor window tab, as shown in the following figure.



Title: Edit Box Using Unit Tag Enumeration

Step 4 of 11

Group: 4

Instruction Text

Edit Box - drop down list of flavors selection.

This list is from the batch's unit tag enumeration value.

Select a flavor from the list.

This will read from the current value from the phase parameter FLAVOR and also get the full FLAVOR enumeration list.

Field Codes: BID(BatchID)

DataEntry

Type: Edit Box

Label: Flavor:

RunTask

Executable:

Runtime Caption:

Signature Requirements

☑ EWI Signature

☑ Performed By: operator

☐ Verified By:

☐ Comment

☐ Deviation

Default Name for Unchecked Properties: LogonUserName

EWI Editor Dialog Box

Understanding Signature Requirements and Security Groups

To verify operator actions during the execution of an EIB, EWI, or Logic Step, you are required to define the necessary electronic signatures required to complete the intended action. To configure an electronic signature, you must define the level of authorization, referred to as the signature requirements, and identify which group of operators you want to authorize to sign each intended action in the Windows security groups field.

There are three types of authorization:

- **None:** no signature required. The Windows computer name or login name is captured, by default.
- **Performed By:** a single signature required, verified against Performed By group.
- **Perform By/Verified By:** two distinct signatures required; first verified against Performed By Group, and second against the Verified By Group.

Security for signing of instructions is provided using domain or local node operating system security, on a group basis. Regardless of the signature requirements, you must define both a Performed By and

Verified By security group for the EIB header as well as for each EWI, since operator comments and deviations in data entry also require valid authorization.

Signature Requirements for Comments

To sign a comment, an operator must be a member of the comment security group. If a user enters comments from the EIB run-time screen, than the operator must be a member of the comment security group defined in the EIB security groups. If a user enters comments while processing an EWI, the operator must be a member of the EWI comment security group.

Signature Requirements for Deviations

To sign a deviation, either a manual or automatic deviation, an operator must be a member of the deviation security group. If the deviation is generated during the verification of data entry within an EWI, then the operator must be a member of the deviation security group defined in the EWI security groups. Operators can also force a deviation at either the EIB level or EWI level. In either case, the operator must be a member of the appropriate group.

Dual Signature Constraints

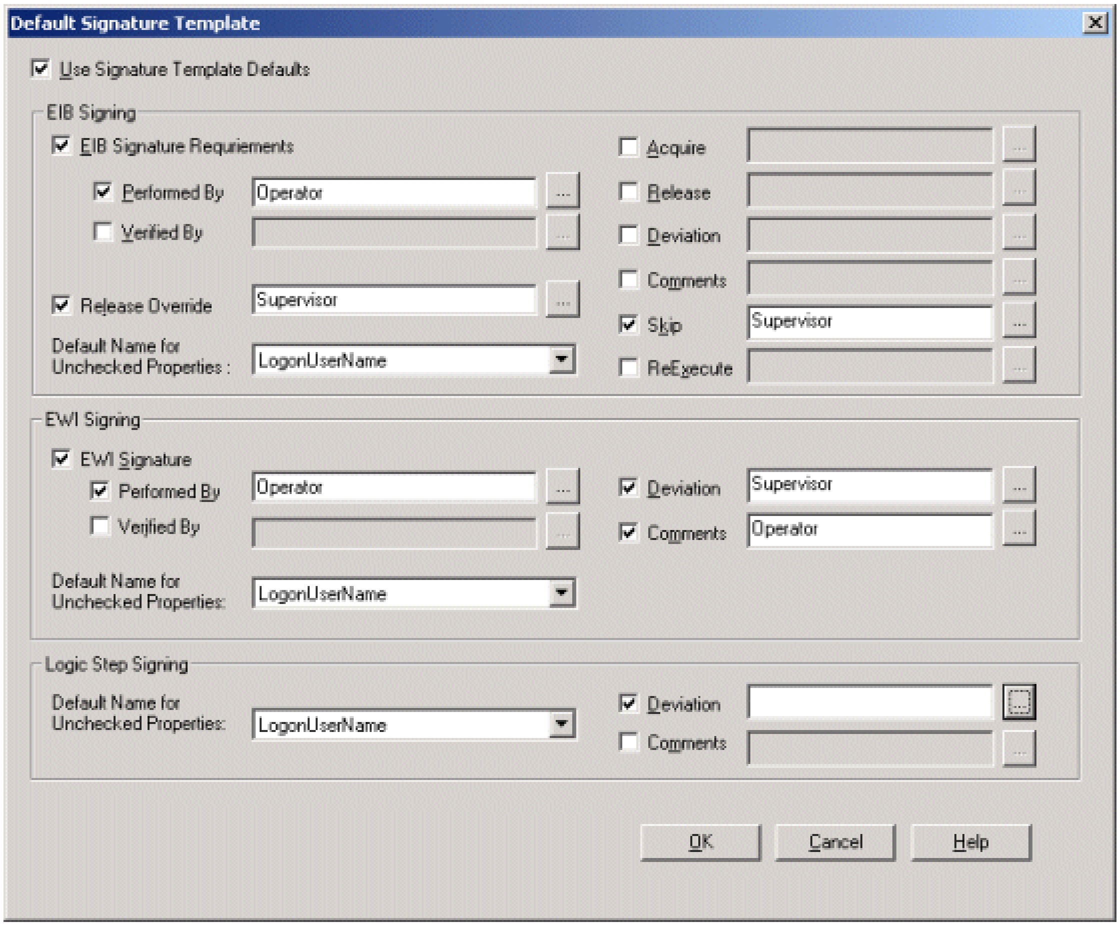
When two distinct signatures are required, you need to select the Performed By/Verified By option. Regardless of the location of the groups (local or domain), the two accounts must have different full user names. This is determined by comparing the Windows full user name of both signatures.

NOTE: *Windows security allows user accounts to be created without a full user name defined. WorkInstruction does not accept the signature if you did not define a full user name. You must define a valid full user name for all operator accounts to be used with the WorkInstruction software.*

Default Signature Template

With WorkInstruction, there is a Default Signature Template dialog box that you can use to automatically populate the signature requirements and security groups for each EWI, Logic Step, and EIB that you author in the WorkInstruction Editor. You can use these defaults or assign other requirements as you author new EIBs or EWIs. By allowing you to set defaults for authoring EIBs, EWIs, and Logic Steps, the template assists you in creating EIBs and EWIs faster.

The following figure shows an example of the Default Signature Template dialog box in the WorkInstruction Editor.



Default Signature Template

☑ Use Signature Template Defaults

EIB Signing

☑ EIB Signature Requriements

☑ Performed By: Operator

☐ Verified By:

☑ Release Override: Supervisor

Default Name for Unchecked Properties : LogonUserName

☐ Acquire

☐ Release

☐ Deviation

☐ Comments

☑ Skip: Supervisor

☐ ReExecute

EWI Signing

☑ EWI Signature

☑ Performed By: Operator

☐ Verified By

Default Name for Unchecked Properties: LogonUserName

☑ Deviation: Supervisor

☑ Comments: Operator

Logic Step Signing

Default Name for Unchecked Properties: LogonUserName

☑ Deviation

☐ Comments

Default Signature Dialog Box

Steps to Configure EIBs, EWIs, and Logic Steps

Once you understand what comprises the EIB, EWIs, and Logic Steps, and how you define signature requirements and set defaults, you can begin configuring your EIBs, EWIs, and Logic Steps.

▸To create a header for your EIB in the WorkInstruction Editor:

1. Click the Start button, and point to Programs, Proficy Batch Execution, and then WorkInstruction Editor.

   NOTE: *You can optionally launch the WorkInstruction Editor from the Recipe Editor.*

2. On the File menu, click New. The EIB Header dialog box appears
3. Enter information in each of the fields.
4. Click OK.

▶To configure your EIBs and EWIs in the WorkInstruction Editor:

1. With the EIB that you just created, double-click a row to add a step. The Step Selection dialog box opens.
2. Select EWI or Logic Step.
3. Enter required information.
4. Repeat steps 1-3 for each EWI you want to add to that EIB.
5. On the EIB menu, click Verify. The Verify Status Results window appears.
6. Click OK when the verification is complete.
7. On the File menu, click Save As. The Save As dialog box appears.
8. Enter a name for the XML file, and then click Save.

▶To check your files into the Work project and then move them into Production:

1. On the DMS menu in the WorkInstruction Editor, click DMS. The WorkInstruction Document Management dialog box appears.
2. Click Add to add the file to the Work Project. The Open dialog box appears.
3. Select an XML file that you previously saved, and then click Open.

   NOTE: *If the Use Signature Verification for DMS Commands option is enabled in the Database Info for Audit Trail dialog box, you will be prompted to enter a user name and password (from the iWorkUsers Windows security group) when you attempt to perform any SourceSafe functions.*

4. Click the file that you just added to the Work folder.
5. Click the Copy to Production button. A message box appears, prompting you to confirm the move.
6. Click Yes to continue. The EIB Label dialog box appears.
7. Enter a label or leave the default.
8. Click OK.
9. Repeat for steps 2-8 for each EIB that you want to add to your project.

   NOTE: *If you edit an existing EWI or Logic Step instead of adding a new one, and want to run that step the next time you launch the EWI ActiveX control, you need to go back to the Recipe Editor to change the version of the step being used. To always run the most current EWI or Logic Step in the EWI ActiveX control (if your company is less strict with version control), you can select Use Latest Version option in the Recipe Editor.*

Adding Field Codes to Instructions

In WorkInstruction, you can create your own field codes to represent:

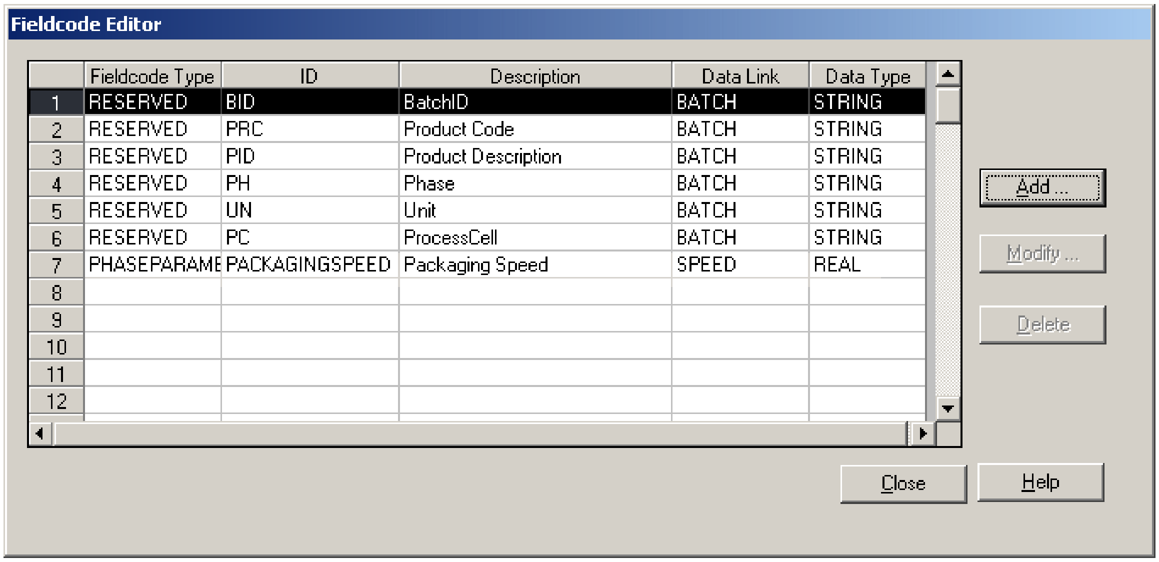
- Unit Tags (read/write)
- Phase Parameters (read/write)
- Report Parameters

- SQL Statements (Select Statements Only)
- Result of Another Step
- Hyperlinks
- Other User Defined Field Codes

Field codes can be used in the instruction text or data entry fields within an EWI or within logic instructions for a specific EIB. The data entry fields include radio buttons and edit boxes (for IVO and IDO data) that you create in the WorkInstruction Editor.

You add field codes though the Field Code Editor. There are six pre-defined field codes that you can also use: Batch ID, Product Code, Product Description, Phase, Unit, and Process Cell. Once the field codes are defined for an EIB, you can use the field codes in any EWI within that EIB.

For example, the EWI demo uses a field code named PACKAGINGSPEED to represent a real data value, as shown in the following figure.



	Fieldcode Type	ID	Description	Data Link	Data Type
1	RESERVED	BID	BatchID	BATCH	STRING
2	RESERVED	PRC	Product Code	BATCH	STRING
3	RESERVED	PID	Product Description	BATCH	STRING
4	RESERVED	PH	Phase	BATCH	STRING
5	RESERVED	UN	Unit	BATCH	STRING
6	RESERVED	PC	ProcessCell	BATCH	STRING
7	PHASEPARAME	PACKAGINGSPEED	Packaging Speed	SPEED	REAL
8					
9					
10					
11					
12					

Field Code Editor

By defining the packing speed as a field code, you can easily change the data type or link in the Field Code Editor, without having to edit each individual EWI. More importantly, however, is the fact that you can also use this field code in any EWI within the EIB named, PACKING_COMPLEX.XML (included with the demo).

When configuring a unit tag, do *not* use the name of the unit tag. Make sure that you use the unit tag's class name. For example, say you had three unit tags named Speed1, Speed2, and Speed3 which were all being used on different units, but were all from the same unit tag class, Speed. In this example, you would use Speed when you configure the unit tag. Since you don't always know which unit may be used for the class-based recipe, this ensures that the appropriate unit tag is selected for the designated unit when the EIB executes.

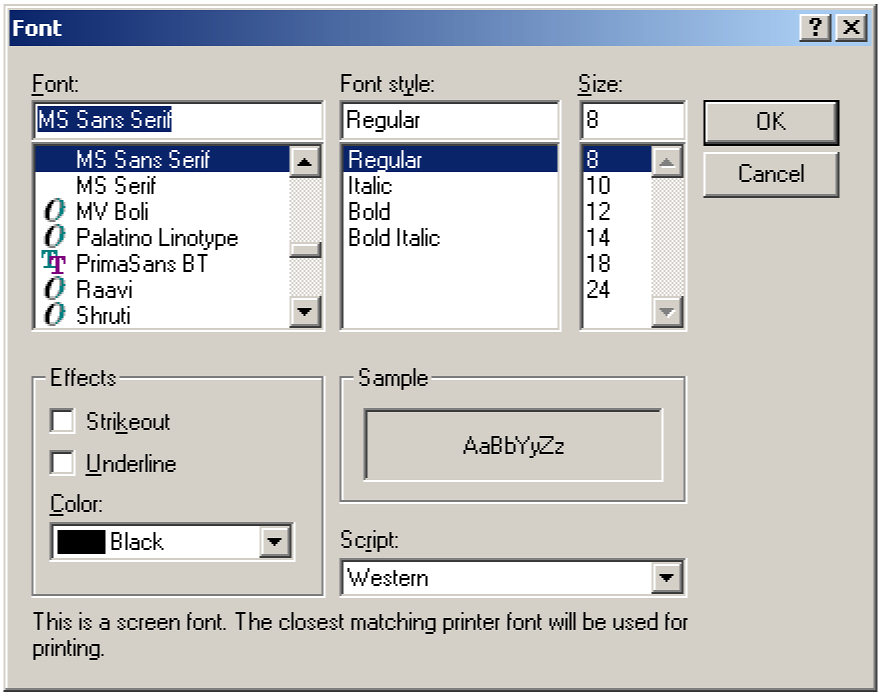
Field Code Types

The following table describes the content of the data link and data type fields based on the type of field code you are defining.

Data Links and Data Types for Field Codes		
Type of Field Code	**Data Link**	**Data Type**
Unit Tag	Unit tag class name as defined in the area model.	Same type as defined in area model for the unit tag (Real, String, Integer, or Enumeration).
Phase Parameter	Step parameter name as defined for a step (phase) within an Unit Operation recipe.	Same type as defined in step parameter list (Real, String, Integer, or Enumeration).
Report Parameter	Phase class report name as defined in the equipment phase class within the area model.	Same type as defined in the area model for the report (Real, String, or Integer).
User Defined	The value you want the field code to be substituted with at run time.	String, Real, or Integer only.
SQL Statement	Valid SQL statement. This statement can include parameters that are field codes.	SQL Statement that returns a single column of data. If more than 1 row is returned, the EWI ActiveX control displays this as a list. All SQL data is returned as Strings.
Hyperlink	Any valid hyperlink.	String.
Step	May be the one of 3 values: State, Value or Iterations. For example, "1.Value" refers to the value of EWI Step 1.	State - String Value – Must be the same as the data type defined for the EWI data entry. Iterations - Integer.

Changing the Font in the Instruction Text

To change the font of all or part of the instructional text, select the text and click on the Font button below the Instruction Text edit box. The Font dialog box displays, as shown in the following figure. Make your selections and click OK to change the font.



Font: MS Sans Serif

Font style: Regular

Size: 8

□ Strikeout

□ Underline

Color: Black

Script: Western

This is a screen font. The closest matching printer font will be used for printing.

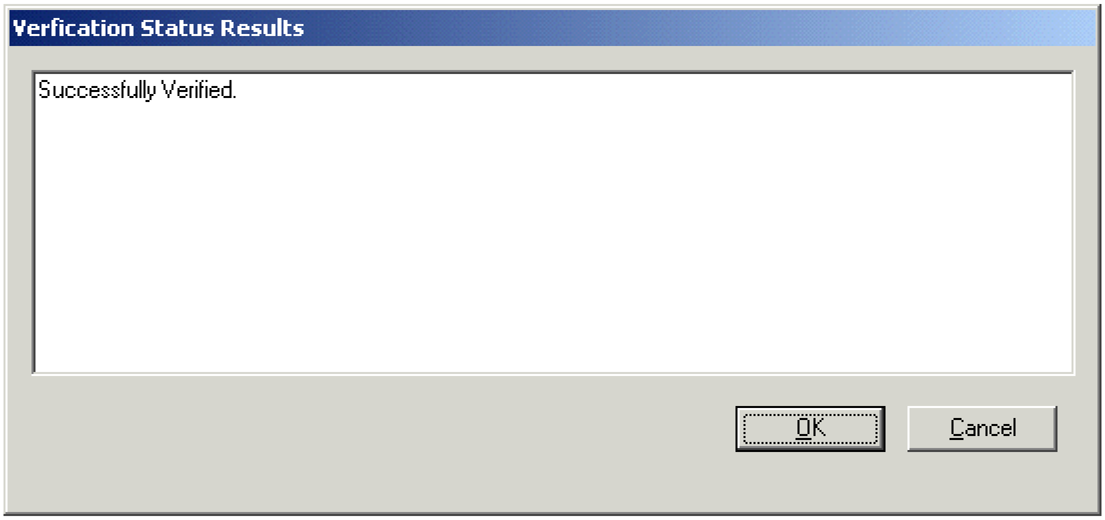
Font Dialog Box

Formatting the Font for Field Code Tags

Although the WorkInstruction Editor allows you to format text within a tag, do not format only part of the tag text. Either format the whole tag, or none of the tag. For instance, %UNITTAG.LEVEL% is a tag. You can bold this tag by selecting all of it including the % symbols, clicking the Font button, and selecting bold. Thus, it appears as **%UNITTAG.LEVEL%** in the instructional text. However, if you only bolded part of the tag, it will not work; the EIB Server reads the partially formatted tag wrong at run time. For instance, if you only bolded the word Level, %UNITTAG.**LEVEL**%, the tag would not be read correctly.

Step Verification

Just as you can verify recipes in the Batch Execution Recipe Editor, the WorkInstruction Editor now has the capability to verify EWI or Logic steps, as shown in the following figure.



Successfully Verified.

Verification Status Results

You must verify every EWI or Logic Step that you create before you can use it in production.

Configuring Phases for EWIs

To configure phases for EWIs, Batch Execution provides an OPC Soft Phase Server. Through the Soft Phase Server, you can send electronic work instructions to the operator with a 7001 request.

A *normal* phase can also trigger an EIB using a 7001 request. If you do not use the Soft Phase Server, you have to program the PLI and phase logic manually to handle the 7001 request.

For information on setting up the Soft Phase Server refer to the Configuring the Soft Phase Server section in the System Configuration Manual.

Configuring Phases in the Batch Execution Equipment Editor

In the Batch Execution Equipment Editor, you need to create a phase class for the EWI and soft phase. Create an instance of this phase for each unit where you want to run EWIs.

If you are making a class-based recipe, the phase instance must be in *every* unit for the phase to be available to an operation. If you are using the Soft Phase Sever, be sure to select Intellution.SPServer as the server for the Equipment Phase Tags.

Configuring Steps in the Batch Execution Recipe Editor

You need to assign EIBs to individual steps in a recipe at the operation-level. The operation may be a part of a procedure or unit procedure. It can also be run as an operation by itself.

When you run the recipe, ensure that the Release to Production is checked and that the recipe is verified.

In the Batch Execution Recipe Editor, you can also select to Use Latest Version of an EIB (Electronic Instruction Block), instead of indicating a specific version to use. With this feature, you always use the most recent EIB version at run time. Otherwise, you must specify the version of the EIB that you want the recipe to run.

You can access the EIB version information when you right-click a unit procedure or operation in the Recipe Editor and then select EWI from the right-click menu. The Use Latest Version option appears in the Version drop-down list.

Configuring iFIX with WorkInstruction

To configure WorkInstruction with iFIX, create a new iFIX picture for the EWI ActiveX control. The EWI ActiveX control is fixed in size, so it may be the only object in the picture. If you are running iFIX on a different computer than the Batch Execution Servers, configure the property of the EWI ActiveX control to point to the server computer.

▶To configure the EIB Server settings for the EWI ActiveX control:

1. Right-click the control and select Properties from the pop-up menu. The control's property pages appear.
2. Select the EWI Server property page.
3. Select Remote, since the EIB Server is running on a different computer from the control.
4. Click the Browse button to select the computer name on which the EIB Server is running. You can also manually enter the computer name, IP address, or other DCOM-compatible computer name.
5. Select the Connect on Startup check box if you want the control to automatically connect to the EIB Server when the control is instantiated in a run-time environment.
6. In the iFIX property window for the EWI ActiveX control, set the IsSelectable property to True so that the operator can use the Tab key within the EWI control.
7. Click OK to save your changes.

Optionally, create another iFIX screen with the Batch List Control. This eliminates the need for the Batch Execution Client.

NOTE: *EWIs can only be viewed from the EWI ActiveX control and not from the Batch Execution Client.*

Configuring the EIB Server to Start from the Command Line

The default way to run the EIB Server is to launch the EIB Server Manager from the Start menu, and select a boot method (Cold or Warm) when prompted. However, you can also configure the boot method for the EIB Server on the Restart Control tab of the Batch Configuration dialog box. You then can start the EIB Server automatically from the command line, using the boot method you configured.

The command line that you run to start the EIB Server can exist in a script, so that it runs automatically, or you can launch it from the Run dialog box.

▶To configure how the EIB Server starts:

1. If the EIB Server or the Batch Server is running, shut it down.
2. Open the Proficy Batch Execution WorkSpace.
3. In the Configuration folder of your project, double-click the Batch Execution Configuration entry. The Batch Configuration dialog box appears.
4. Select the Restart Control tab.
5. In the EIB Server Restart Control group box, select COLD as the restart type.
6. Click OK.
7. Save the project.

▶To start the EIB Server from the command line:

1. Restart the Batch Server.
2. Enter the following command into the Run dialog box, or into a script that you develop to launch the EIB Server:

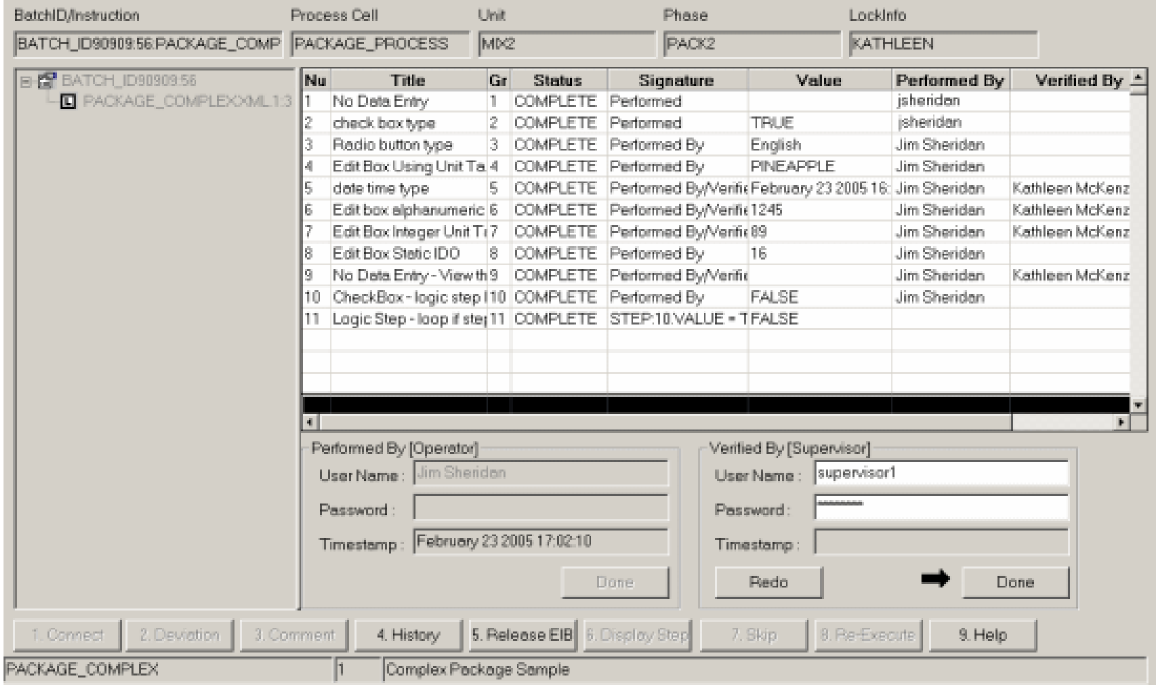
   ```
   "C:\Program Files\Proficy\Proficy Batch
   Execution\Bin\eibServerManager.exe" /autostart /minimized
   ```

***NOTE:** If you want the EIB Server Manager to appear on screen (maximized), do not use the /minimized option.*

The EIB Server Manager starts, which in turn starts the EIB Server. The EIB Server Manager dialog box will be minimized, unless you run the command without the /minimized option.

Configuring the EWI ActiveX Control

The "Intellution EWI Control" provides a display screen for operators using electronic work instructions. Developers can configure the run-time appearance and functionality of the user interface for the Electronic Work Instruction (EWI) ActiveX control using the control's property pages. The following figure illustrates the EWI ActiveX control.



BatchID/Instruction	Process Cell	Unit	Phase	LockInfo
BATCH_ID90909:56:PACKAGE_COMP	PACKAGE_PROCESS	MIX2	PACK2	KATHLEEN

BATCH_ID90909:56
PACKAGE_COMPLEX.XML:1:3

Nu	Title	Gr	Status	Signature	Value	Performed By	Verified By
1	No Data Entry	1	COMPLETE	Performed		jsheridan	
2	check box type	2	COMPLETE	Performed	TRUE	jsheridan	
3	Radio button type	3	COMPLETE	Performed By	English	Jim Sheridan	
4	Edit Box Using Unit Ta	4	COMPLETE	Performed By	PINEAPPLE	Jim Sheridan	
5	date time type	5	COMPLETE	Performed By/Verifie	February 23 2005 16:	Jim Sheridan	Kathleen McKenz
6	Edit box alphanumeric	6	COMPLETE	Performed By/Verifie	1245	Jim Sheridan	Kathleen McKenz
7	Edit Box Integer Unit T	7	COMPLETE	Performed By/Verifie	89	Jim Sheridan	Kathleen McKenz
8	Edit Box Static IDO	8	COMPLETE	Performed By	16	Jim Sheridan	
9	No Data Entry - View th	9	COMPLETE	Performed By/Verifie		Jim Sheridan	Kathleen McKenz
10	CheckBox - logic step l	10	COMPLETE	Performed By	FALSE	Jim Sheridan	
11	Logic Step - loop if step	11	COMPLETE	STEP:10.VALUE = T	FALSE		

Performed By [Operator]

User Name : Jim Sheridan

Password :

Timestamp : February 23 2005 17:02:10

Verified By [Supervisor]

User Name : supervisor1

Password : ********

Timestamp :

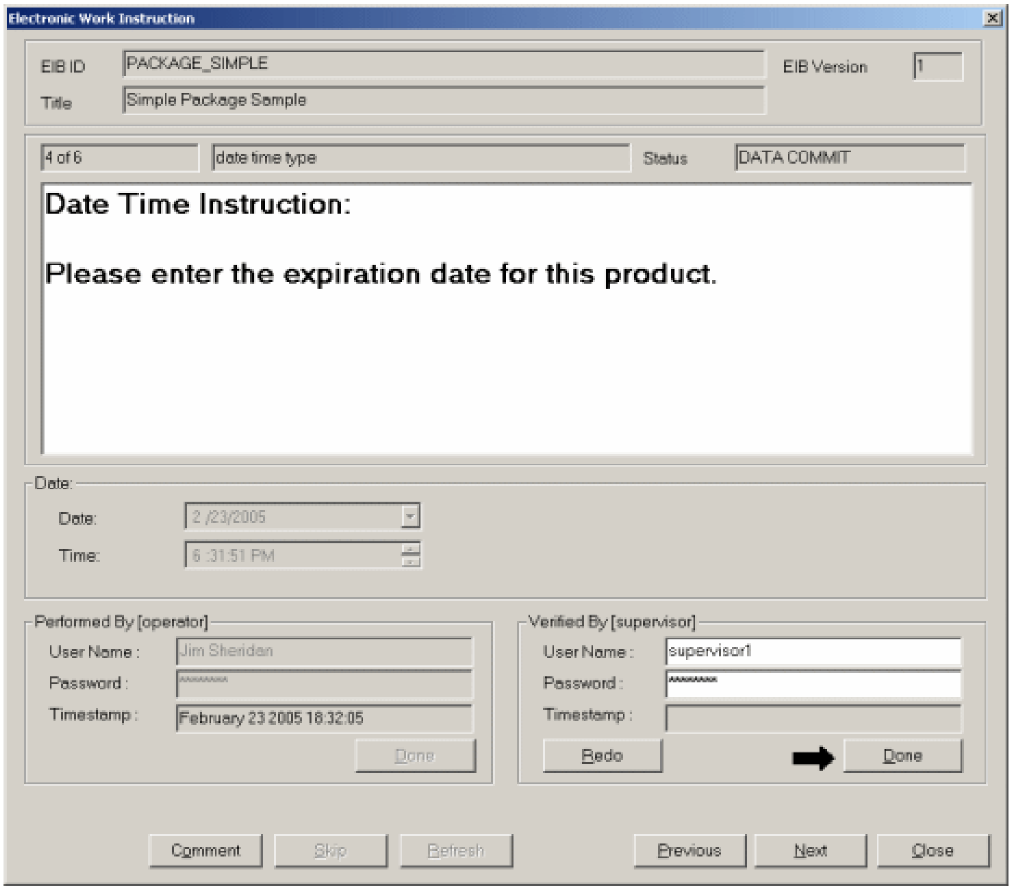
PACKAGE_COMPLEX | 1 | Complex Package Sample

EWI ActiveX Control

Developers can configure the EWI ActiveX control programmatically through the Proficy iFIX WorkSpace, Visual Basic, or Visual C++ using the control's properties and methods. The properties, methods, and events for the EWI control are described in the following sections.

EWI Display Screen

When an operator opens an individual EWI (that you developed) from the EWI ActiveX control in run-time, the specific EWI displays. You configure the information, such as the data entry, signature, run tasks, and Windows security group requirements, that displays in this screen through the WorkInstruction Editor. An example of an individual EWI is shown in the following figure.



Electronic Work Instruction

EIB ID: PACKAGE_SIMPLE

EIB Version: 1

Title: Simple Package Sample

4 of 6 | date time type | Status: DATA COMMIT

Date Time Instruction:

Please enter the expiration date for this product.

Date:

Date: 2 /23/2005

Time: 6 :31:51 PM

Performed By [operator]

User Name : Jim Sheridan

Password : ********

Timestamp : February 23 2005 18:32:05

Verified By [supervisor]

User Name : supervisor1

Password : ********

Timestamp :

Sample EWI Display

The sections that follow describe how to use the EWI ActiveX control. These sections do not describe the EWI display portion, shown in the sample EWI display figure. To learn more about how to configure individual EWIs in the WorkInstruction Editor, refer to the Building Electronic Work Instructions in the WorkInstruction Editor section.

Overview of the Batch Execution EWI ActiveX Control

You can integrate the EWI control into custom applications. Each EWI control provides programmability through OLE Automation to take advantage of the control's features through Microsoft's Visual Basic and Visual C++ programming languages, or any other COM-compatible development environment.

ActiveX Control Modes

The EWI ActiveX control runs as a client of the EIB Server. The EIB Server is a client of VBIS. There are two modes of use for the EWI ActiveX control: run time and design time. Run time occurs when the control is in a run-time container and interacting with the EIB Server. At run time, there must be a connection to the EIB Server, either locally or remotely over a network. Design time occurs when a developer configures the properties of the control. The control may be embedded in a control container or included as part of an application. A connection to the EIB Server is not necessary during design time.

OLE Containers

You can embed the ActiveX controls in any OLE-compliant container, such as:

- Proficy iFIX WorkSpace
- Web browsers, such as Internet Explorer
- Visual Basic
- Visual C++
- Batch Execution WorkSpace

Property Pages

The EWI control provides property pages that designers or operators can use to view and change the control's properties.

EWI ActiveX Control Properties

The following sections describe each property for the EWI control. The properties are grouped by the following functions:

- EWI Column properties
- EWI Filter properties
- EWI Security properties
- EWI Server properties
- EWI Command Buttons properties
- EWI Miscellaneous properties
- EWI Fonts properties
- EWI Colors properties

EWI Column Properties

The EWI Column properties are not displayed on the property pages. They can be set in the design time property list provided by Visual Basic and the Proficy iFIX WorkSpace. These properties can also be set at run time in a script during the loading of the form or picture. At run time you can also use the mouse to drag the column separators on the EWI spreadsheet to change the width of a column. However, once you close your application and then restart it, your changes are lost unless you write a script in VBA to save the column width values when the form or picture closes and use them when the form or picture loads.

You cannot graphically set the column width fields while in configure/design mode within iFIX or Visual Basic. However, you can graphically set these fields in design mode from the ActiveX Control Test Container and the Batch Execution Workspace.

The following table lists the properties that control the column settings for the EWI control.

EWI Column Filter Properties	
Property	**Syntax**
EWIGroupWidth Width of the EWI group column.	**C++ Syntax:** **double** CEWIX::GetEWIGroupWidth(); **void** CEWIX::SetEWIGroupWidth(double); **Visual Basic Syntax:** [*form*.]*Control*.EWIGroupWidth [= *value!*]
EWINumberWidth Width of the EWI number column.	**C++ Syntax:** **double** CEWIX::GetEWINumberWidth(); **void** CEWIX::SetEWINumberWidth(double); **Visual Basic Syntax:** [*form*.]*Control*.EWINumberWidth [= *value!*]
EWISignatureWidth Width of the EWI signature column.	**C++ Syntax:** **double** CEWIX::GetEWISignatureWidth(); **void** CEWIX::SetEWISignatureWidth(double); **Visual Basic Syntax:** [*form*.]*Control*.EWISignatureWidth [= *value!*]
EWIStatusWidth Width of the EWI status column.	**C++ Syntax:** **double** CEWIX::GetEWIStatusWidth(); **void** CEWIX::SetEWIStatusWidth(double); **Visual Basic Syntax:** [*form*.]*Control*.EWIStatusWidth [= *value!*]
EWITitleWidth Width of the EWI title column.	**C++ Syntax:** **double** CEWIX::GetEWITitleWidth(); **void** CEWIX::SetEWITitleWidth(double); **Visual Basic Syntax:** [*form*.]*Control*.EWITitleWidth [= *value!*]
EWIValueWidth Width of the EWI value column.	**C++ Syntax:** **double** CEWIX::GetEWIValueWidth(); **void** CEWIX::SetEWIValueWidth(double); **Visual Basic Syntax:** [*form*.]*Control*.EWIValueWidth [= *value!*]

EWI Column Filter Properties	
Property	**Syntax**
EWIPerformedByWidth Width of the EWI performed by column.	**C++ Syntax:** **double** CEWIX::GetEWIPerformedByWidth(); **void** CEWIX::SetEWIPerformedByWidth(double); **Visual Basic Syntax:** [*form*.]*Control*.EWIPerformedByWidth [= *value!*]
EWIVerifiedByWidth Width of the EWI verified by column.	**C++ Syntax:** **double** CEWIX::GetEWIVerifiedByWidth(); **void** CEWIX::SetEWIVerifiedByWidth(double); **Visual Basic Syntax:** [*form*.]*Control*.EWIVerifiedByWidth [= *value!*]

Visual Basic Example

```
EWIX1.EWIGroupWidth = 2.5
EWIX1.EWINumberWidth = 4.5
EWIX1.EWISignatureWidth = 20
EWIX1.EWIStatusWidth = 15
EWIX1.EWITitleWidth = 15.75
EWIX1.EWIValueWidth = 20
EWIX1.EWIPerformedByWidth = 20
EWIX1.EWIVerifyByWidth = 20
```

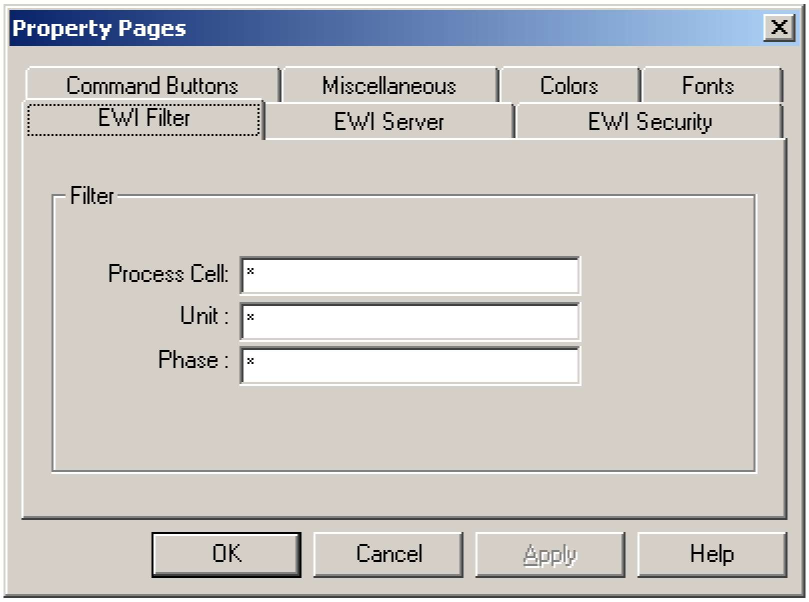
C++ Example

```
EWIX1.SetEWIGroupWidth(2.5);
EWIX1.SetEWINumberWidth(4.5);
EWIX1.SetEWISignatureWidth(20);
EWIX1.SetEWIStatusWidth(15);
EWIX1.SetEWITitleWidth(15.75);
EWIX1.SetEWIValueWidth(20);
EWIX1.SetEWIPerformedByWidth(20);
EWIX1.SetEWIVerifyByWidth(20);
```

EWI Filter Properties

The Filter property page lets you configure how the data is filtered in the control. The Filter property page is available at design time and at run time.

The following figure shows the Filter property page for the EWI control.



Process Cell: *

Unit : *

Phase : *

EWI Filter Property Page

You can set filters using * and ? symbols for wild card matches. The * symbol searches for multiple unknown characters in a word or phrase. The ? symbol searches for a single unknown character in a word. You can also use the | symbol to specify multiple OR conditions. The \ escape character followed by the \, *, or ? symbol allows you to search with the actual `\, *, or ? symbol. Pattern matching is case sensitive. The following table shows examples of what you can enter as a pattern or string when looking for a certain process cell, unit, or phase, and whether a match can be found.

Pattern Matching Examples for the EWI Filter Tab

Pattern You Enter...	**String You Are Looking for...**	**Is There a Match?**
batch1	Batch1	No
?atch1	Batch1	Yes
B*	Batch1	Yes
B*1	Batch1	Yes
B*1	Batch2	No
tch	Batch1	Yes
tch	Batch	Yes
*tch?	Batch	No

Pattern Matching Examples for the EWI Filter Tab		
Pattern You Enter...	**String You Are Looking for...**	**Is There a Match?**
Batch?	Batch1	Yes
Batch?	Batch2	Yes
B*\\Week?	BatchWeek1	No
B*\\Week?	Batch\Week1	Yes
Batch*One	Batch*One	Yes
Batch\?One	Batch?One	Yes

The following table lists the properties that control the filter settings for the EWI control.

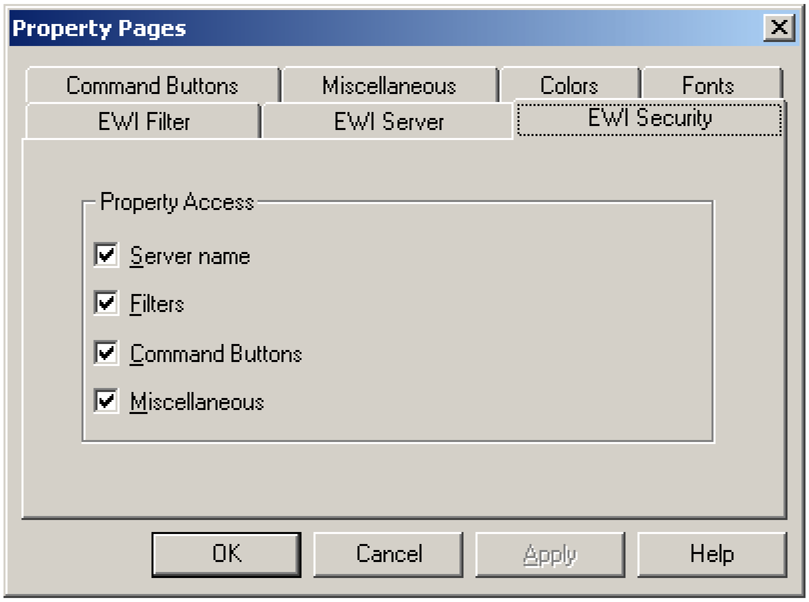
EWI Control Filter Properties	
Property	**Syntax**
ProcessCellFilter Sets the filter for the Process Cells.	**C++ Syntax:** **CString** CEWIX::GetProcessCellFilter(); **void** CEWIX::SetProcessCellFilter(**LPCTSTR** value); **Visual Basic Syntax:** [*form.*]*Control*.ProcessCellFilter[= *text$*]
UnitFilter Sets the filter for the Units.	**C++ Syntax:** **CString** CEWIX::GetUnitFilter(); **void** CEWIX::SetUnitFilter(**LPCTSTR** value); **Visual Basic Syntax:** [*form.*]*Control*.UnitFilter[= *text$*]
PhaseFilter Sets the filter for the Phases.	**C++ Syntax:** **CString** CEWIX::GetPhaseFilter(); **void** CEWIX::SetPhaseFilter(**LPCTSTR** value); **Visual Basic Syntax:** [*form.*]*Control*.PhaseFilter[= *text$*]

EWI Security Properties

The Security property page lets you configure the properties that the run-time operator can access in the control. The Security property page is available only at design time. The following are design-time containers:

- Visual Basic
- Visual C++
- Proficy iFIX WorkSpace

The following figure shows the Security property page for the EWI control.



Property Access

☑ Server name

☑ Filters

☑ Command Buttons

☑ Miscellaneous

EWI Security Property Page

The following table lists the properties that control the security settings for the EWI control.

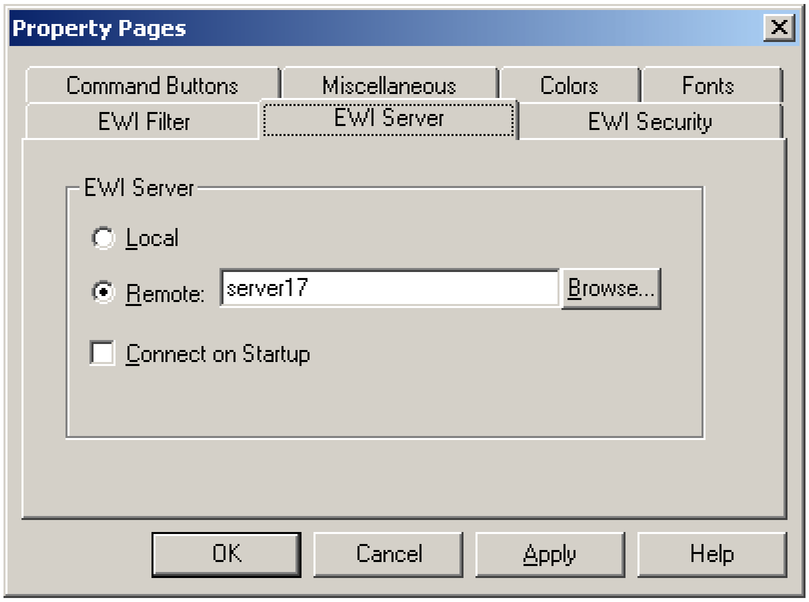
EWI Control Security Properties	
Property	**Syntax**
ServerEditEnabled Sets whether or not the operator can edit the EIB Server name for the control.	**C++ Syntax:** **BOOL** CEWIX::GetServerEditEnabled(); **void** CEWIX::SetServerEditEnabled(**BOOL** *value*); **Visual Basic Syntax:** [*form.*]*Control.*ServerEditEnabled [= *boolvalue*]
FilterEditEnabled Sets whether or not the operator can edit the filters for the EWI control.	**C++ Syntax:** **BOOL** CEWIX::GetFilterEditEnabled(); **void** CEWIX::SetFilterEditEnabled(**BOOL** *value*); **Visual Basic Syntax:** [*form.*]*Control.*FilterEditEnabled [= *boolvalue*]
MiscEditEnabled Sets whether or not the operator can edit the miscellaneous properties for the EWI control.	**C++ Syntax:** **BOOL** CEWIX::GetMiscEditEnabled(); **void** CEWIX::SetMiscEditEnabled(**BOOL** *value*); **Visual Basic Syntax:** [*form.*]*Control.*MiscEditEnabled [= *boolvalue*]
CommandBtnsEditEnabled Sets whether or not the operator can edit the command buttons displayed for the EWI control.	**C++ Syntax:** **BOOL** CEWIX::GetCommandBtnsEditEnabled(); **void** CEWIX::SetCommandBtnsEditEnabled(**BOOL** *value*); **Visual Basic Syntax:** [*form.*]*Control.*CommandBtnsEditEnabled [= *boolvalue*]

EWI Server Properties

The EWI Server property page:

- Lets you configure the EIB Server settings for the ActiveX control, such as whether the EIB Server is local or remote.
- Is available at design time and may be enabled at run time using the Security property page.

The following figure shows the Server property page for the EWI control.



EWI Server

○ Local

● Remote: server17

□ Connect on Startup

EWI Server Property Page

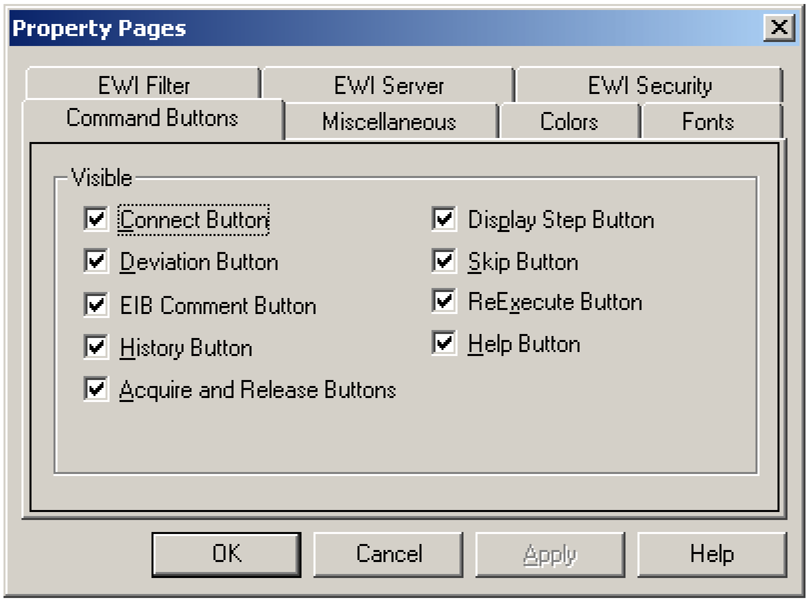
The following table lists the properties that control the EWI Server settings for the EWI control.

EWI Control Server Properties	
Property	**Syntax**
ConnectAtStartup If TRUE, the control connects to the EIB Server when instantiated at run-time.	**C++ Syntax:** **BOOL** CEWIX::GetConnectAtStartup(); **void** CEWIX::SetConnectAtStartup(**BOOL** *value*); **Visual Basic Syntax:** [*form.*]*Control*.ConnectAtStartup[= *boolvalue*]
ServerComputer The name of the EIB Server to connect to. If the string is empty, the local EIB Server is used.	**C++ Syntax:** **CString** CEWIX::GetServerComputer(); **void** CEWIX::SetServerComputer(**LPCTSTR** *value*); **Visual Basic Syntax:** [*form.*]*Control*.ServerComputer[= *text$*]

EWI Command Buttons Properties

The EWI Command Buttons property page allows you to select which buttons display on the EWI ActiveX control.

The following figure shows the Command Buttons property page for the EWI control.



Visible

☑ Connect Button

☑ Deviation Button

☑ EIB Comment Button

☑ History Button

☑ Acquire and Release Buttons

☑ Display Step Button

☑ Skip Button

☑ ReExecute Button

☑ Help Button

EWI Command Buttons Property Page

The following table lists the properties that control the display of command buttons in the EWI control.

EWI Control Command Buttons Properties

Property	**Syntax**
ConnectButton Sets whether or not to display the Connect button.	**C++ Syntax:** **BOOL** CEWIX::GetConnectButton (); **void** CEWIX::SetConnectButton (**BOOL** value); **Visual Basic Syntax:** [*form.*]*Control*.ConnectButton[= *boolvalue*]
DeviationButton Sets whether or not to display the Deviation button.	**C++ Syntax:** **BOOL** CEWIX::GetDeviationButton(); **void** CEWIX::SetDeviationButton(**BOOL** value); **Visual Basic Syntax:** [*form.*]*Control*.DeviationButton[= *boolvalue*]
EIBCommentButton Sets whether or not to display the EIB Comment button.	**C++ Syntax:** **BOOL** CEWIX::GetEIBCommentButton(); **void** CEWIX::SetEIBCommentButton(**BOOL** value); **Visual Basic Syntax:** [*form.*]*Control*.EIBCommentButton[= *boolvalue*]

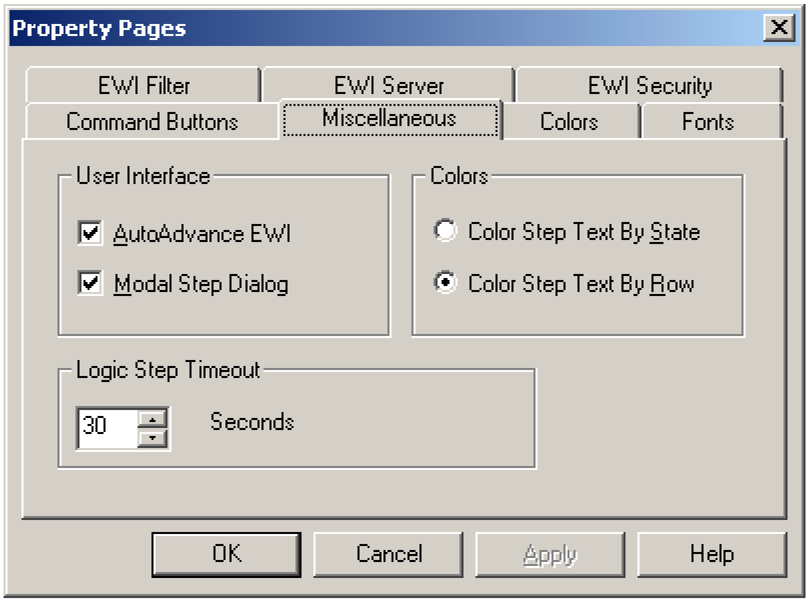
EWI Control Command Buttons Properties	
Property	**Syntax**
HistoryButton Sets whether or not to display the History button.	**C++ Syntax:** **BOOL** CEWIX::GetHistoryButton(); **void** CEWIX::SetHistoryButton(**BOOL** value); **Visual Basic Syntax:** [*form.*]*Control.*HistoryButton[= *boolvalue*]
AcquireReleaseButton Sets whether or not to display the Acquire/Release button.	**C++ Syntax:** **BOOL** CEWIX::GetAcquireReleaseButton(); **void** CEWIX::SetAcquireReleaseButton(**BOOL** value); **Visual Basic Syntax:** [*form.*]*Control.*AcquireReleaseButton[= *boolvalue*]
DisplayStepButton Sets whether or not to display the Display Step button.	**C++ Syntax:** **BOOL** CEWIX::GetDisplayStepButton(); **void** CEWIX::SetDisplayStepButton(**BOOL** value); **Visual Basic Syntax:** [*form.*]*Control.*DisplayStepButton[= *boolvalue*]
SkipStepButton Sets whether or not to display the Skip button.	**C++ Syntax:** **BOOL** CEWIX::GetSkipStepButton(); **void** CEWIX::SkipStepButton(**BOOL** value); **Visual Basic Syntax:** [*form.*]*Control.*SkipStepButton[= *boolvalue*]
ReExecuteStepButton Sets whether or not to display the Re-Execute button.	**C++ Syntax:** **BOOL** CEWIX::GetReExecuteStepButton(); **void** CEWIX::SetReExecuteStepButton(**BOOL** value); **Visual Basic Syntax:** [*form.*]*Control.*ReExecuteStepButton[= *boolvalue*]

EWI Control Command Buttons Properties	
Property	**Syntax**
HelpButton Sets whether or not to display the Help button.	**C++ Syntax:** **BOOL** CEWIX::GetHelpButton(); **void** CEWIX::SetHelpButton(**BOOL** value); **Visual Basic Syntax:** [*form.*]*Control*.HelpButton[= *boolvalue*]

EWI Miscellaneous Properties

The EWI Miscellaneous property page allows you to set certain miscellaneous properties for the EWI ActiveX control, such as the logic step time-out value.

The following figure shows the Miscellaneous property page for the EWI control.



☑ AutoAdvance EWI

☑ Modal Step Dialog

○ Color Step Text By State

● Color Step Text By Row

30 Seconds

EWI Miscellaneous Property Page

The following table lists the properties that control the display of miscellaneous properties in the EWI control.

EWI Control Miscellaneous Properties	
Property	**Syntax**
AutoAdvanceEWI When enabled, this property advances the EWI dialog box to next EWI after the required signatures are entered. The default is TRUE.	**C++ Syntax:** **BOOL** CEWIX::GetAutoAdvanceEWI(); **void** CEWIX::SetAutoAdvanceEWI (**BOOL** value); **Visual Basic Syntax:** [*form.*]*Control.*AutoAdvanceEWI[= *boolvalue*]
ModalStepDialog When enabled, the EWI and Logic Step dialog box are modal. Otherwise, these dialog boxes are modeless. The default is TRUE. For example, if you want to be able to access other iFIX pictures while the EWI ActiveX control displays the EWI or Logic Step dialog box, clear the Modal Step dialog property (to set the property to False).	**C++ Syntax:** **BOOL** CEWIX::GetModalStepDialog(); **void** CEWIX::SetModalStepDialog(**BOOL** value); **Visual Basic Syntax:** [*form.*]*Control.*ModalStepDialog[= *boolvalue*]
LogicStepTimeout The number of seconds the logic step loops before prompting the user of the potential infinite loop. The default is 30 seconds.	**C++ Syntax:** **short** CEWIX::GetLogicStepTimeout(); **void** CEWIX::SetLogicStepTimeout(**short** value); **Visual Basic Syntax:** [*form.*]*Control.*LogicStepTimeout[= *value%*]
ColorEWIGridText Represents a pair of radio buttons: • Color Step Text By State • Color Step Text By Row (the default) The Color property page describes the colors for the states and rows. The constants of these values are: EWIXColorStepTextByEWIState = 0 EWIXColorStepTextByOddEvenRow = 1	**C++ Syntax:** **short** CEWIX::GetColorEWIGridText(); **void** CEWIX::SetColorEWIGridText(**short** value); **Visual Basic Syntax:** [*form.*]*Control.*ColorEWIGridText[= *value%*]

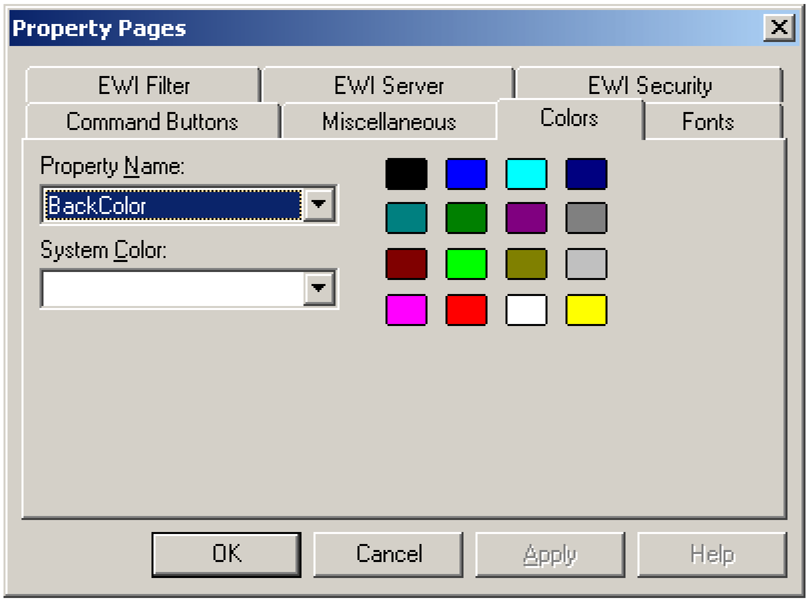
Example: ColorEWIGridText Property

If you write a Visual Basic script to set the colors of the EWI grid:

```
EWIX1.ColorEWIGridText = EWIXColorStepTextByEWIState

Dim ReadyColor As Long

ReadyColor = 65280     ' green

EWIX1.ReadyStateTextColor = ReadyColor
```

EWI Colors Properties

The Colors property page lets you set the control's colors, such as the color of the grid lines and the color of column header text. You can choose a color from the displayed color chart or you can use the System Color drop-down list box to assign a pre-defined system color to a property. The Colors property page, shown in the following figure, is available at design time and at run time.



Property Name: BackColor

System Color:

EWI Colors Property Page

The following table lists the properties that control the colors in the EWI control.

EWI Control Color Properties	
Property	**Syntax**
BackColor Sets the background color (the border around the edge) of the control.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetBackColor(); **void** CEWIX::SetBackColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*BackColor[= *color%*]
EvenRowBackColor Sets the background color of even rows in the data list. The default color is white.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetEvenRowBackColor(); **void** CEWIX::SetEvenRowBackColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*EvenRowBackColor[= *color%*]
EvenRowTextColor Sets the color of text in the even rows in the data list. The default color is black.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetEvenRowTextColor(); **void** CEWIX::SetEvenRowTextColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*EvenRowTextColor[= *color%*]
GridColor Sets the color of the grid lines in the data list. The default is light gray.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetGridColor(); **void** CEWIX::SetGridColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*GridColor[= *color%*]
HeaderBackColor Sets the background color for the column headers. The default color is gray.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetHeaderBackColor(); **void** CEWIX::SetHeaderBackColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*HeaderBackColor[= *color%*]

EWI Control Color Properties	
Property	**Syntax**
HeaderTextColor Sets the color for the header text. The default color is black.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetHeaderTextColor(); **void** CEWIX::SetHeaderTextColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*HeaderTextColor[= *color%*]
OddRowBackColor Sets the background color for odd rows in the data list. The default color is white.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetOddRowBackColor(); **void** CEWIX::SetOddRowBackColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*OddRowBackColor[= *color%*]
OddRowTextColor Sets the text color for odd rows in the data list. The default color is black.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetOddRowTextColor(); **void** CEWIX::SetOddRowTextColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*OddRowTextColor[= *color%*]
StateActiveTextColor Sets the text color for the Active state text. The default color is black.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetStateActiveTextColor(); **void** CEWIX::StateActiveTextColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*StateActiveTextColor[= *color%*]
StateCompleteTextColor Sets the text color for the Complete state text. The default color is black.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetStateCompleteTextColor(); **void** CEWIX::SetStateCompleteTextColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*StateCompleteTextColor[= *color%*]

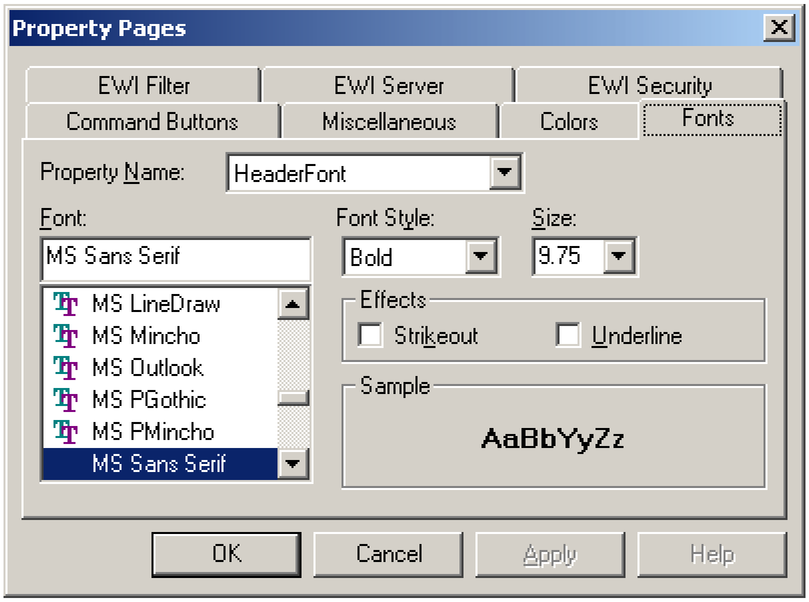
EWI Control Color Properties

Property	**Syntax**
StateDataEntryCompleteTextColor Sets the text color for the Data Entry Complete state text. The default color is black.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetStateDataEntryCompleteTextColor(); **void** CEWIX::SetStateDataEntryCompleteTextColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*StateDataEntryCompleteTextColor[= *color%*]
StateIdleTextColor Sets the text color for the Idle state text. The default color is black.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetStateIdleTextColor(); **void** CEWIX::SetStateIdleTextColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*StateIdleTextColor[= *color%*]
StateNotRunTextColor Sets the text color for the Not Run state text. The default color is black.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetStateNotRunTextColor(); **void** CEWIX::SetStateNotRunTextColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*StateNotRunTextColor[= *color%*]
StateReadyTextColor Sets the text color for the Ready state text. The default color is black.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetStateReadyTextColor(); **void** CEWIX::SetStateReadyTextColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*StateReadyTextColor[= *color%*]
StateSkippedTextColor Sets the text color for the Skipped state text. The default color is black.	**C++ Syntax:** **OLE_COLOR** CEWIX::GetStateSkippedTextColor(); **void** CEWIX::SetStateSkippedTextColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form.*]*Control.*StateSkippedTextColor[= *color%*]

EWI Control Color Properties	
Property	**Syntax**
DeviationValueTextColor Sets the text color for the value column text on the EWI grid display. The default color is black. For example, if you set this color to RED and if the value in the value column was a deviation from a normal value (for instance, the range is 1 to 10 and the user enters a value of 20), then the text would be colored red. Otherwise, it would be colored the same color as the other text for that row	**C++ Syntax:** **OLE_COLOR** CEWIX::GetDeviationValueTextColor(); **void** CEWIX::SetDeviationValueTextColor(**OLE_COLOR** *value*); **Visual Basic Syntax:** [*form*.]*Control*.DeviationValueTextColor[= *color%*]

EWI Fonts Properties

The Fonts property page lets you set the font attributes for the text in the column headers and the data list. The Fonts property page, shown in the following figure, is available at design time and run time.



Property Name: HeaderFont

Font: MS Sans Serif

Font Style: Bold

Size: 9.75

☐ Strikeout ☐ Underline

EWI Fonts Property Page

The following table lists the properties that control the fonts in the EWI control.

EWI Control Font Properties	
Property	**Syntax**
HeaderFont Sets the font of the text in the headers. The default font is MS Sans Serif, Bold 9.75.	**C++ Syntax:** **COleFont** CEWIX::GetHeaderFont(); **void** CEWIX::SetHeaderFont(**LPDISPATCH** *value*); **Visual Basic Syntax:** [*form.*]*Control.*HeaderFont[= *stdfontvariable*]
TextFont Sets the font of the text in the data list. The default font is MS Sans Serif, Bold 9.75.	**C++ Syntax:** **COleFont** CEWIX::GetTextFont(); **void** CEWIX::SetTextFont(**LPDISPATCH** *value*); **Visual Basic Syntax:** [*form.*]*Control.*TextFont[= *stdfontvariable*]

EWI ActiveX Control Methods

The EWI control exposes the following methods:

- AcquireEIB Method
- Comment Method
- ConnectToServer Method
- Deviation Method
- DisplayEWI Method
- History Method
- ReleaseEIB Method
- SuperUserReleaseEIB Method
- SkipStep Method
- ReExecuteEWI Method
- SelectFirstEIBInTree Method
- SelectEIBInTree Method

The following sections describe each method.

AcquireEIB Method

Description

Displays the Acquire Signature dialog box to allow the user to acquire the current EIB.

C++ Syntax

```
BOOL CEWIX:: AcquireEIB();
```

Visual Basic Syntax

```
[form.]Control.AcquireEIB() As Boolean
```

Parameters

None.

Return Type

Boolean.

- TRUE if the acquire command is executed.
- FALSE if the acquire command is not executed.

C++ Example

```
BOOL result = pEWIX->AcquireEIB();
```

Visual Basic Example

```
EWIX1.AcquireEIB
```

Comment Method

Description

Displays the EIB Comment dialog box.

C++ Syntax

```
BOOL CEWIX:: Comment();
```

Visual Basic Syntax

```
[form.]Control.Comment() As Boolean
```

Parameters

None.

Return Type

Boolean.

- TRUE if the comment command is executed.
- FALSE if the comment command is not executed.

C++ Example

```
BOOL result = pEWIX->Comment();
```

Visual Basic Example

```
EWIX1.Comment
```

ConnectToServer Method

Description

Establishes a connection to the EIB Server.

C++ Syntax

```
BOOL CEWIX:: ConnectToServer();
```

Visual Basic Syntax

```
[form.]Control.ConnectToServer() As Boolean
```

Parameters

None.

Return Type

Boolean.

- TRUE if a connection is made.
- FALSE if a connection is not made or already exists.

C++ Example

```
BOOL result = pEWIX->ConnectToServer();
```

Visual Basic Example

```
EWIX1.ConnectToServer
```

Deviation Method

Description

Displays the EIB Deviation dialog box.

C++ Syntax

```
BOOL CEWIX:: Deviation();
```

Visual Basic Syntax

```
[form.]Control.Deviation() As Boolean
```

Parameters

None.

Return Type

Boolean.

- TRUE if the deviation command is executed.
- FALSE if the deviation command is not executed.

C++ Example

```
BOOL result = pEWIX->Deviation();
```

Visual Basic Example

```
EWIX1.Deviation
```

DisplayEWI Method

Description

Displays the currently selected EWI.

C++ Syntax

```
BOOL CEWIX:: DisplayEWI();
```

Visual Basic Syntax

```
[form.]Control.DisplayEWI() As Boolean
```

Parameters

None.

Return Type

Boolean.

- TRUE if the display EIB command is executed.
- FALSE if the display EIB command is not executed.

C++ Example

```
BOOL result = pEWIX->DisplayEWI();
```

Visual Basic Example

```
EWIX1.DisplayEWI
```

History Method

Description

Displays the History dialog box for the current EIB.

C++ Syntax

```
BOOL CEWIX:: History();
```

Visual Basic Syntax

```
[form.]Control.History() As Boolean
```

Parameters

None.

Return Type

Boolean.

- TRUE if the history command is executed.
- FALSE if the history command is not executed.

C++ Example

```
BOOL result = pEWIX->History();
```

Visual Basic Example

```
EWIX1.History
```

ReleaseEIB Method

Description

Displays the Release Signature dialog box to allow the user to release the current EIB.

C++ Syntax

```
BOOL CEWIX:: ReleaseEIB();
```

Visual Basic Syntax

```
[form.]Control.ReleaseEIB() As Boolean
```

Parameters

None.

Return Type

Boolean.

- TRUE if the release command is executed.
- FALSE if the release command is not executed.

C++ Example

```
BOOL result = pEWIX->ReleaseEIB();
```

Visual Basic Example

```
EWIX1.ReleaseEIB
```

SuperUserReleaseEIB Method

Description

The SuperUserReleaseEIB method is used to unlock an EIB locked by another user when there is no other way to unlock it. Normally, only the same user can unlock an EIB while running a session of the EWI ActiveX control.

If the EWI container program abnormally ends, then the EIB lock remains and it can only be unlocked with this method or the Emergency Release EIB command in the right-click menu. The SuperUserReleaseEIB method should only be used by supervisors.

The SuperUserReleaseEIB method is also known as the Release Override in the WorkInstruction Editor.

When you use the SuperUserReleaseEIB method, select the EIB in the EWI ActiveX tree control. Depending upon the EIB configuration for this feature, you may be prompted to enter a release signature. If you are not prompted, then the default EIB signature is used (the computer name or login name). It is strongly recommended that you specify a Release Override signature to be required when configuring your EIBs in the WorkInstruction Editor.

C++ Syntax

```
BOOL CEWIX:: SuperUserReleaseEIB();
```

Visual Basic Syntax

```
[form.]Control.SuperUserReleaseEIB() As Boolean
```

Parameters

None.

Return Type

Boolean.

- TRUE if the super user release command is executed.
- FALSE if the super user release command is not executed.

C++ Example

```
BOOL result = pEWIX->SuperUserReleaseEIB();
```

Visual Basic Example

```
EWIX1.SuperUserReleaseEIB
```

SkipStep Method

Description

Displays the Deviation dialog box to allow the user to skip the current step.

C++ Syntax

```
BOOL CEWIX:: SkipStep();
```

Visual Basic Syntax

```
[form.]Control.SkipStep() As Boolean
```

Parameters

None.

Return Type

Boolean.

- TRUE if the SkipStep command is executed.
- FALSE if the SkipStep command is not executed.

C++ Example

```
BOOL result = pEWIX->SkipStep();
```

Visual Basic Example

```
EWIX1.SkipStep
```

ReExecuteEWI Method

Description

Allows the user to re-execute the current EWI.

C++ Syntax

```
BOOL CEWIX:: ReExecuteEWI();
```

Visual Basic Syntax

```
[form.]Control.ReExecuteEWI() As Boolean
```

Parameters

None.

Return Type

Boolean.

- TRUE if the ReExecuteEWI command is executed.
- FALSE if the ReExecuteEWI command is not executed.

C++ Example

```
BOOL result = pEWIX->ReExecuteEWI();
```

Visual Basic Example

```
EWIX1.ReExecuteEWI
```

SelectFirstEIBInTree Method

Description

The SelectFirstEIBInTree method programmatically selects the first EIB of the first batch in the tree.

C++ Syntax

```
BOOL CEWIX:: SelectFirstEIBInTree();
```

Visual Basic Syntax

```
[form.]Control.SelectFirstEIBInTree() As Boolean
```

Parameters

None.

Return Type

Boolean.

- TRUE if the SelectFirstEIBInTree command is executed.
- FALSE if the SelectFirstEIBInTree command is not executed.

C++ Example

```
BOOL result = pEWIX->SelectFirstEIBInTree();
```

Visual Basic Example

```
EWIX1.SelectFirstEIBInTree
```

SelectEIBInTree Method

Description

The SelectEIBInTree method programmatically selects the indicated EIB within the indicated batch in the EIB tree. All five parameters must be included with the command. These parameters make up the unique batch name and unique EIB name.

C++ Syntax

```
BOOL CEWIX:: SelectEIBInTree(BSTR lpstrBatchID, BSTR lpstrBatchSerialNumber, BSTR
lpstrEIBName, BSTR lpstrEIBVersion, BSTR lpstrEIBEventID);
```

Visual Basic Syntax

```
[form.]Control.SelectEIBInTree(lpstrBatchID as String, lpstrBatchSerialNumber as String,
lpstrEIBName as String, lpstrEIBVersion as String, lpstrEIBEventID as String) As Boolean
```

Parameters

lpstrBatchID	The user defined batch ID number.
lpstrBatchSerialNumber	Batch serial number associated with this batch, generated internally by the server.
lpstrEIBName	The name of the EIB.
lpstrEIBVersion	The EIB version number from the DMS (Document Management System), Microsoft Visual Source Safe.
lpstrEIBEventID	The ID number of the event.

Return Type

Boolean.

- TRUE if the SelectEIBInTree command is executed.
- FALSE if the SelectEIBInTree command is not executed.

C++ Example

```
BOOL result = pEWIX->SelectEIBInTree(BatchID, BatchSerialNumber, EIBName, EIBVersion,
EventID);
```

Visual Basic Example

```
EWIX1.SelectEIBInTree(BatchID, BatchSerialNumber, EIBName, EIBVersion, EventID)
```

EWI ActiveX Control Events

The EWI control generates the following events:

- ConnectedToServer Event
- DisconnectedFromServer Event
- ServerChanged Event

ConnectedToServer Event

Description

Occurs when the control has connected to the EIB Server.

Event ID

2

C++ Syntax

```
void ConnectedToServer();
```

Visual Basic Syntax

```
Event ConnectedToSever()
```

DisconnectedFromServer Event

Description

Occurs when the control has disconnected from the EIB Server.

Event ID

3

C++ Syntax

```
void DisconnectedFromServer();
```

Visual Basic Syntax

```
Event DisconnectedFromServer()
```

ServerChanged Event

Description

Occurs when the control has switched EIB Servers.

Event ID

1

C++ Syntax

```
void ServerChanged();
```

Visual Basic Syntax

```
Event ServerChanged()
```

C++ Event Sink Map

The following is an example of an event sink map:

```
BEGIN_EVENTSINK_MAP(CTestDlg, CDialog)
    //{{AFX_EVENTSINK_MAP(CTestDlg)
     ON_EVENT(CTestDlg, IDC_EWIXCTRL1, 1 /* ServerChanged */, OnServerChangedEwixctrl1,
VTS_NONE)
     ON_EVENT(CTestDlg, IDC_EWIXCTRL1, 3 /* DisconnectedFromServer */,
OnDisconnectedFromServerEwixctrl1, VTS_NONE)
     ON_EVENT(CTestDlg, IDC_EWIXCTRL1, 2 /* ConnectedToServer */,
OnConnectedToServerEwixctrl1, VTS_NONE)
     //}}AFX_EVENTSINK_MAP
END_EVENTSINK_MAP()
```

Visual Basic Event Procedures

The following examples show procedures that Visual Basic creates for the EWI events that you include in your project.

```
' - Automation event
Private Sub EWIX1_ConnectedToServer()
MsgBox "Connected to the Server " + EWIX1.ServerComputer
End Sub

' - Automation event
Private Sub EWIX1_DisconnectedFromServer()
MsgBox "Disconnected from the server"
End Sub

' - Automation event
Private Sub EWIX1_ServerChanged()
MsgBox "server changed to " + EWIX1.ServerComputer
End Sub
```

Control Shortcut Keys

The following accelerator shortcut keys can be used for the EIB screen of the EWI ActiveX control.

Control Shortcut Keys	
Use this key combination...	**To...**
Ctrl + 1	Connect to the EIB Server.
Ctrl + 2	Perform an EIB deviation.
Ctrl + 3	Add an operator comment to an EIB.
Ctrl + 4	Display the history for the EIB.
Ctrl + 5	Acquire or release an EIB.
Ctrl + 6	Display the electronic work instructions for the selected phase.
Ctrl + 7	Skip the step.
Ctrl + 8	Re-execute the step.
Ctrl + 9	Display the EWI Help.
Ctrl + 0	Display the EWI Properties page. This accelerator is not represented as a button. You can also access the EWI Properties page at run time in the right-click menu.

Running a Control from a Web Page

Batch Execution supplies a sample HTML (HyperText Markup Language) page that shows the HTML code that you can use to run Batch Execution EWI ActiveX control on a web page. The following example is the HTML code in the EWI.htm file that is located in the Batch Execution Samples directory, typically C:\Program Files\Proficy\Proficy Batch Execution\samples\controls\HTML. Additional samples of HTML code also reside in this directory. You can also use the Index.htm or Menu.htm to view a menu of all the samples.

NOTE: *To obtain all of the graphics for the HTML samples, run the setup.bat file located in the HTML folder. Select Run from the Start menu, browse to the setup.bat file, and click OK to run.*

```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<base target="_top">
<title>Toothpaste Factory - Electronic WorkInstruction</title>
</head>

<body background="Images/BackgroundLogo.gif">

<p align="center"><font face="Impact" size="6" color="#FF0000">Electronic
WorkInstruction:</font></p>

<p align="center">
<object classid="clsid:A6B9D53F-41F8-11D4-9527-00A076A00776" width="1100" height="500">
<param name="_Version" value="131072">
<param name="_ExtentX" value="31009">
<param name="_ExtentY" value="17304">
<param name="_StockProps" value="0">
<param name="ServerComputer" value>
<param name="ProcessCellFilter" value="*">
<param name="UnitFilter" value="*">
<param name="PhaseFilter" value="*">
<param name="ConnectAtStartup" value="1">
<param name="ServerEditEnabled" value="1">
<param name="FilterEditEnabled" value="1">
<param name="EWINumberWidth" value="4.5">
<param name="EWITitleWidth" value="15.75">
<param name="EWIGroupWidth" value="2.5">
<param name="EWIStatusWidth" value="15">
<param name="EWISignatureWidth" value="20">
<param name="EWIValueWidth" value="20">
<param name="AutoAdvanceEWI" value="1">
<!-- set ServerEditEnabled to 0 if you do not want the user to edit the server settings -
->
<!-- set ComputerName if you want to set the server name to a remote EIB server(change
the "ComputerName" to the actual PC name) -->
<!-- <param name="ServerComputer" value="ComputerName"> -->
</object>
</p>

</body>
</html>
```

Configuring Internet Explorer (IE) to Run the Batch Execution ActiveX Controls

If you are running the EWI ActiveX controls from Internet Explorer, the browser must run in a separate process and security must be configured for the EWI ActiveX control.

▶To set up the browser to run the ActiveX control:

1. Start Internet Explorer. Do not connect to the web server.
2. On the Tools menu, click Internet Options.
3. Select the Advanced tab.

4. Select the Launch Browser Window in a New Process check box in the Browsing list, if it appears in the list.
5. Select the Security tab.
6. Select the Local Intranet icon.
7. Click the Sites button. The Local Intranet dialog box appears.
8. Click the Advanced button.
9. Enter the computer's path in the Add This Web Site to the Zone field. For example, the path might be similar to this: \\computername\share.

   NOTE: *Make sure that the path is a shared folder.*

10. Click OK.
11. Click OK again to close the Local Intranet dialog box.
12. Click the Custom Level button on the Internet Options dialog box. The Security Settings dialog box appears.
13. Confirm that the following options are defined as detailed in the following table.

Setting	**Option**
Download signed ActiveX controls	Prompt
Download unsigned ActiveX controls	Disable
Initialize and script ActiveX controls not marked as safe	Enable
Run ActiveX controls and plug-ins	Enable
Script ActiveX controls marked safe for scripting	Enable

14. Make sure the security level for the zone used is *not* set to High. If it is set to High, reset it. The ActiveX controls will not load if it is set to High.
15. Click OK.
16. Click OK from the Internet Options dialog box.

Configuring the Tab and Enter key for the EWI ActiveX Control

The IsSelectable property enables an ActiveX control in iFIX to respond to the Tab and Enter keys. This property is available to every object you draw in the WorkSpace; the default value of the IsSelectable property is False. To set this property, right-click on the EWI control in design mode, and then select the Property Window option to display the iFIX property window for the EWI control. Toggle the value of IsSelectable from False to True.

This can also be done with a VBA script when the picture loads at run time:

```
Private Sub CFixPicture_Initialize()
EWIX1.IsSelectable = True
End Sub
```

This only applies to tabbing on the first level screen of the control. Pop-up dialog boxes like the EWI dialog do not require these steps.

Adjusting Column Widths of the Grid in the EWI ActiveX Control

You can change the column widths in design mode in the property list provided by Visual Basic and the Proficy iFIX WorkSpace. Users can change column widths in the EWI Display Component while in run mode. However, once you close the application and then restart it, the changes are lost unless you write a script in VBA to save the column width values when the picture/form closes and set them when the picture/form loads.

The default properties for the column widths are as follows:

```
EWIX1.EWIGroupWidth = 2.5

EWIX1.EWINumberWidth = 4.5

EWIX1.EWISignatureWidth = 20

EWIX1.EWIStatusWidth = 15

EWIX1.EWITitleWidth = 15.75

EWIX1.EWIValueWidth = 20

EWIX1.EWIPerformedByWidth = 20

EWIX1.EWIVerifyByWidth = 20
```

You cannot graphically set the column width fields while in configure/design mode within iFIX or Visual Basic. However, you can graphically set these fields in design mode from the ActiveX Control Test Container and the Batch Execution Workspace.

Using the WorkInstruction Run-time Control

The main screens that you use from the EWI ActiveX control are the:

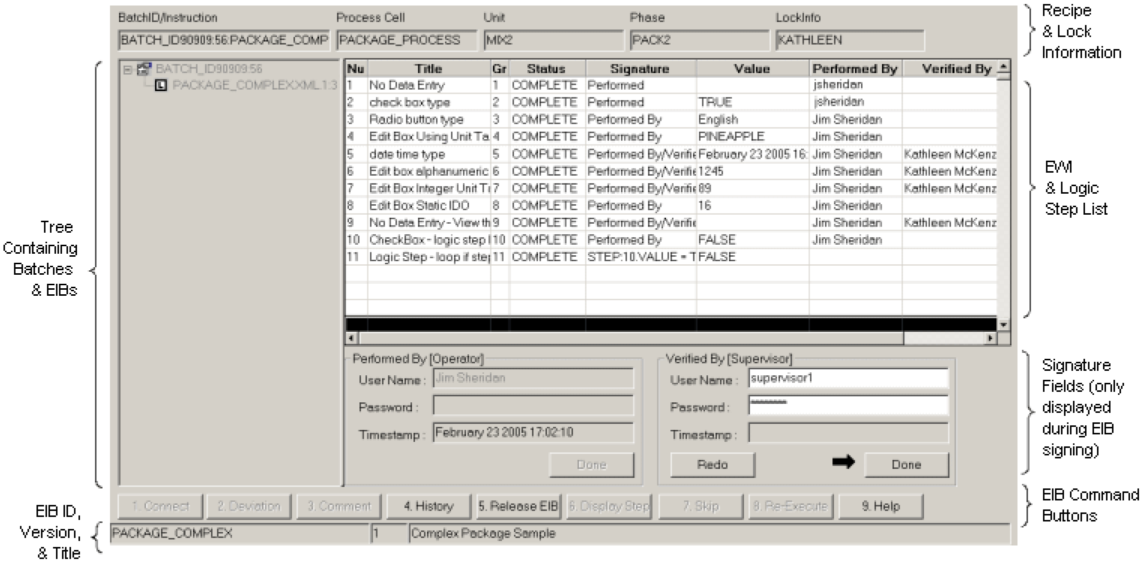
- Main EIB Display screen
- EWI Step Display screen
- Logic Step Display screen

The sections that follow describe these screens and functions that you can perform from the main EIB Display screen, such as:

- Acquiring an EIB
- Displaying EWI or Logic Steps
- Adding Comments
- Viewing the History of EIB actions
- Releasing an EIB
- Emergency Release
- Skipping or Re-Executing Steps

EIB Display Screen

An example of the EIB Display screen is shown in the following figure.



BatchID/Instruction	Process Cell	Unit	Phase	LockInfo
BATCH_ID90909:56:PACKAGE_COMP	PACKAGE_PROCESS	MIX2	PACK2	KATHLEEN

Nu	Title	Gr	Status	Signature	Value	Performed By	Verified By
1	No Data Entry	1	COMPLETE	Performed		jsheridan	
2	check box type	2	COMPLETE	Performed	TRUE	jsheridan	
3	Radio button type	3	COMPLETE	Performed By	English	Jim Sheridan	
4	Edit Box Using Unit Ta	4	COMPLETE	Performed By	PINEAPPLE	Jim Sheridan	
5	date time type	5	COMPLETE	Performed By/Verifie	February 23 2005 16:	Jim Sheridan	Kathleen McKenz
6	Edit box alphanumeric	6	COMPLETE	Performed By/Verifie	1245	Jim Sheridan	Kathleen McKenz
7	Edit Box Integer Unit Ta	7	COMPLETE	Performed By/Verifie	89	Jim Sheridan	Kathleen McKenz
8	Edit Box Static IDO	8	COMPLETE	Performed By	16	Jim Sheridan	
9	No Data Entry - View th	9	COMPLETE	Performed By/Verifie		Jim Sheridan	Kathleen McKenz
10	CheckBox - logic step I	10	COMPLETE	Performed By	FALSE	Jim Sheridan	
11	Logic Step - loop if ste	11	COMPLETE	STEP:10.VALUE = T	FALSE		

Performed By [Operator]
User Name : Jim Sheridan
Password :
Timestamp : February 23 2005 17:02:10

Verified By [Supervisor]
User Name : supervisor1
Password : ********
Timestamp :

PACKAGE_COMPLEX | 1 | Complex Package Sample

EIB Display Screen

General EIB Fields

Each EIB contains the following information:

Recipe/Instruction – the name of the selected batch.

The following fields are empty if an EIB is not selected:

Process Cell – the process cell that belongs to the unit that is running the phase for the currently selected batch.

Unit – the unit that is running the phase for the currently selected batch.

Phase – the phase that is running for the currently selected batch.

LockInfo – the computer name where this EIB is locked. This field is empty if the EIB is unlocked.

Columns Displayed

The EWI ActiveX control displays the following columns in the grid display:

Number – the number of the EWI or Logic Step. This number is an internal number generated by WorkInstruction.

Title – the title of the EWI or Logic Step.

Group – the group number to which the EWI or Logic Step belongs.

Status – the status of the EWI or Logic Step. The values that you can view in the status field:

COMPLETE – the EWI or Logic Step is complete.

DATA COMMIT – the EWI or Logic Step value is entered and the Perform By signature has been completed, but the Verified By signature has not been completed yet.

READY – the EWI or Logic Step can be performed.

IDLE – the EWI or Logic Step cannot be performed until all of the EWIs in previous groups are completed.

ACTIVE – the step is currently being executed, but is not signed off.

SKIPPED – the step has been skipped via the SKIP function.

NOT RUN – the step has not been run. Indicates that the step has been "jumped over" via a logic step and has not been executed.

Signature – authorization level for the signature. The values that can appear in this field are Performed, Performed By, and Performed By/Verified By. If a step is a logic step, the expression displays instead.

Value – the value of the data type for the EWI or Logic Step.

Performed By – the full name of the user who signed the Performed or Performed By signature. If the user was not required to enter a user name and password, the Windows computer name or login name is recorded here.

Verified By – the full name of the user who signed the Verified By signature.

Batch and EIB Tree

The tree contains two levels. The first level contains the batches, and the second level contains the EIBs of the batch.

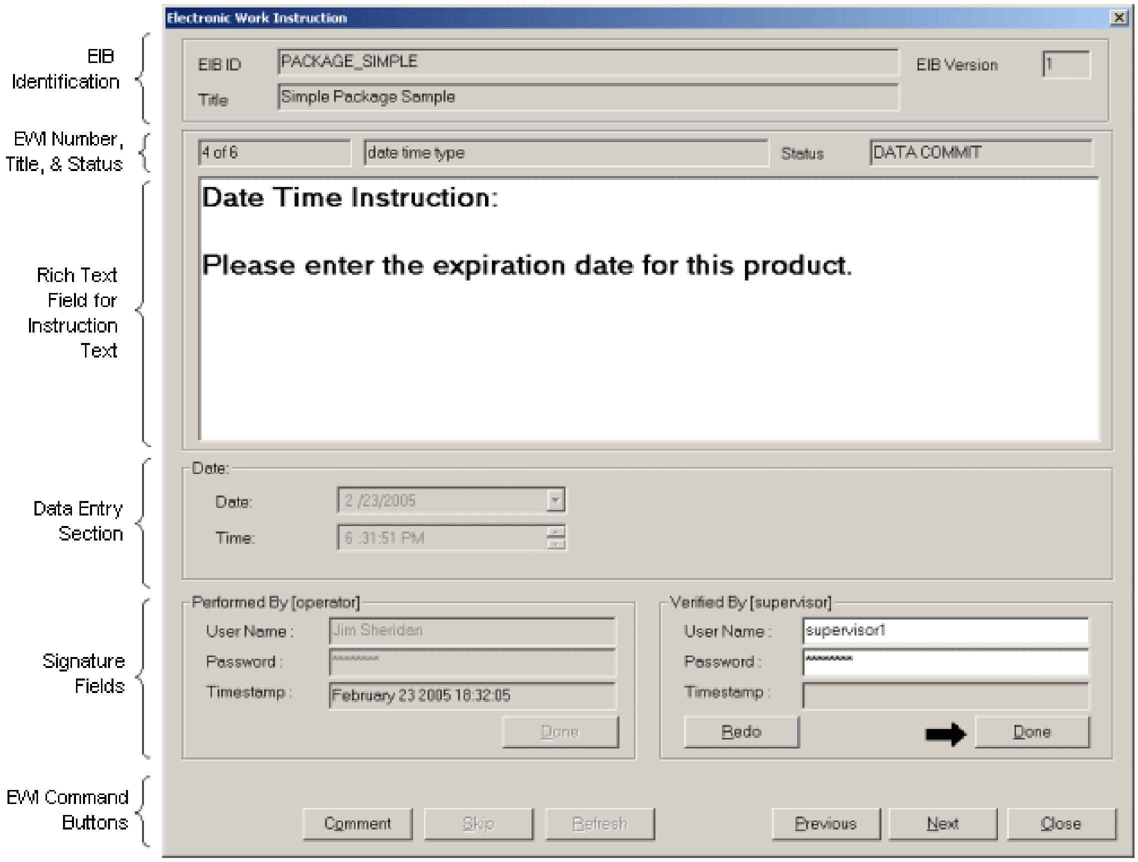
EIB Command Buttons and Shortcut Keys

The following command buttons and shortcut keys can be used from the EIB Display screen of the EWI ActiveX control.

Control Shortcut Keys		
Button Name	**Shortcut**	**Description**
Connect	Ctrl + 1	Connect to the EIB Server.
Deviation	Ctrl + 2	Perform an EIB deviation.
Comment	Ctrl + 3	Add an operator comment to an EIB.
History	Ctrl + 4	Display the history for the EIB.
Acquire or Release EIB	Ctrl + 5	Acquire or release an EIB.
Display Step	Ctrl + 6	Display the electronic work instructions or logic step for the selected phase.
Skip	Ctrl + 7	Skip the selected EIB.
ReExecute	Ctrl + 8	Re-execute the selected EIB.
Help	Ctrl + 9	Display the EWI Help.
–	Ctrl + 0	Display the EWI Properties page. This accelerator is not represented as a button. You can also access the EWI Properties page at run time in the right-click menu.

EWI Step Display Screen

An example of an individual Step Display screen for EWIs is shown in the following figure.



EIB ID: PACKAGE_SIMPLE

EIB Version: 1

Title: Simple Package Sample

4 of 6 | date time type | Status: DATA COMMIT

Date Time Instruction:

Please enter the expiration date for this product.

Date:

Date: 2 /23/2005

Time: 6 :31:51 PM

Performed By [operator]

User Name : Jim Sheridan

Password : ********

Timestamp : February 23 2005 18:32:05

Verified By [supervisor]

User Name : supervisor1

Password : ********

Timestamp :

Sample EWI Step Display Screen

Step Groups

Operators perform EWI or Logic Steps in the order defined in the WorkInstruction Editor. The EWI or Logic Step with the lowest group number must be performed before an EWI or Logic Step with a higher group number can be performed. More than one EWI or Logic Step can have the same group number, allowing EWI or Logic Steps with the same group number to be performed in any order.

The EWI or Logic Step is enabled for data entry if you lock the EIB and if the EWI is in the lowest group that has not been completed yet.

EWI Command Buttons

The following list describes the EWI command buttons shown at the bottom of the EWI Step Display screen:

Run Task – enabled if the EIB is locked by you, the EWI is enabled for data entry, and there is a run task command configured in the WorkInstruction Editor. Clicking this button executes the command.

Comment – enabled if the EIB is locked by you and the EWI or Logic Step is enabled for data entry. Allows you to enter one or more comments for the selected EWI. These comments are visible in the history log and are part of the EIB data stored in the database.

Skip – enabled if the EIB is locked by you and the EWI or Logic Step is enabled for data entry. Allows you skip a step by entering a deviation. Deviations are visible in the history log and are part of the EIB data stored in the database.

Previous – displays the previous EWI or Logic Step.

Next – displays the next EWI or Logic Step.

Close – closes the EWI or Logic Step display.

Refer to the EWI Step Display Screen section for an illustration of this screen.

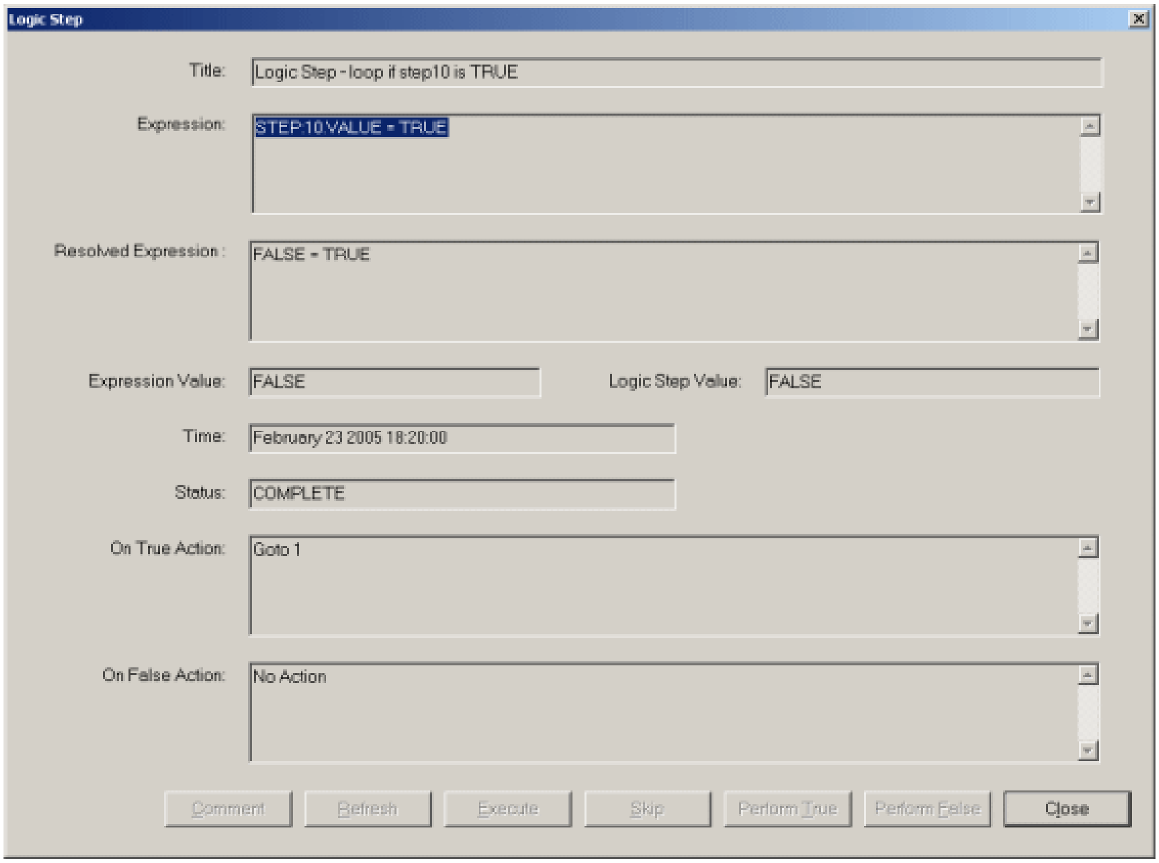
Shortcut Keys from the Step Display Screen

The following accelerator shortcut keys can be used from the step display screen.

Control Shortcut Keys	
Use this key combination...	**To...**
Alt + C	Close the Step Display screen.
Alt + O	Add a comment.
Alt + D	Commit the signature.
Alt + R	Redo the EWI data entry and signature.
Alt + N	Move to the next EWI in the sequence.
Alt + P	Move to the previous EWI in the sequence.

Logic Step Display Screen

An example of the Logic Step Display screen is shown in the following figure.



Title: Logic Step - loop if step10 is TRUE

Expression: STEP:10.VALUE = TRUE

Resolved Expression : FALSE = TRUE

Expression Value: FALSE

Logic Step Value: FALSE

Time: February 23 2005 18:20:00

Status: COMPLETE

On True Action: Goto 1

On False Action: No Action

Logic Step Display Screen

General Logic Step Fields

Each Logic Step contains the following information:

Title – the name of the Logic Step.

Expression – the original expression.

Resolved Expression – the resolved expression. Any variables that cannot be resolved are displayed as <*unresolved values...*>.

Expression Value – the value of the expression.

Logic Step Value – the value of the Logic Step. Normally, this value matches the Expression Value. The operator can also choose to perform the opposite action. If the expression value is unresolved, the operator can choose to perform the True, False, or Skip action.

Time – the time that the expression was resolved.

Status – the Logic Step status.

On True Action – the action to perform on True, such as: Goto, MessageBox, Comment, Deviation, End EIB, or No Action.

On False Action – the action to perform on False, such as: Goto, MessageBox, Comment, Deviation, End EIB, or No Action.

Logic Step Command Buttons

The following list describes the Logic Step command buttons shown at the bottom of the EWI Step Display screen:

Comment – enters a comment. This button is only enabled when the Logic Step is active.

Refresh – refreshes the Logic Step value. This button is only enabled when the Logic Step is active.

Execute – executes the Logic Step based on the current expression value. This button is only enabled when the Logic Step is active, and if the expression value is either True or False.

Skip – skips the Logic Step. This button is only enabled when the Logic Step is active.

Perform True – executes the Logic Step and performs the True action. If the expression is not True, a Deviation dialog box displays. The operator must enter the required fields into the Deviation dialog box before proceeding. The Perform True button is only enabled when the Logic Step is active.

Perform False – executes the Logic Step and performs the False action. If the expression is not False, a Deviation dialog box displays. The operator must enter the required fields into the Deviation dialog box before proceeding. The Perform False button is only enabled when the Logic Step is active.

Close – closes the Logic Step Display screen. This button is only enabled when the Logic Step is active.

Refer to the EWI Step Display Screen section for an illustration of this screen.

Looping and Operator Prompts

If a transition contains a recipe parameter and this parameter requires an operator prompt, the prompt is only issued once. So, if you put this phase in a loop, the next time through, the transition retains the old value and does not re-prompt.

If you want to re-prompt the user each time the phase goes through the loop, add a condition to the transition expression to check when the phase state = COMPLETE. For example, an expression you might use could be:

```
PHASE1:1.STATE = COMPLETE AND PHASE1:1.PARAMETER1 > 2
```

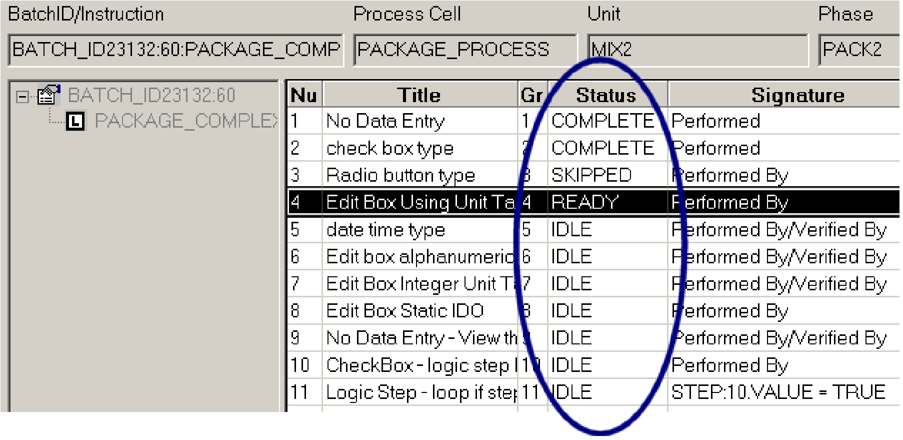
In this example, the next time through the loop the user is re-prompted. If you do not use PHASE1:1.STATE = COMPLETE in the transition expression, after the operator prompt has been issued the first time and the operator enters a valid value, the transition expression always evaluates to True. The next time through the loop, the transition will issue a Stop command as soon as it evaluates its expression and then continue in the loop without ever re-prompting the user for a value.

EWI and Logic Step States

The following table describes all of the possible states for EWI or Logic steps.

State	Description
Ready	The step is ready for execution.
Idle	The step cannot be executed yet.
Active	The step is currently being executed, but is not signed off.
Data Commit	The first signature has been captured. The step is awaiting the second signature.
Complete	The step is completed and signed off.
Skipped	The step has been skipped via the SKIP function.
Not Run	The step has not been run. Indicates that the step has been "jumped over" via a logic step and has not been executed.

The operator can view the state from the Status field of the individual EWI or Logic Step. You can also view these states in the Status column of the EWI display control, as shown in the following figure.



BatchID/Instruction	Process Cell	Unit	Phase
BATCH_ID23132:60:PACKAGE_COMP	PACKAGE_PROCESS	MIX2	PACK2

Nu	Title	Gr	Status	Signature
1	No Data Entry	1	COMPLETE	Performed
2	check box type	2	COMPLETE	Performed
3	Radio button type	3	SKIPPED	Performed By
4	Edit Box Using Unit Ta	4	READY	Performed By
5	date time type	5	IDLE	Performed By/Verified By
6	Edit box alphanumeric	6	IDLE	Performed By/Verified By
7	Edit Box Integer Unit T	7	IDLE	Performed By/Verified By
8	Edit Box Static IDO	8	IDLE	Performed By
9	No Data Entry - View th	9	IDLE	Performed By/Verified By
10	CheckBox - logic step I	10	IDLE	Performed By
11	Logic Step - loop if step	11	IDLE	STEP:10.VALUE = TRUE

Status Column in the EWI Display Control

Where to Start in the EIB Display Screen

To begin working with EIBs, EWIs, and Logic Steps, you must first connect to the EIB Server. To do this, click the Connect button. Assuming your connection to the server is successful, you can now begin working on EIBs.

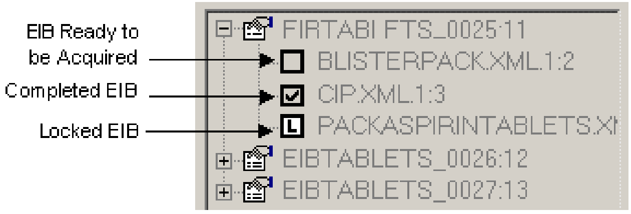
To work on an EIB, you first select it from the tree and then you click the Acquire EIB button. After you click Acquire EIB, the EIB becomes locked. When an EIB is locked, an L displays next to the EIB you are currently working on in the tree on the left side of your screen.

Before you display a step, look at the group numbers displayed in the Group column of the grid. You must follow the sequence of the group numbers when you select an EWI or Logic Step to complete. If you go out of order, the data entry fields are unavailable. If there are two or more group numbers that are the same, you can perform those EIBs with the same group in any order.

Select a step with the Ready state, and click the Display Step button to review the EWI Step or execute the Logic Step.

To complete an EIB, enter the Performed By or Verified By signatures, if required, and then click Release EIB. The EIB signature section is not displayed until the EIB is ready for signing (that is, all EWI and Logic Steps are completed). When an EIB is completed and released, it is then unlocked; a check box displays next to that EIB. The EIB is automatically removed from the list when the EIB Server completes processing it.

The following figure shows an example of how a locked and a completed EIB displays in the EIB tree.

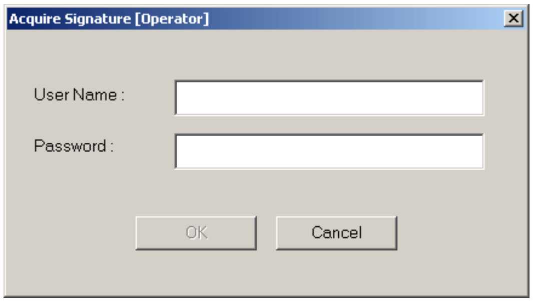


Example of a Locked EWI

Acquiring an EIB

Click on the Acquire EIB button to lock the selected EIB. You can configure the Acquire EIB command so that the user must enter an electronic signature (a user name and password) when acquiring the EIB. Otherwise, the Windows computer name or login name is recorded as the electronic signature. The EIB Performed by Group name is displayed in the title bar of the Acquire Signature dialog as shown in the following figure.

NOTE: *Once you lock an EIB you cannot select any other EIBs until you unlock the current EIB.*



Acquire Signature Dialog Box

Displaying EWI or Logic Steps

When an EIB is locked (acquired), you can access the EWI or Logic Steps for the selected EIB by clicking the Display Step button. When you are finished entering the data and signatures on the step display screen, by default, you will automatically advance to the next EWI or Logic Step in the sequence. If the Auto Advance EWI property is set to False, you need to click Next to proceed to the next EWI or Logic Step.

There are several types of data entry that can appear in the Step Display screen for an EWI:

- Check boxes
- Radio buttons
- Date/time (automatic or manual)
- Edit box
- No data entry

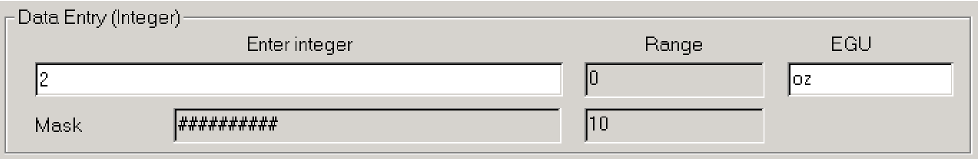
Only one type of data entry displays, per EWI. The following sections display examples of each of the data entry types.

Example: Edit Box Data Entry

For the edit box in the Data Entry group box, there are three data types that you could be required to enter:

- Alphanumeric
- Real
- Integer

The following figure shows an example of a Data Entry group box requesting an integer value.



Enter integer: 2
Range: 0
EGU: oz
Mask: ##########
Range: 10

Example of Edit Box, Data Entry

Refer to the figure illustrated in the EWI Step Display Screen section if you want to see where the Data Entry group box appears in the Step Display screen.

The Mask field displays the type of value you can enter in the edit box. This field is read-only. Each character in the Mask field maps to a placeholder for a digit that you can enter. For example, if the data type is real and the Mask is defined as ####.###, the largest value that you can enter can only be 7 digits long, with three places after the decimal point. Each # symbol represents a digit the user can enter. The Mask field only applies to integer and real values.

An edit box can optionally be configured to display a range. If a range appears, the value you enter in the edit box must be between the high and low values. You cannot change the high and low range from the Step Display screen. If there is a discrepancy in the range, a Deviation dialog box appears. Refer to the Data Deviations section for details. Ranges only apply to integer and real values.

In addition, an edit box can optionally be configured to include an EGU field. If an EGU field appears, as shown the Data Entry edit box example in the previous figure, you must enter the engineering units for the data you entered in the edit box.

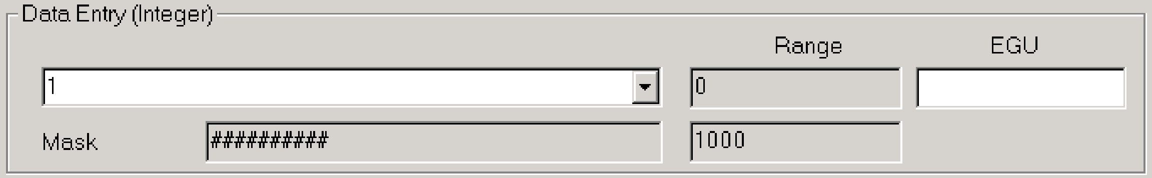
Edit boxes can optionally be configured boxes to display suggested values. These suggested values are called IDOs (Intelligent Display Objects). There are three types of IDOs that can appear for suggest values:

Static (user defined) – displays a static list of values in a drop-down list.

Dynamic (SQL Statement) – displays a dynamic list of values in a drop-down list. Click the Refresh button to update this list with the latest values from the database.

Unit Tag Class – displays the unit tag value. Click the Refresh button to get the latest value from the system.

These IDOs can display integer, real, or alphanumeric string values. Select a suggested value or enter another value in this data entry field. Refer to the following figure for an example of an edit box with static IDO. Note the drop-down list for the edit box.



Data Entry (Integer)

1

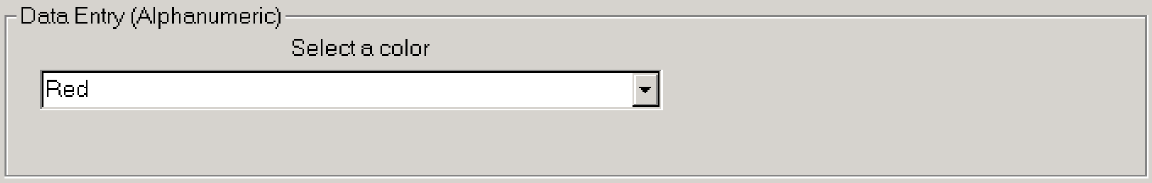
Range
0
1000

EGU

Mask ##########

Example of Edit Box with Static IDO

Refer to the next figure for an example of an edit box with dynamic IDO values. This data entry field is similar to the static IDO data entry field, except that it contains a Refresh button at the bottom of the dialog box that you can use to update the list with the latest values from the database.



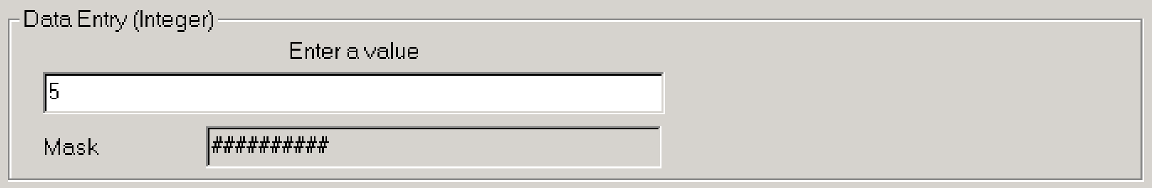
Data Entry (Alphanumeric)

Select a color

Red

Example of Edit Box with Dynamic IDO

Refer to the next figure for an example of an edit box with unit tag IDO. You can use the sample value for the unit tag or enter one manually. The Refresh button at the bottom of the dialog box allows the user to update the display with the latest unit tag value.



Data Entry (Integer)

Enter a value

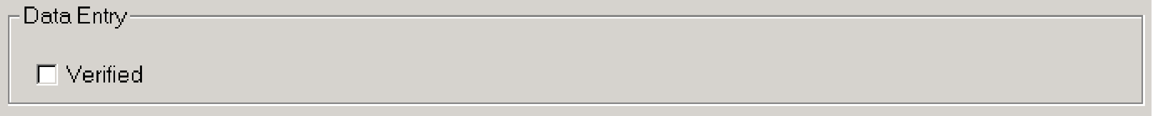
5

Mask ##########

Example of Edit Box with a Unit Tag IDO

Example: Check Box Data Entry

If a check box appears in the Data Entry group box, select the check box to select the option, or leave it unchecked. At most, only one check mark will be displayed. The following figure shows an example of a Data Entry group box with a check box.



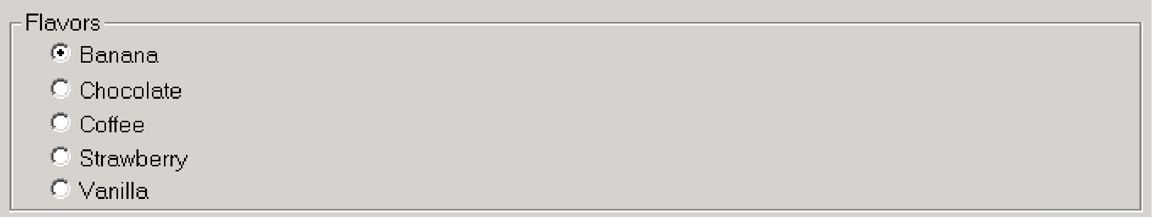
Data Entry

□ Verified

Example of Check Box Data Entry

Example: Radio Button Data Entry

If radio buttons appear in the Data Entry group box, select a radio button to select the option. The following figure shows an example of a Data Entry group box with radio buttons.



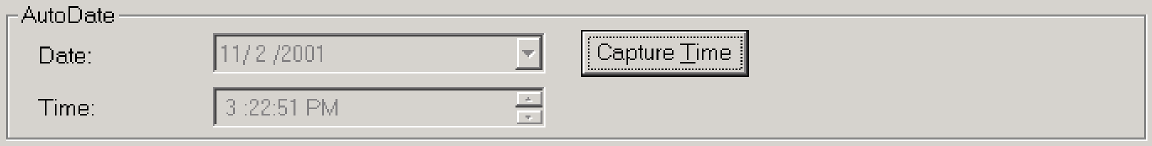
Flavors

● Banana
○ Chocolate
○ Coffee
○ Strawberry
○ Vanilla

Example of Radio Button Data Entry

Example: Automatic Date/Time Data Entry

If an automatic date/time selection appears in the Data Entry group box, the Date and Time fields are read-only. You must click Capture Time when you are finished following the instructions displayed in the EWI in order to proceed to the signature field. The following figure shows an example of a Data Entry group box with an automatic date/time entry.



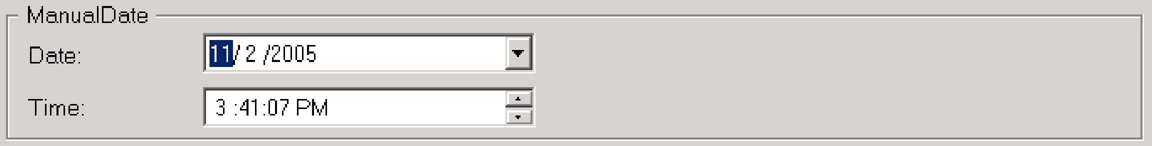
Date: 11/ 2 /2001

Time: 3 :22:51 PM

Example of Automatic Date/Time Data Entry

Example: Manual Date/Time Data Entry

If a manual date/time selection appears in the Data Entry group box, the Date and Time fields are available for editing. Either enter a date and time or use the drop-down list and spin box to capture values. The following figure shows an example of a Data Entry group box with manual date/time entry.



Date: 11/ 2 /2005

Time: 3 :41:07 PM

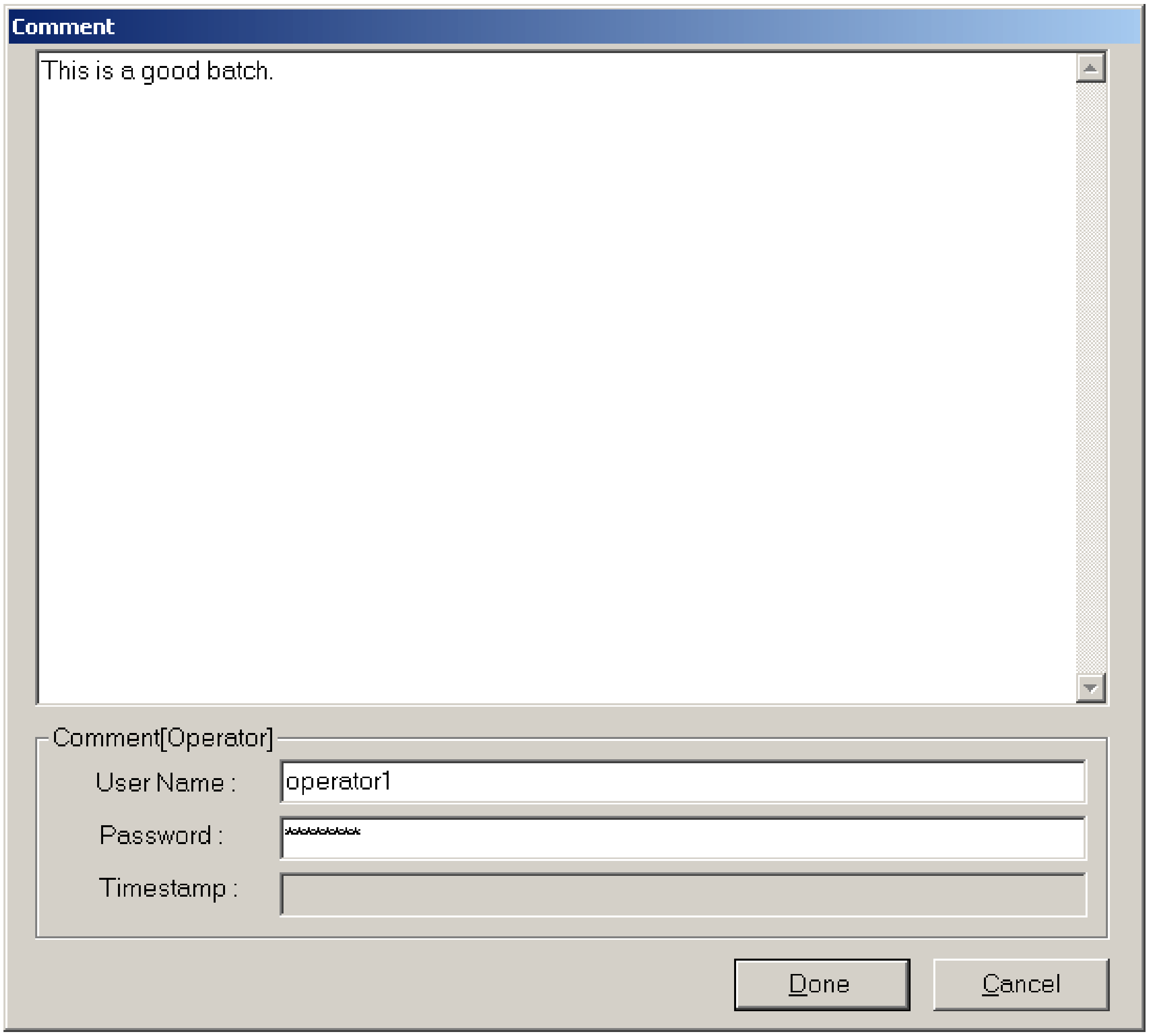
Example of Manual Date/Time Data Entry

Adding Comments

When the EIB is locked, you can add comments to the EIB or EWI. Each comment may optionally require a signature. To sign a comment, the user must enter an electronic signature (a user name and password). Otherwise, the Windows computer name or login name is recorded as the electronic signature.

NOTE: *The acceptable group, if required, is displayed in brackets in the Comment group box.*

The following figure shows an example of the Comments dialog box.



Comment

This is a good batch.

Comment[Operator]

User Name : operator1

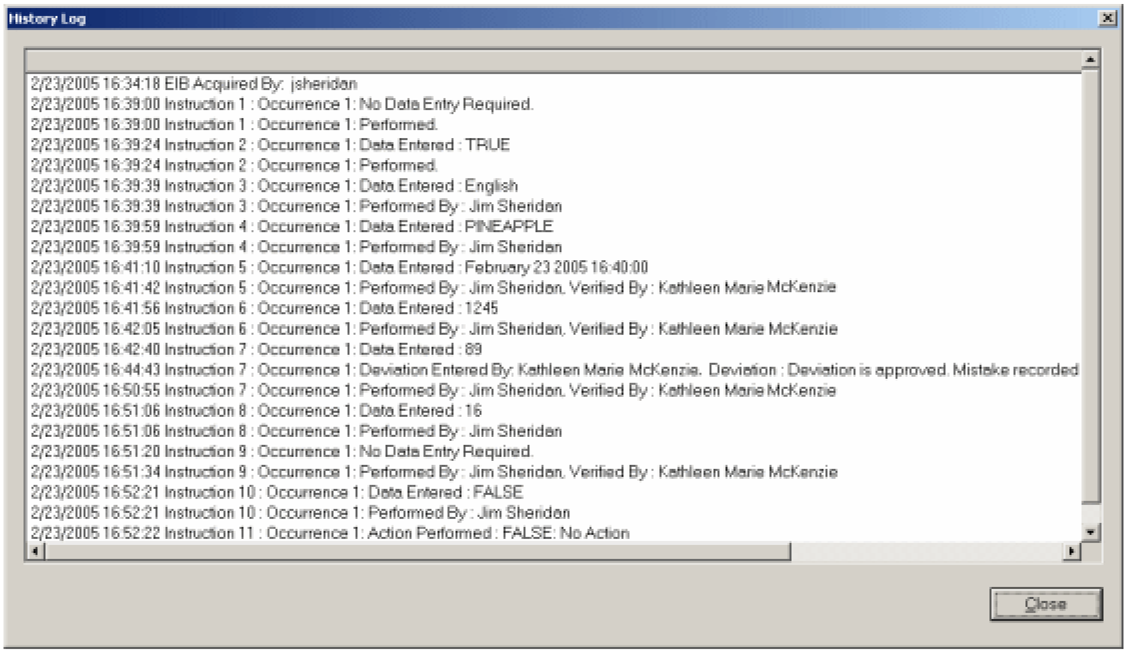
Password : ********

Timestamp :

Comments Dialog Box

Viewing the History

From the EIB Display screen you can view a history log of all the actions that an operator or supervisor performed. To view the history, click the History button from the EIB Display screen. The information in this window is provided for viewing purposes only. The following figure shows an example of a history log.



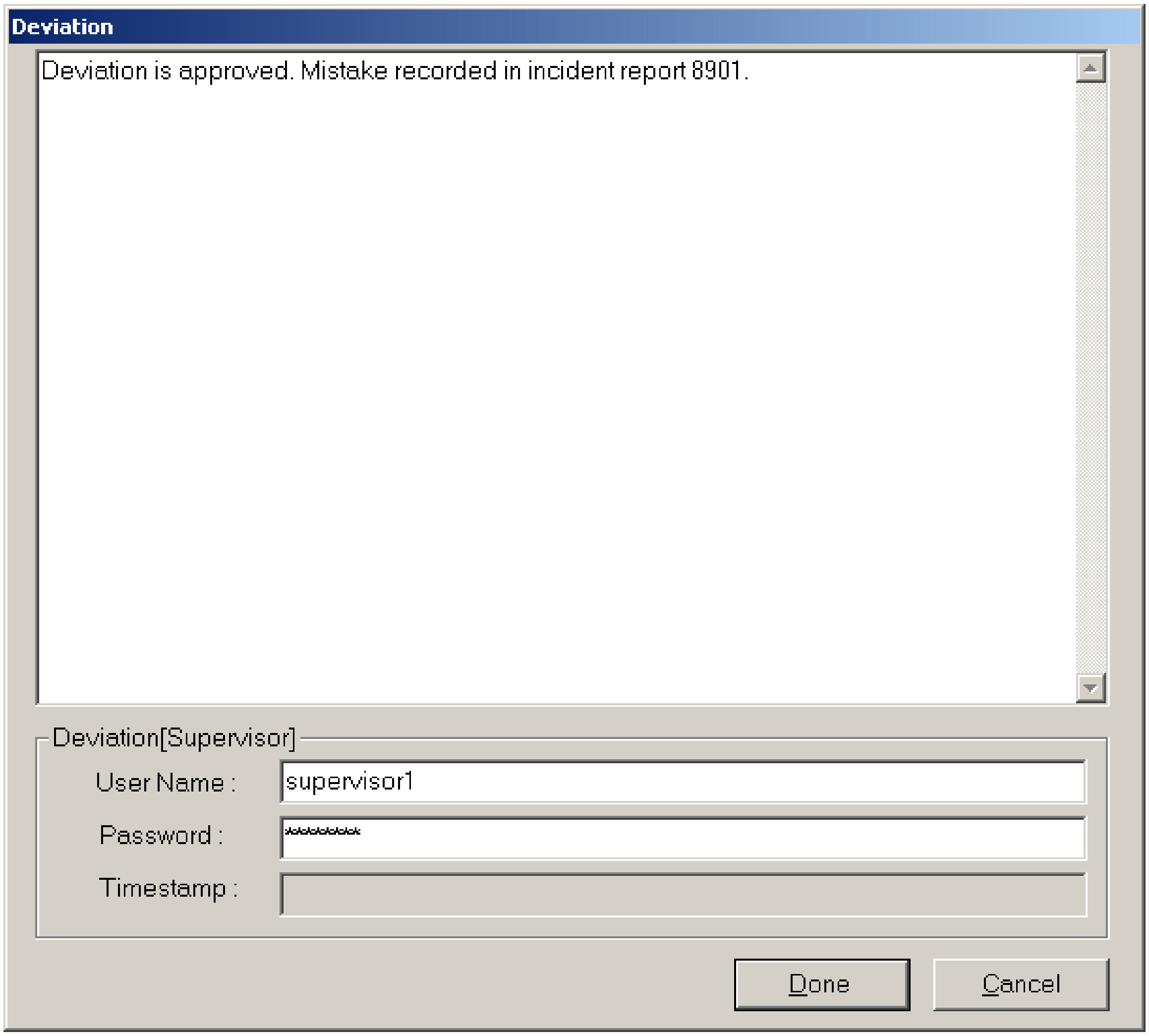
```
2/23/2005 16:34:18 EIB Acquired By:  jsheridan
2/23/2005 16:39:00 Instruction 1 : Occurrence 1: No Data Entry Required.
2/23/2005 16:39:00 Instruction 1 : Occurrence 1: Performed.
2/23/2005 16:39:24 Instruction 2 : Occurrence 1: Data Entered : TRUE
2/23/2005 16:39:24 Instruction 2 : Occurrence 1: Performed.
2/23/2005 16:39:39 Instruction 3 : Occurrence 1: Data Entered : English
2/23/2005 16:39:39 Instruction 3 : Occurrence 1: Performed By : Jim Sheridan
2/23/2005 16:39:59 Instruction 4 : Occurrence 1: Data Entered : PINEAPPLE
2/23/2005 16:39:59 Instruction 4 : Occurrence 1: Performed By : Jim Sheridan
2/23/2005 16:41:10 Instruction 5 : Occurrence 1: Data Entered : February 23 2005 16:40:00
2/23/2005 16:41:42 Instruction 5 : Occurrence 1: Performed By : Jim Sheridan. Verified By : Kathleen Marie McKenzie
2/23/2005 16:41:56 Instruction 6 : Occurrence 1: Data Entered : 1245
2/23/2005 16:42:05 Instruction 6 : Occurrence 1: Performed By : Jim Sheridan, Verified By : Kathleen Marie McKenzie
2/23/2005 16:42:40 Instruction 7 : Occurrence 1: Data Entered : 89
2/23/2005 16:44:43 Instruction 7 : Occurrence 1: Deviation Entered By: Kathleen Marie McKenzie.  Deviation : Deviation is approved. Mistake recorded
2/23/2005 16:50:55 Instruction 7 : Occurrence 1: Performed By : Jim Sheridan. Verified By : Kathleen Marie McKenzie
2/23/2005 16:51:06 Instruction 8 : Occurrence 1: Data Entered : 16
2/23/2005 16:51:06 Instruction 8 : Occurrence 1: Performed By : Jim Sheridan
2/23/2005 16:51:20 Instruction 9 : Occurrence 1: No Data Entry Required.
2/23/2005 16:51:34 Instruction 9 : Occurrence 1: Performed By : Jim Sheridan, Verified By : Kathleen Marie McKenzie
2/23/2005 16:52:21 Instruction 10 : Occurrence 1: Data Entered : FALSE
2/23/2005 16:52:21 Instruction 10 : Occurrence 1: Performed By : Jim Sheridan
2/23/2005 16:52:22 Instruction 11 : Occurrence 1: Action Performed : FALSE: No Action
```

History Window

Data Deviations

Deviations can occur in an EIB or EWI. Each deviation records an electronic signature. The user name and password that the user enters is checked against the Windows security group that was defined for the authority check. Otherwise, the Windows computer name or login name is recorded as the electronic signature.

When a deviation occurs in the Step Display screen, a window automatically appears. When a deviation occurs, either forced or automatic, the dialog box shown in appears.



Deviation

Deviation is approved. Mistake recorded in incident report 8901.

Deviation[Supervisor]

User Name : supervisor1

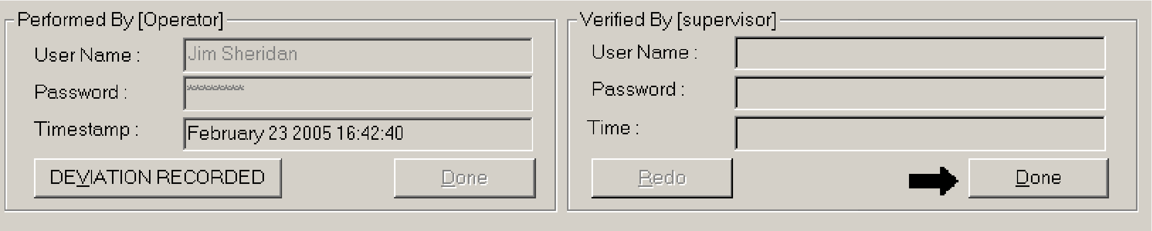
Password : ********

Timestamp :

Deviation Dialog Box

Enter the user name and password of the user to accept the deviation, if required. You will know which signature group is required by looking at the group listed within the brackets in the text of title bar in the Release Signature dialog box. The Verified By Group is displayed in the title bar. Click Done to finish the signature for the deviation.

Once a deviation has been entered for an EWI, a Deviation Recorded button displays underneath the signature fields. Click Deviation Recorded to review the deviation information at any time. Refer to the following figure for an example of the Deviation Recorded button.



Performed By [Operator]

User Name : Jim Sheridan

Password : ********

Timestamp : February 23 2005 16:42:40

Verified By [supervisor]

User Name :

Password :

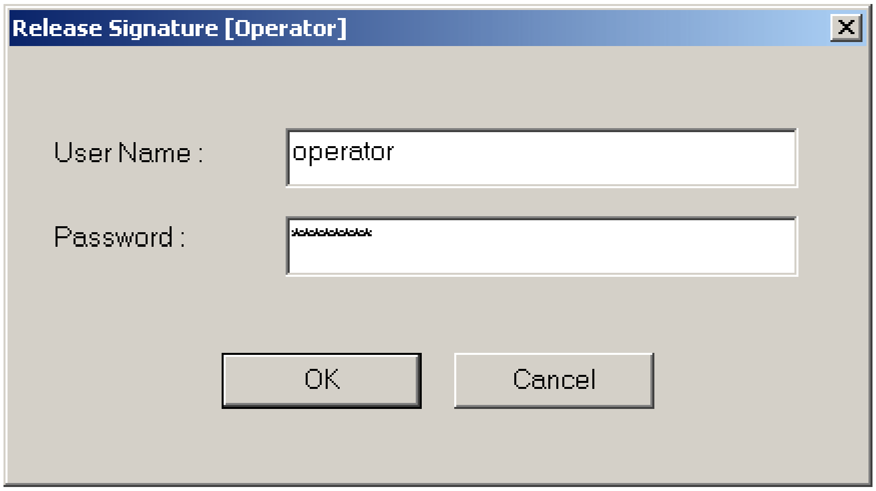
Time :

Signature with Deviation Record

Releasing an EIB

When you are finished working with an EIB, you must release the EIB. You can configure the Release EIB command so that the user must enter an electronic signature (a user name and password) when releasing an EIB. Otherwise, the Windows computer name or login name is recorded as the electronic signature.

To release the EIB, click Release EIB. The Release Signature dialog box appears, as shown in the following figure, if an electronic signature (user name and password) is required from the user.



User Name : operator

Password : ********

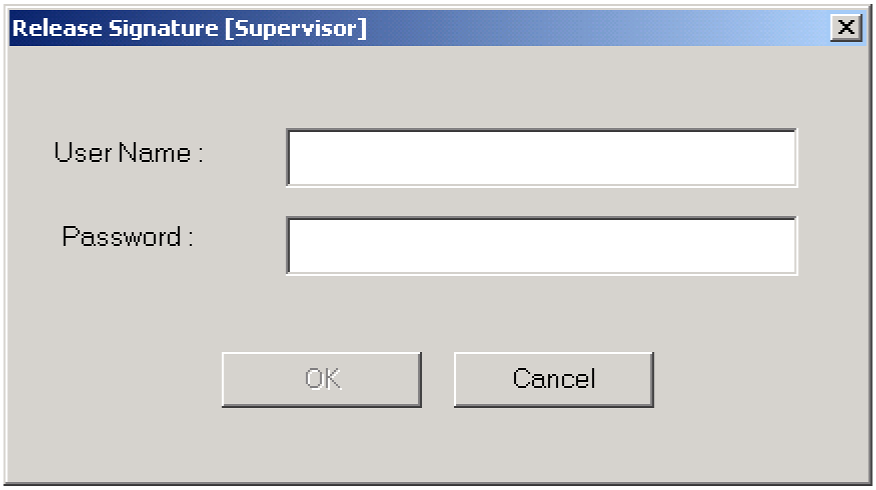
Release Signature Dialog Box

Enter the user name and password of the user. You can determine the signature group required by looking at the group listed within the brackets in the title bar of the Release Signature dialog box. For example, in the previous figure the group name is *operator*. Click OK to finish the signature.

Emergency Release

The EWI ActiveX control provides a right-click menu option, Emergency Release EIB, to allow the supervisor to unlock EIBs that were locked on other stations that are no longer accessible. The Emergency Release should only be performed when it is not possible to unlock the EIB by other means.

NOTE: *The Emergency Release EIB right-click option can only unlock an EIB that is locked on another EWI ActiveX control display.*



Emergency Release Signature Dialog Box

Enter the user name and password of the user. You can determine the signature group required by looking at the group listed within the brackets in the title bar of the Release Signature dialog box. For example, in the previous figure the group name is *Supervisor*. Click OK to finish the signature.

For information about configuring this method, refer to the SuperUserReleaseEIB Method section. Refer to the Visual Basic or C++ ActiveX control samples Batch Execution/Samples/Controls folder for a code sample that displays the Emergency Release button with the EWI ActiveX control.

Skipping or Re-Executing Steps

In the EWI Display control, an operator can choose to skip an EWI or Logic Step. An operator can also re-execute an EWI Step. In order for the operator to do either, you must set up the signature requirements ahead of time. If you do not capture a Performed By or Verified By signature, the computer or login name is captured when an operator skips or re-executes a step. Skipped and re-executed steps are recorded in the History Log for that EWI, as illustrated in the Viewing the History section.

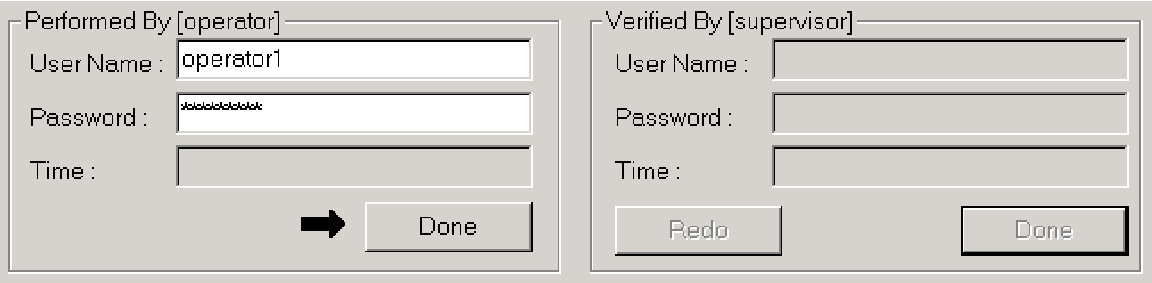
Signing from the EIB or Step Display Screen

To verify your actions during the execution of an EIB, EWI, or Logic Step you must enter the necessary electronic signatures required to complete the intended action. An electronic signature, in this case, is a user name and password. This user name and password is defined through Windows security. An EIB, EWI, or Logic Step screen can require no signature, one signature (from an operator or supervisor), or two signatures (from an operator and supervisor). The following list outlines the signature requirement types:

- **None - Performed:** no signature required. The Windows computer name or login name is captured, by default.
- **Performed By:** a single signature required, verified against the Performed By group.
- **Perform By/Verified By:** two distinct signatures required; the first signature is verified against Performed By group, and the second signature is verified against the Verified By group.

Example: Dual Signatures

If the EIB, EWI, or Logic Step requires two signatures, you first enter the user name and password in the Performed By group box. After you enter the password, an arrow appears next to the Done button, as shown in the following figure.



Performed By [operator]

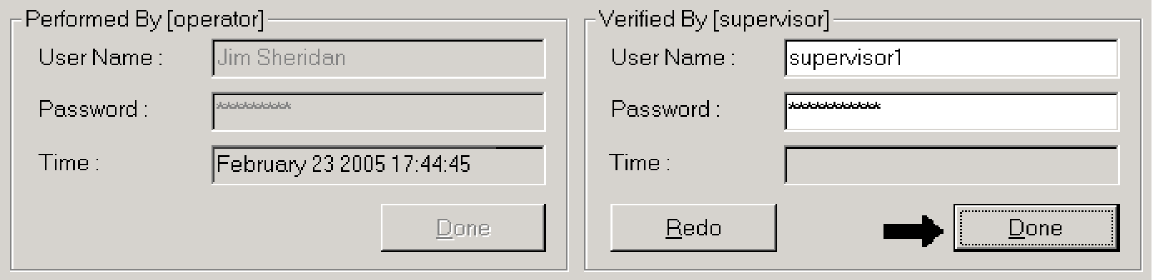
User Name : operator1

Password : *********

Operator Signature

To complete the signature for the EIB, EWI, or Logic Step, click the Done button. Similarly, after you enter the Performed By signature, a supervisor must enter the user name and password in the Performed By group box and click Done, as shown in the following figure.

NOTE: *The current time of the signature displays in the Time field and the full user name appears in the User Name field, when you click Done.*



Performed By [operator]

User Name : Jim Sheridan

Password : *********

Time : February 23 2005 17:44:45

Verified By [supervisor]

User Name : supervisor1

Password : ***********

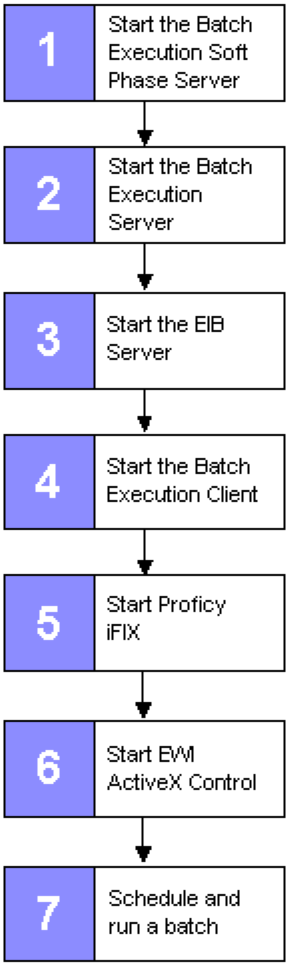
Supervisor Signature

It is important to note that not all EIBs, EWI, or Logic Steps require dual signatures. The number of signatures is dependent upon how your administrator configured the EIB, EWI, and Logic Steps.

NOTE: *The Windows security groups always display in brackets above the user name and password fields for all signatures. If no signature is required, then "Signature Not Required" displays within the brackets.*

Using WorkInstruction with Batch Execution

To begin using WorkInstruction with your Batch Execution system, once you have configured all the components, you need to start the applications. The following figure shows the order that you should use when starting your batch system.


1
Start the Batch Execution Soft Phase Server
2
Start the Batch Execution Server
3
Start the EIB Server
4
Start the Batch Execution Client
5
Start Proficy iFIX
6
Start EWI ActiveX Control
7
Schedule and run a batch

Overview of Starting WorkInstruction

Viewing EIBs

EIBs cannot be viewed from the Batch Execution Client. You can only view EIBs through the EWI ActiveX control. Refer to the Configuring the EWI ActiveX Control section for information on configuring the EWI control on your operator's computers.

Overview of Signing

You can record the following signature types upon completion of EIB, EWI, or Logic Step:

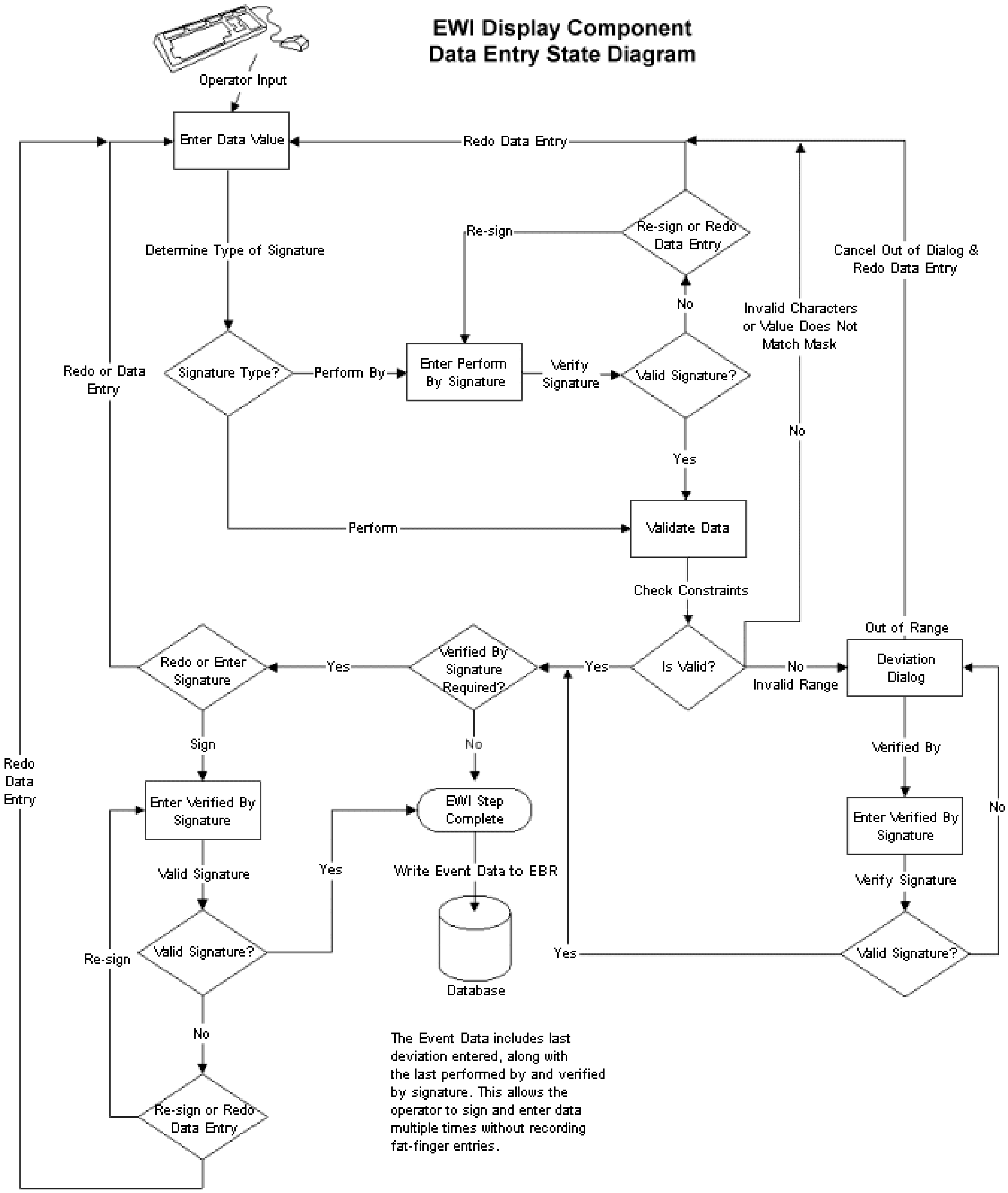
- None
- Performed By
- Performed By/Verified By

If no signature is required, an operator does not need to enter a user name and password to complete the step. WorkInstruction captures the Windows computer name or login name, the default signature.

The Performed By signature requires the signature of the operator entering the data. The user name and password of the operator is validated against the Windows security group. For example, if the Performed By security group was the Operator group, then the user and password would be validated against the Operator group.

The Performed By/Verified By signature requires both the signature of the operator and the supervisor. The Verified By signature must be different than the Performed By signature.

The following figure describes the signing process for WorkInstruction.



EWI Display Component Data Entry State Diagram

The Event Data includes last deviation entered, along with the last performed by and verified by signature. This allows the operator to sign and enter data multiple times without recording fat-finger entries.

EWI Display Component Data Enter State Diagram

Understanding the WorkInstruction Data Model

The WorkInstruction logical data model contains EWI data that is written to the relational database. This database can be SQL Server-based or Oracle-based and accessible through a standard ODBC connection.

Information contained within the WorkInstruction database tables can be accessed during system setup and during batch operations.

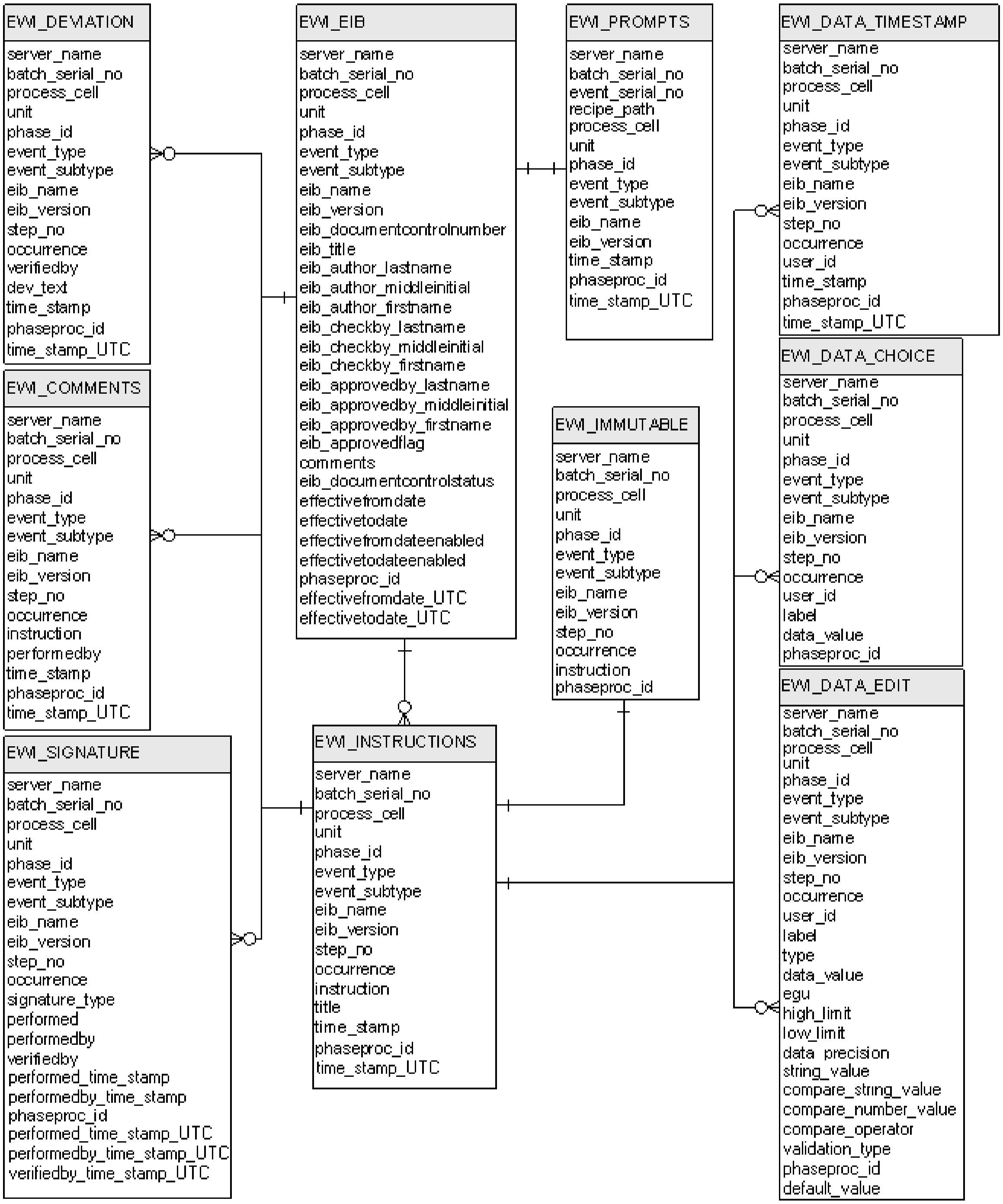
You create the WorkInstruction database tables by running a setup script. For instructions on running the setup script, refer to the Batch Execution System Configuration Manual. For instructions on running the database upgrade script, if you are upgrading from an earlier version of Batch Execution, refer to the Batch Execution Upgrade Guide.

Entity Relationship Diagram for EWIs

The WorkInstruction data model defines the structures and rules that represent the instructions, signatures, prompts, EIBs, and so on. The data model:

- Is not specific to a particular vendor's implementation.
- Represents the tables, table attributes, constraints, and relationships among the tables.

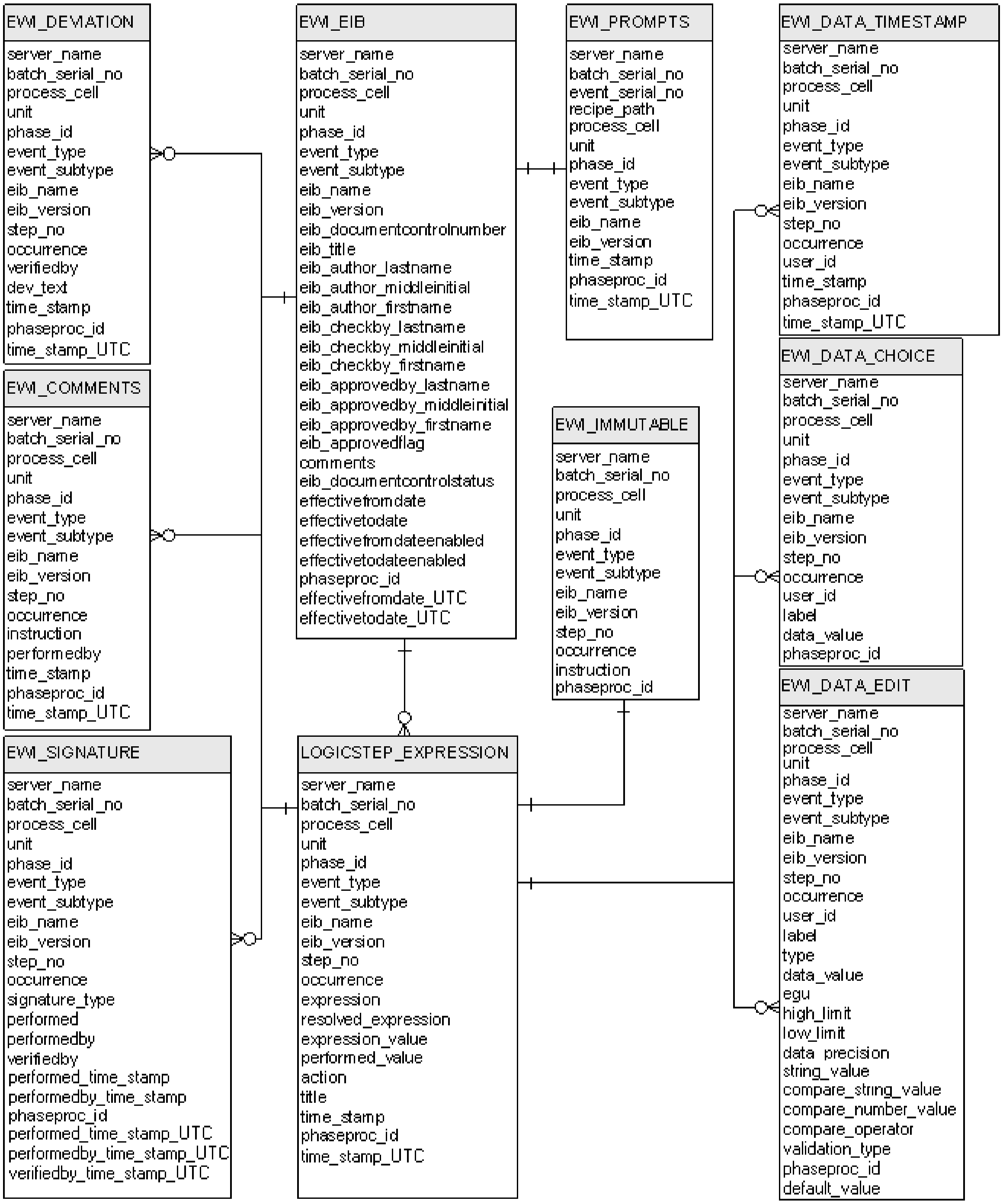
After you run the setup script for your relational database, the data model, shown in the following figure, is created. This data model includes the relational database tables you need to set up and use WorkInstruction with Batch Execution. The entity relationship diagram for logic steps is almost the same, except the EWI_INSTRUCTIONS table is replaced with the LOGICSTEP_EXPRESSION table. Refer to the Entity Relationship Diagram for Logic Steps section.



WorkInstruction EWI Entity Relationship Diagram

Entity Relationship Diagram for Logic Steps

The following figure describes the entity relationship diagram for logic steps. This table is the same as the EWI entity relationship diagram in the following figure, except that the EWI_INSTRUCTIONS table is replaced with the LOGICSTEP_EXPRESSION table.



WorkInstruction Logic Step Entity Relationship Diagram

Attribute Domains Described in Tables

An *attribute* is a fact or non-decomposable piece of information describing the content of a table. Each table in the WorkInstruction data model is comprised of one or more attributes. An *attribute domain* defines a set of valid values for an attribute. Domain characteristics define both generic characteristics and entity related characteristics. The generic characteristics describe the attributes' data type, length, allowable values, and meaning. The entity (table) characteristics define the attributes' uniqueness, null support, default value, and additional constraints.

The following table lists the characteristics associated with each attribute in the WorkInstruction data model:

Attribute Domain	**Value/Description**
Data Type	Integer, Real, String, or Date.
Length	The maximum number of characters or digits supported by the data type.
Allowable Values	The range of values supported.
Uniqueness	Unique or non-unique. By default, attributes are non-unique.
Null Support	Allowed or not allowed.
Key	Indicates whether the attribute is part of a primary, alternate, or foreign key.
Default Value	The attribute value if no value is supplied.
Description	Text describing the function of the attribute.
Constraints	Rules that enforce data integrity between and within tables.

The following sections describe each table in the WorkInstruction data model, using these attributes.

Event Types

Events are recorded by the EIB Server to the Electronic Batch Record, after completion and signing of the EWI and EIB using the EWI ActiveX control. The events are stored to SQL Server or Oracle database tables. The valid event types for the WorkInstruction tables are as follows:

- EIB_COMMENTS
- EIB_DEVIATION
- EIB_SIGNING
- EWI
- EWI_COMMENTS
- EWI_DATAENTRY
- EWI_DEVIATION

- EWI_EIB
- EWI_IMMUTABLE
- EWI_INSTRUCTION
- EWI_SIGNING
- LOGICSTEP_COMMENTS
- LOGICSTEP_DEVIATION
- LOGICSTEP_EXPRESSION

Subevent Types

Subevents are recorded by the EIB Server to the Electronic Batch Record, after completion and signing of the EWI and EIB.

Event and Subevent Mapping

The following table describes which subevents are associated with each EWI and EIB event. This table also specifies the Archiver tables in the relational database where the events and subevents are stored.

EWI Event and Subevent Mappings		
EWI Event Type	**EWI Subevent Type**	**Archiver Table**
EIB_COMMENTS	–	EWI_COMMENTS
EWI_COMMENTS	–	EWI_COMMENTS
LOGICSTEP_COMMENTS	–	EWI_COMMENTS
EIB_DEVIATION	–	EWI_DEVIATION
EWI_DEVIATION	–	EWI_DEVIATION
EWI_DEVIATION	EWI_SKIP	EWI_DEVIATION
EWI_DEVIATION	EWI_REEXECUTE	EWI_DEVIATION
EWI_DEVIATION	EWI_REEXECUTE_CANCEL	EWI_DEVIATION
LOGICSTEP_DEVIATION	–	EWI_DEVIATION
LOGICSTEP_EXPRESSION	–	LOGICSTEP_EXPRESSIONS

EWI Event and Subevent Mappings

EWI Event Type	**EWI Subevent Type**	**Archiver Table**
EIB_SIGNING	EIB_ACQUIRE	EWI_SIGNATURE
EIB_SIGNING	EIB_RELEASE	EWI_SIGNATURE
EWI_SIGNING	EWI_VERIFIEDBY	EWI_SIGNATURE
EWI_SIGNING	EWI_PERFORMED	EWI_SIGNATURE
EWI_SIGNING	EWI_PERFORMEDBY	EWI_SIGNATURE
EIB_SIGNING	EIB_VERIFIEDBY	EWI_SIGNATURE
EIB_SIGNING	EIB_PERFORMED	EWI_SIGNATURE
EIB_SIGNING	EIB_PERFORMEDBY	EWI_SIGNATURE
EWI_DATAENTRY	EWI_DATETIME	EWI_DATA_TIMESTAMP
EWI_DATAENTRY	EWI_EDITBOX	EWI_DATA_EDIT
EWI_DATAENTRY	EWI_CHECKBOX	EWI_DATA_CHOICE
EWI_DATAENTRY	EWI_RADIOBUTTON	EWI_DATA_CHOICE
EWI_IMMUTABLE	–	EWI_IMMUTABLE
EWI_INSTRUCTION	–	EWI_INSTRUCTIONS
EWI_EIB	EIB_HEADER	EWI_EIB
EWI	–	EWI_PROMPTS

Events, Subevents, and Their Actions

The following table describes the actions that occur for specific events and subevents.

Actions that Occur for EWI Event and Subevents		
EWI Event Type	**EWI Subevent Type**	**Occurs When...**
EIB_COMMENTS	–	You click the Comment button and enter comment text in the main EIB display.
EWI_COMMENTS	–	You click the Comment button and enter comment text in the EWI dialog box.
LOGICSTEP_COMMENTS	–	• You click the Comment button from the Logic Step dialog box and enter a comment. • The logic step action prompts you to enter a comment.
EIB_DEVIATION	–	You click the Deviation button on the main EIB display and enter a deviation.
EWI_DEVIATION	–	The data entry fails validation and you enter a deviation.
EWI_DEVIATION	EWI_SKIP	You click the Skip button on the main EIB display, the EWI dialog box, or the Logic Step dialog box, and then skip the step.
EWI_DEVIATION	EWI_REEXECUTE	You click the ReExecute button on the main EIB display for an EWI that is completed or skipped. A deviation dialog box must be completed.

Actions that Occur for EWI Event and Subevents		
EWI Event Type	**EWI Subevent Type**	**Occurs When...**
EWI_DEVIATION	EWI_REEXECUTE_CANCEL	You click the ReExecute button and then choose not to complete the EWI data entry. No deviation dialog box is displayed.
LOGICSTEP_DEVIATION	–	• You enter a deviation when the logic step action prompts you to. • You manually choose the logic step action from the Logic Step dialog box and the action does not match the logic step value.
LOGICSTEP_EXPRESSION	–	A logic step is completed.
EIB_SIGNING	EIB_ACQUIRE	You acquire an EIB.
EIB_SIGNING	EIB_RELEASE	You release an EIB, or perform the Emergency Release EIB action.
EWI_SIGNING	EWI_VERIFIEDBY	You sign the EWI dialog box with the "Verified By" signature.
EWI_SIGNING	EWI_PERFORMED	There are no EWI signatures required and you click the Done button for the EWI data entry.
EWI_SIGNING	EWI_PERFORMEDBY	You sign the EWI dialog box with the "Performed By" signature.
EIB_SIGNING	EWI_VERIFIEDBY	You sign the EIB dialog box with the "Verified By" signature.

Actions that Occur for EWI Event and Subevents		
EWI Event Type	**EWI Subevent Type**	**Occurs When...**
EIB_SIGNING	EWI_PERFORMED	There are no EIB signature requirements and you click the Done button for the EIB.
EIB_SIGNING	EWI_PERFORMEDBY	You sign the EIB with the "Performed By" signature.
EWI_DATAENTRY	EWI_DATETIME	You enter a value for the date/time field and completely sign the EWI.
EWI_DATAENTRY	EWI_EDITBOX	You enter a value for the edit or combo box field and completely sign the EWI.
EWI_DATAENTRY	EWI_CHECKBOX	You select or clear the check box and completely sign the EWI.
EWI_DATAENTRY	EWI_RADIOBUTTON	You select or deselect the radio button and completely sign the EWI.
EWI_IMMUTABLE	–	You complete the EWI step.
EWI_INSTRUCTION	–	You complete the EWI step.
EWI_EIB	EIB_HEADER	You complete the first EWI step of an EIB.
EWI	–	The Batch Execution Server generates an EIB prompt.

Tables in the WorkInstruction Data Model

The WorkInstruction data model contains the following:

- EWI_COMMENTS Table
- EWI_DATA_CHOICE Table
- EWI_DATA_EDIT Table

- EWI_DATA_TIMESTAMP Table
- EWI_DEVIATION Table
- EWI_EIB Table
- EWI_IMMUTABLE Table
- EWI_INSTRUCTIONS Table
- LOGICSTEP_EXPRESSIONS Table
- EWI_PROMPTS Table
- EWI_SIGNATURE Table

The following sections describe the tables in the WorkInstruction data model.

EWI_COMMENTS Table

The EWI_COMMENTS table stores operator comments for EWI steps, an EIB, or logic step. The source of the comment is indicated by the EVENT_TYPE. The following table lists the attributes for the EWI_COMMENTS table.

EWI_COMMENTS Table	
Attribute	**Attribute Domain**
SERVER_NAME	Data Type: String. Length: 64 characters. Null Support: Nulls not allowed. Allowable Values: Windows computer name. Default Value: None. Description: The computer name on which the EIB Server resides.
BATCH_SERIAL_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number generated by the Batch Execution Server. This number can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies an instance of a control recipe (a batch).

EWI_COMMENTS Table	
Attribute	**Attribute Domain**
PROCESS_CELL	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The process cell on which the batch executed.
UNIT	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The relevant unit.
PHASE_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The equipment phase instance name.
EIB_NAME	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The name of the EIB stored in the EWI.

EWI_COMMENTS Table	
Attribute	**Attribute Domain**
EIB_VERSION	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Any number, 0 or greater. A value of 0 means to use the most current version of the EIB. The greatest allowable value cannot be greater than the highest version for that EIB in the production store. You can view the latest version in the WorkInstruction Document Management dialog box in the WorkInstruction Editor. Default Value: 0. Description: EIB Version Number from the DMS (Document Management System), Microsoft Visual SourceSafe.
EVENT_TYPE	Data Type: String. Usually one of the following event types: EWI_COMMENT, EIB_COMMENTS, LOGICSTEP_COMMENTS, LOGICSTEP_COMMENTS, or EWI_COMMENTS. Length: 40 characters. Null Support: Nulls not allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Type of event.
EVENT_SUBTYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Classification of event type.

EWI_COMMENTS Table

Attribute	**Attribute Domain**
STEP_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies the step in an EIB.
OCCURRENCE	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A counter that defines the number of times that a logic step or an EWI (electronic work instruction) step executes. This counter increases by one for each EWI step completion, logic step completion, skip step, or not run event. The re-execution of an EWI step does not increase the counter. You cannot re-execute a logic step. An EWI or logic step can be executed more than once by logic step looping. A loop is treated as a new occurrence.
INSTRUCTION	Data Type: String. Length: 6500 characters (SQL Server), 4000 characters (Oracle). Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: Operator instructions prompted and stored electronically.
PERFORMEDBY	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: Name of the person who performed the EWI.

EWI_COMMENTS Table

Attribute	**Attribute Domain**
TIME_STAMP	Data Type: Date/time. Length: Not applicable. Null Support: Nulls not allowed. Allowable Values: Valid time stamp. Default Value: Current date and time. Description: Date and time the operator entered the comment.
PHASEPROC_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscores (_). Default Value: None. Description: The recipe phase name.
TIME_STAMP_UTC	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp, in UTC format. Default Value: Current date and time, in UTC format. Description: Date and time, in UTC format, that the operator entered the comment.

EWI_DATA_CHOICE Table

The EWI_DATA_CHOICE table stores radio, check boxes, and data entry values entered by the operator during the execution of an EWI. The following table lists the attributes for the EWI_DATA_CHOICE table.

EWI_DATA_CHOICE Table

Attribute	**Attribute Domain**
SERVER_NAME	Data Type: String. Length: 64 characters. Null Support: Nulls not allowed. Allowable Values: Windows computer name. Default Value: None. Description: The computer name on which the EIB Server resides.
BATCH_SERIAL_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number generated by the Batch Execution Server. This number can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies an instance of a control recipe (a batch).
PROCESS_CELL	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The process cell on which the batch executed.

EWI_DATA_CHOICE Table	
Attribute	**Attribute Domain**
UNIT	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The relevant unit.
PHASE_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and the underscore character. Default Value: None. Description: The equipment phase instance name.
EVENT_TYPE	Data Type: String. Length: 40 characters. Null Support: Nulls not allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Type of event for the journal entry.
EVENT_SUBTYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Classification of event type.

EWI_DATA_CHOICE Table	
Attribute	**Attribute Domain**
EIB_NAME	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and the underscore character. Default Value: None. Description: The name of the EIB stored in the EWI.
EIB_VERSION	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Any number, 0 or greater. 0 means use the most current version of the EIB. The greatest allowable value cannot be greater than the highest version for that EIB in the production store. You can view the latest version in the WorkInstruction Document Management dialog box in the WorkInstruction Editor. Default Value: 0. Description: EIB Version Number from the DMS (Document Management System), Microsoft Visual SourceSafe.
STEP_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies the step in an EIB.

EWI_DATA_CHOICE Table	
Attribute	**Attribute Domain**
OCCURRENCE	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A counter that defines the number of times that a logic step or an EWI (electronic work instruction) step executes. This counter increases by one for each EWI step completion, logic step completion, skip step, or not run event. The re-execution of an EWI step does not increase the counter. You cannot re-execute a logic step. An EWI or logic step can be executed more than once by logic step looping. A loop is treated as a new occurrence.
USER_ID	Data Type: String. Length: 255 characters. Null Support: Nulls allowed. Allowable Values: NULL or user name. Default Value: NULL. Description: The name of the user who acquired the EIB, or the default EIB user. See the EWI_SIGNATURE Table for the associated signatures.
LABEL	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and the underscore character. Default Value: None. Description: The EWI label.

EWI_DATA_CHOICE Table	
Attribute	**Attribute Domain**
DATA_VALUE	Data Type: String. Length: 5 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and the underscore character. Default Value: None. Description: The value of the data. The radio button fields record a value for each radio button, with the selected one having a DATA_VALUE of TRUE and the rest having a DATA_VALUE of FALSE.
PHASEPROC_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and the underscore character. Default Value: None. Description: The recipe phase name.

EWI_DATA_EDIT Table

The EWI_DATA_EDIT table stores alphanumeric, integer, or real data entry entered by the operator during the execution of an EWI. This includes IDO and non-IDO data entry. The following table lists the attributes for the EWI_DATA_EDIT table.

EWI_DATA_EDIT Table	
Attribute	**Attribute Domain**
SERVER_NAME	Data Type: String. Length: 64 characters. Null Support: Nulls not allowed. Allowable Values: Windows computer name. Default Value: None. Description: The computer name on which the EIB Server resides.

EWI_DATA_EDIT Table	
Attribute	**Attribute Domain**
BATCH_SERIAL_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number generated by the Batch Execution Server. This number can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies an instance of a control recipe (a batch).
PROCESS_CELL	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The process cell on which the batch executed.
UNIT	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The relevant unit.
PHASE_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The equipment phase instance name.

EWI_DATA_EDIT Table	
Attribute	**Attribute Domain**
EVENT_TYPE	Data Type: String. Length: 40 characters. Null Support: Nulls not allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Type of event for the journal entry.
EVENT_SUBTYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Classification of event type.
EIB_NAME	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The name of the EIB stored in the EWI.
EIB_VERSION	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Any number, 0 or greater. 0 means use the most current version of the EIB. The greatest allowable value cannot be greater than the highest version for that EIB in the production store. You can view the latest version in the WorkInstruction Document Management dialog box in the WorkInstruction Editor. Default Value: 0. Description: EIB Version Number from the DMS (Document Management System), Microsoft Visual SourceSafe.

EWI_DATA_EDIT Table	
Attribute	**Attribute Domain**
STEP_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies the step in an EIB.
OCCURRENCE	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A counter that defines the number of times that a logic step or an EWI (electronic work instruction) step executes. This counter increases by one for each EWI step completion, logic step completion, skip step, or not run event. The re-execution of an EWI step does not increase the counter. You cannot re-execute a logic step. An EWI or logic step can be executed more than once by logic step looping. A loop is treated as a new occurrence.
USER_ID	Data Type: String. Length: 255 characters. Null Support: Nulls allowed. Allowable Values: NULL or user name. Default Value: NULL. Description: The name of the user who acquired the EIB, or the default EIB user. See the EWI_SIGNATURE Table for the associated signatures.

EWI_DATA_EDIT Table	
Attribute	**Attribute Domain**
LABEL	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The EWI label.
TYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: String, Integer, and Real. Default Value: None. Description: The data type of EWI data (String, Integer, or Real).
DATA_VALUE	Data Type: Real. Length: If the type is Integer, then this number can be up to 14 digits. If the type is Real, then this number can be 38 digits of precision. If the type is String, then this field is not used and will have a 0 value. Null Support: Nulls allowed. Allowable Values: Integer, and Real. Default Value: None. Description: The value of the data.
EGU	Data Type: String. Length: 60 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The engineering units for the data value.

EWI_DATA_EDIT Table	
Attribute	**Attribute Domain**
HIGH_LIMIT	Data Type: Real. Length: If the type is Integer, then this number can be up to 14 digits. If the type is Real, then this number can be 38 digits of precision. If the type is String, then this field is not used and will have a 0 value. Null Support: Nulls allowed. Allowable Values: Integer, and Real. Default Value: None. Description: The high limit for the data entry value.
LOW_LIMIT	Data Type: Real. Length: If the type is Integer, then this number can be up to 14 digits. If the type is Real, then this number can be 38 digits of precision. If the type is String, then this field is not used and will have a 0 value. Null Support: Nulls allowed. Allowable Values: Integer, and Real. Default Value: None. Description: The low limit for the data entry value.
DATA_PRECISION	Data Type: String. Length: 20 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: A mask that consists of the "#" characters and an optional `.' character. The mask forces the data entry to conform to a pattern. For example, ###.## would indicate no more than 3 digits to the left and at least 2 digits to the right of the decimal point.

EWI_DATA_EDIT Table	
Attribute	**Attribute Domain**
STRING_VALUE	Data Type: String. Length: 100 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: If the value is String, then the user enters the value of the string.
COMPARE_NUMBER_VALUE	Data Type: Real. Length: If the type is Integer, then this number can be up to 14 digits. If the type is Real, then this number can be 38 digits of precision. If the type is String, then this field is not used and will have a 0 value. Null Support: Nulls allowed. Allowable Values: Integer, and Real. Default Value: 0. Description: The numeric value used for long or real data type comparisons.
COMPARE_STRING_VALUE	Data Type: String. Length: 100 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The string value used for comparison for alphanumeric data types.
COMPARE_OPERATOR	Data Type: String. Length: 5 characters. Null Support: Nulls allowed. Allowable Values: =, >, >=, <, or <=. Default Value: None. Description: The comparison operator such as the less than (<), greater than (>), or equal to (=) sign.

EWI_DATA_EDIT Table	
Attribute	**Attribute Domain**
VALIDATION_TYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: RANGE, COMPARISON, COMPARISON_CASE_SENSITIVE, or NONE. Default Value: None. Description: The type of field being evaluated: RANGE, COMPARISON, COMPARISON_CASE_SENSITIVE, or NONE.
PHASEPROC_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscores (_). Default Value: None. Description: The recipe phase name.
DEFAULT_VALUE	Data Type: String. Length: 100 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscores (_). Default Value: None. Description: The default value for the data entry field.

.

EWI_DATA_TIMESTAMP Table

The EWI_DATA_TIMESTAMP table stores the time stamp data entry value. The following table lists the attributes for the EWI_DATA_TIMESTAMP table.

EWI_DATA_TIMESTAMP Table	
Attribute	**Attribute Domain**
SERVER_NAME	Data Type: String. Length: 64 characters. Null Support: Nulls not allowed. Allowable Values: Windows computer name. Default Value: None. Description: The computer name on which the EIB Server resides.
BATCH_SERIAL_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number generated by the Batch Execution Server. This number can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies an instance of a control recipe (a batch).
PROCESS_CELL	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The process cell on which the batch executed.
UNIT	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The relevant unit.

EWI_DATA_TIMESTAMP Table	
Attribute	**Attribute Domain**
PHASE_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The equipment phase instance name.
EVENT_TYPE	Data Type: String. Length: 40 characters. Null Support: Nulls not allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Type of event for the journal entry.
EVENT_SUBTYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Classification of event type.
EIB_NAME	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The name of the EIB stored in the EWI.

EWI_DATA_TIMESTAMP Table	
Attribute	**Attribute Domain**
EIB_VERSION	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Any number, 0 or greater. 0 means use the most current version of the EIB. The greatest allowable value cannot be greater than the highest version for that EIB in the production store. You can view the latest version in the WorkInstruction Document Management dialog box in the WorkInstruction Editor. Default Value: 0. Description: EIB Version Number from the DMS (Document Management System), Microsoft Visual SourceSafe.
STEP_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies the step in an EIB.
OCCURRENCE	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A counter that defines the number of times that a logic step or an EWI (electronic work instruction) step executes. This counter increases by one for each EWI step completion, logic step completion, skip step, or not run event. The re-execution of an EWI step does not increase the counter. You cannot re-execute a logic step. An EWI or logic step can be executed more than once by logic step looping. A loop is treated as a new occurrence.

EWI_DATA_TIMESTAMP Table	
Attribute	**Attribute Domain**
USER_ID	Data Type: String. Length: 255 characters. Null Support: Nulls allowed. Allowable Values: NULL or user name. Default Value: NULL. Description: The name of the user who acquired the EIB, or the default EIB user. See the EWI_SIGNATURE Table for the associated signatures.
TIME_STAMP	Data Type: Date/time. Length: Not applicable. Null Support: Nulls not allowed. Allowable Values: Valid time stamp. Default Value: Current date and time. Description: Date and time the data entry value was entered by the operator.
PHASEPROC_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscores (_). Default Value: None. Description: The recipe phase name.
TIME_STAMP_UTC	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp, in UTC format. Default Value: Current date and time, in UTC format. Description: Date and time, in UTC format, that the data entry value was entered by the operator.

EWI_DEVIATION Table

The EWI_DEVIATION table stores deviations for EWIs. The deviation types include: EWI Deviation, EIB Deviation, Logic Step Deviation, EWI Skip, Logic Step Skip, Re-Execute, and Re-Execute Cancel. Operators account for deviations both automatically and manually. The following table lists the attributes for the EWI_DEVIATION table.

EWI_DEVIATION Table	
Attribute	**Attribute Domain**
SERVER_NAME	Data Type: String. Length: 64 characters. Null Support: Nulls not allowed. Allowable Values: Windows computer name. Default Value: None. Description: The computer name on which the EIB Server resides.
BATCH_SERIAL_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number generated by the Batch Execution Server. This number can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies an instance of a control recipe (a batch).
PROCESS_CELL	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The process cell on which the batch executed.
UNIT	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The relevant unit.

EWI_DEVIATION Table	
Attribute	**Attribute Domain**
PHASE_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The equipment phase instance name.
EVENT_TYPE	Data Type: String. Length: 40 characters. Null Support: Nulls not allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Type of event for the journal entry.
EVENT_SUBTYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Classification of event type.
EIB_NAME	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The name of the EIB stored in the EWI.

EWI_DEVIATION Table

Attribute	**Attribute Domain**
EIB_VERSION	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Any number, 0 or greater. 0 means use the most current version of the EIB. The greatest allowable value cannot be greater than the highest version for that EIB in the production store. You can view the latest version in the WorkInstruction Document Management dialog box in the WorkInstruction Editor. Default Value: 0. Description: EIB Version Number from the DMS (Document Management System), Microsoft Visual SourceSafe.
STEP_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies the step in an EIB.
OCCURRENCE	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A counter that defines the number of times that a logic step or an EWI (electronic work instruction) step executes. This counter increases by one for each EWI step completion, logic step completion, skip step, or not run event. The re-execution of an EWI step does not increase the counter. You cannot re-execute a logic step. An EWI or logic step can be executed more than once by logic step looping. A loop is treated as a new occurrence.

EWI_DEVIATION Table	
Attribute	**Attribute Domain**
VERIFIEDBY	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: Name of the person who approved the deviation.
DEV_TEXT	Data Type: String. Length: 6500 characters (SQL Server), 4000 characters (Oracle). Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: Informational text describing the deviation.
TIME_STAMP	Data Type: Date/time. Length: Not applicable. Null Support: Nulls not allowed. Allowable Values: Valid time stamp. Default Value: Current date and time. Description: Date and time the event occurred.
PHASEPROC_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscores (_). Default Value: None. Description: The recipe phase name.

EWI_DEVIATION Table	
Attribute	**Attribute Domain**
TIME_STAMP_UTC	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp, in UTC format. Default Value: Current date and time, in UTC format. Description: Date and time, in UTC format, that the event occurred.

EWI_EIB Table

The EWI_EIB table contains header data (all of the EIB document information) for all EIBs. The following table lists the attributes for the EWI_EIB table.

EWI_EIB Table	
Attribute	**Attribute Domain**
SERVER_NAME	Data Type: String. Length: 64 characters. Null Support: Nulls not allowed. Allowable Values: Windows computer name. Default Value: None. Description: The computer name on which the EIB Server resides.
BATCH_SERIAL_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number generated by the Batch Execution Server. This number can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies an instance of a control recipe (a batch).

EWI_EIB Table	
Attribute	**Attribute Domain**
PROCESS_CELL	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The process cell on which the batch executed.
UNIT	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The relevant unit.
PHASE_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The equipment phase instance.
EVENT_TYPE	Data Type: String. Length: 40 characters. Null Support: Nulls not allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Type of event for the journal entry.

EWI_EIB Table	
Attribute	**Attribute Domain**
EVENT_SUBTYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Classification of event type.
EIB_NAME	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The name of the EIB stored in the EWI.
EIB_VERSION	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Any number, 0 or greater. 0 means use the most current version of the EIB. The greatest allowable value cannot be greater than the highest version for that EIB in the production store. You can view the latest version in the WorkInstruction Document Management dialog box in the WorkInstruction Editor. Default Value: 0. Description: EIB Version Number from the DMS (Document Management System), Microsoft Visual SourceSafe.

EWI_EIB Table

Attribute	Attribute Domain
EIB_DOCUMENTCONTROLNUMBER	Data Type: String. Length: 30 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: User defined version or reference number for the EIB.
EIB_TITLE	Data Type: String. Length: 80 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: Title of the EIB.
EIB_AUTHOR_LASTNAME	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The last name of the EIB author.
EIB_AUTHOR_MIDDLEINITIAL	Data Type: String. Length: 3 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The middle initial of the EIB author.

EWI_EIB Table	
Attribute	**Attribute Domain**
EIB_AUTHOR_FIRSTNAME	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The first name of the EIB author.
EIB_CHECKBY_LASTNAME	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The last name of the person who verified the EIB.
EIB_CHECKBY_MIDDLEINITIAL	Data Type: String. Length: 3 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The middle initial of the person who verified the EIB.
EIB_CHECKBY_FIRSTNAME	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The first name of the person who verified the EIB.

EWI_EIB Table	
Attribute	**Attribute Domain**
EIB_APPROVEDBY_LASTNAME	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The last name of the person who approved the EIB.
EIB_APPROVEDBY_MIDDLEINITIAL	Data Type: String. Length: 3 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The middle initial of the person who approved the EIB.
EIB_APPROVEDBY_FIRSTNAME	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The first name of the person who approved the EIB.
EIB_APPROVEDFLAG	Data Type: String. Length: 5 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: Indicates whether the EIB has been approved.

EWI_EIB Table	
Attribute	**Attribute Domain**
COMMENTS	Data Type: String. Length: 4000 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: Description entered in the EIB header.
EIB_DOCUMENTCONTROLSTATUS	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The status of the EIB document.
EFFECTIVEFROMDATE	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp. Default Value: Current date and time. Description: Date and time from which the EIB is effective.
EFFECTIVEFROMDATEENABLED	Data Type: String. Length: 5 characters. Null Support: Nulls allowed. Allowable Values: TRUE or FALSE. Description: Flag indicating whether the from effective date option is enabled.

EWI_EIB Table	
Attribute	**Attribute Domain**
EFFECTIVETODATE	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp. Default Value: Current date and time. Description: Date and time to which the EIB is effective.
EFFECTIVETODATEENABLED	Data Type: String. Length: 5 characters. Null Support: Nulls allowed. Allowable Values: TRUE or FALSE. Description: Flag indicating whether the to effective date option is enabled.
PHASEPROC_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscores (_). Default Value: None. Description: The recipe phase name.
EFFECTIVEFROMDATE_UTC	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp, in UTC format. Default Value: Current date and time, in UTC format. Description: Date and time, in UTC format, from which the EIB is effective.

EWI_EIB Table	
Attribute	**Attribute Domain**
EFFECTIVETODATE_UTC	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp, in UTC format. Default Value: Current date and time, in UTC format. Description: Date and time, in UTC format, to which the EIB is effective.

EWI_IMMUTABLE Table

The EWI_IMMUTABLE table captures all events, such as the data entry, signature, deviations, and comment actions by an operator for each EWI. The following table lists the attributes for the EWI_IMMUTABLE table.

EWI_IMMUTABLE Table	
Attribute	**Attribute Domain**
SERVER_NAME	Data Type: String. Length: 64 characters. Null Support: Nulls not allowed. Allowable Values: Windows computer name. Default Value: None. Description: The computer name on which the EIB Server resides.
BATCH_SERIAL_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number generated by the Batch Execution Server. This number can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies an instance of a control recipe (a batch).

EWI_IMMUTABLE Table	
Attribute	**Attribute Domain**
PROCESS_CELL	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The process cell on which the batch executed.
UNIT	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The relevant unit.
PHASE_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and the underscore character. Default Value: None. Description: The equipment phase instance name.
EVENT_TYPE	Data Type: String. Length: 40 characters. Null Support: Nulls not allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Type of event for the journal entry.

EWI_IMMUTABLE Table

Attribute	**Attribute Domain**
EVENT_SUBTYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Classification of event type.
EIB_NAME	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and the underscore character. Default Value: None. Description: The name of the EIB stored in the EWI.
EIB_VERSION	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Any number, 0 or greater. 0 means use the most current version of the EIB. The greatest allowable value cannot be greater than the highest version for that EIB in the production store. You can view the latest version in the WorkInstruction Document Management dialog box in the WorkInstruction Editor. Default Value: 0. Description: EIB Version Number from the DMS (Document Management System), Microsoft Visual SourceSafe.
STEP_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies the step in an EIB.

EWI_IMMUTABLE Table	
Attribute	**Attribute Domain**
OCCURRENCE	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A counter that defines the number of times that a logic step or an EWI (electronic work instruction) step executes. This counter increases by one for each EWI step completion, logic step completion, skip step, or not run event. The re-execution of an EWI step does not increase the counter. You cannot re-execute a logic step. An EWI or logic step can be executed more than once by logic step looping. A loop is treated as a new occurrence.
INSTRUCTION	Data Type: String. Length: 6500 characters (SQL Server), 4000 characters (Oracle). Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and the underscore character. Default Value: None. Description: All operator instructions for this EIB.
PHASEPROC_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and the underscore character. Default Value: None. Description: The recipe phase name.

EWI_INSTRUCTIONS Table

The EWI_INSTRUCTIONS table captures the instruction descriptions for each EWI when the step is completed. The following table lists the attributes for the EWI_INSTRUCTIONS table.

EWI_INSTRUCTIONS Table	
Attribute	**Attribute Domain**
SERVER_NAME	Data Type: String. Length: 64 characters. Null Support: Nulls not allowed. Allowable Values: Windows computer name. Default Value: None. Description: The computer name on which the EIB Server resides.
BATCH_SERIAL_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number generated by the Batch Execution Server. This number can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies an instance of a control recipe (a batch).
PROCESS_CELL	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The process cell on which the batch executed.
UNIT	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The relevant unit.

EWI_INSTRUCTIONS Table	
Attribute	**Attribute Domain**
PHASE_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The equipment phase instance name.
EVENT_TYPE	Data Type: String. Length: 40 characters. Null Support: Nulls not allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Type of event for the journal entry.
EVENT_SUBTYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Classification of event type.
EIB_NAME	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The name of the EIB stored in the EWI.

EWI_INSTRUCTIONS Table

Attribute	Attribute Domain
EIB_VERSION	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Any number, 0 or greater. 0 means use the most current version of the EIB. The greatest allowable value cannot be greater than the highest version for that EIB in the production store. You can view the latest version in the WorkInstruction Document Management dialog box in the WorkInstruction Editor. Default Value: 0. Description: EIB Version Number from the DMS (Document Management System), Microsoft Visual SourceSafe.
STEP_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies the step in an EIB.
OCCURRENCE	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A counter that defines the number of times that a logic step or an EWI (electronic work instruction) step executes. This counter increases by one for each EWI step completion, logic step completion, skip step, or not run event. The re-execution of an EWI step does not increase the counter. You cannot re-execute a logic step. An EWI or logic step can be executed more than once by logic step looping. A loop is treated as a new occurrence.

EWI_INSTRUCTIONS Table	
Attribute	**Attribute Domain**
INSTRUCTION	Data Type: String. Length: 6500 characters (SQL Server), 4000 characters (Oracle). Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: EWI instruction text displayed to the operator.
TITLE	Data Type: String. Length: 80 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The title of the EWI.
TIME_STAMP	Data Type: Date/time. Length: Not applicable. Null Support: Nulls not allowed. Allowable Values: Valid time stamp. Default Value: Current date and time. Description: Date and time the event occurred.
PHASEPROC_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscores (_). Default Value: None. Description: The recipe phase name.

EWI_INSTRUCTIONS Table	
Attribute	**Attribute Domain**
TIME_STAMP_UTC	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp, in UTC format. Default Value: Current date and time, in UTC format. Description: Date and time, in UTC format, that the event occurred.

LOGICSTEP_EXPRESSIONS Table

The LOGICSTEP_EXPRESSIONS Table stores the logic step events.

LOGICSTEP_EXPRESSIONS Table	
Attribute	**Attribute Domain**
SERVER_NAME	Data Type: String. Length: 64 characters. Null Support: Nulls not allowed. Allowable Values: Windows computer name. Default Value: None. Description: The computer name on which the EIB Server resides.
BATCH_SERIAL_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number generated by the Batch Execution Server. This number can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies an instance of a control recipe (a batch).

LOGICSTEP_EXPRESSIONS Table	
Attribute	**Attribute Domain**
PROCESS_CELL	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The process cell on which the batch executed.
UNIT	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The relevant unit.
PHASE_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The equipment phase instance name.
EVENT_TYPE	Data Type: String. Length: 40 characters. Null Support: Nulls not allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Type of event for the journal entry.

LOGICSTEP_EXPRESSIONS Table	
Attribute	**Attribute Domain**
EVENT_SUBTYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Classification of event type.
EIB_NAME	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The name of the EIB.
EIB_VERSION	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Any number, 0 or greater. 0 means use the most current version of the EIB. The greatest allowable value cannot be greater than the highest version for that EIB in the production store. You can view the latest version in the WorkInstruction Document Management dialog box in the WorkInstruction Editor. Default Value: 0. Description: EIB Version Number from the DMS (Document Management System), Microsoft Visual SourceSafe.
STEP_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies the step in an EIB.

LOGICSTEP_EXPRESSIONS Table	
Attribute	**Attribute Domain**
OCCURRENCE	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A counter that defines the number of times that a logic step or an EWI (electronic work instruction) step executes. This counter increases by one for each EWI step completion, logic step completion, skip step, or not run event. The re-execution of an EWI step does not increase the counter. You cannot re-execute a logic step. An EWI or logic step can be executed more than once by logic step looping. A loop is treated as a new occurrence.
EXPRESSION	Data Type: String. Length: 3000 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The original expression as it was typed into the Logic Step Editor dialog box in the WorkInstruction Editor.
RESOLVED_EXPRESSION	Data Type: String. Length: 3000 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The expression after the field codes, tags, enumerations, or strings are replaced with values. Any variable that cannot be evaluated is replaced with <UNRESOLVED VALUE>.

LOGICSTEP_EXPRESSIONS Table	
Attribute	**Attribute Domain**
EXPRESSION_VALUE	Data Type: String. Length: 20 characters. Null Support: Nulls not allowed. Allowable Values: TRUE, FALSE, or FAILURE. Default Value: None. Description: The TRUE, FALSE, or FAILURE value that the resolved expression evaluates to.
PERFORMED_VALUE	Data Type: String. Length: 20 characters. Null Support: Nulls not allowed. Allowable Values: TRUE or FALSE. Default Value: None. Description: When the logic step processes, it normally performs the TRUE action if the expression is TRUE or the FALSE action if the expression is FALSE. However, you can also override this action. For instance, if there is a FAILURE value, you can override it and choose the TRUE or FALSE action instead.
ACTION	Data Type: String. Length: 50 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The action as defined in the EWI Logic Expression Builder in the WorkInstruction Editor. These actions include the GOTO #, Comment, Deviation, Message, No Action, and End EIB actions.
TITLE	Data Type: String. Length: 80 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The title of the logic step.

LOGICSTEP_EXPRESSIONS Table	
Attribute	**Attribute Domain**
TIME_STAMP	Data Type: Date/time. Length: Not applicable. Null Support: Nulls not allowed. Allowable Values: Valid time stamp. Default Value: Current date and time. Description: Date and time the event occurred.
PHASEPROC_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscores (_). Default Value: None. Description: The recipe phase name.
TIME_STAMP_UTC	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp, in UTC format. Default Value: Current date and time, in UTC format. Description: Date and time, in UTC format, that the event occurred.

EWI_PROMPTS Table

The EWI_PROMPTS table captures a list of EWI prompts. The following table lists the attributes for the EWI_PROMPTS table.

EWI_PROMPTS Table	
Attribute	**Attribute Domain**
SERVER_NAME	Data Type: String. Length: 64 characters. Null Support: Nulls not allowed. Allowable Values: Windows computer name. Default Value: None. Description: The computer name on which the EIB Server resides.
BATCH_SERIAL_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number generated by the Batch Execution Server. This number can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies an instance of a control recipe (a batch).
EVENT_SERIAL_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number generated by the Batch Execution Server for each batch and for each event. Default Value: 1. Description: Used to distinguish between multiple events occurring at the same time.
RECIPE_PATH	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The directory path of the recipe.

EWI_PROMPTS Table	
Attribute	**Attribute Domain**
PROCESS_CELL	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The process cell on which the batch executed.
UNIT	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The relevant unit.
PHASE_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The equipment phase instance name.
EVENT_TYPE	Data Type: String. Length: 40 characters. Null Support: Nulls not allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Type of event for the journal entry.

EWI_PROMPTS Table

Attribute	**Attribute Domain**
EVENT_SUBTYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: System-defined. Default Value: None. Description: Classification of event type.
EIB_NAME	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The name of the EIB.
EIB_VERSION	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Any number, 0 or greater. 0 means use the most current version of the EIB. The greatest allowable value cannot be greater than the highest version for that EIB in the production store. You can view the latest version in the WorkInstruction Document Management dialog box in the WorkInstruction Editor. Default Value: 0. Description: EIB Version Number from the DMS (Document Management System), Microsoft Visual SourceSafe.
TIME_STAMP	Data Type: Date/time. Length: Not applicable. Null Support: Nulls not allowed. Allowable Values: Valid time stamp. Default Value: Current date and time. Description: Date and time the event occurred.

EWI_PROMPTS Table	
Attribute	**Attribute Domain**
PHASEPROC_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscores (_). Default Value: None. Description: The recipe phase name.
TIME_STAMP_UTC	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp, in UTC format. Default Value: Current date and time, in UTC format. Description: Date and time, in UTC format, that the event occurred.

EWI_SIGNATURE Table

The EWI_SIGNATURE table captures signatures for each EWI and EIB. These signatures stored include: EIB Acquire, EIB Release, EIB Performed, EIB Performed By, EIB Verified By, EWI Performed, EWI Performed By, and EWI Verified By. It does not capture signatures for comments and deviations since they get stored in their own tables. The following table lists the attributes for the EWI_SIGNATURE table.

EWI_SIGNATURE Table	
Attribute	**Attribute Domain**
SERVER_NAME	Data Type: String. Length: 64 characters. Null Support: Nulls not allowed. Allowable Values: Windows computer name. Default Value: None. Description: The computer name on which the EIB Server resides.

EWI_SIGNATURE Table	
Attribute	**Attribute Domain**
BATCH_SERIAL_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number generated by the Batch Execution Server. This number can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies an instance of a control recipe (a batch).
PROCESS_CELL	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The process cell on which the batch executed.
UNIT	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, the underscore character, and the colon character. Default Value: None. Description: The relevant unit.
PHASE_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The equipment phase instance name.

EWI_SIGNATURE Table	
Attribute	**Attribute Domain**
EVENT_TYPE	Data Type: String. Length: 40 characters. Null Support: Nulls not allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Type of event for the journal entry.
EVENT_SUBTYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: System-defined. Refer to the Event and Subevent Mapping section for a list. Default Value: None. Description: Classification of event type.
EIB_NAME	Data Type: String. Length: 255 characters. Null Support: Nulls not allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The name of the EIB stored in the EWI.

EWI_SIGNATURE Table

Attribute	**Attribute Domain**
EIB_VERSION	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Any number, 0 or greater. 0 means use the most current version of the EIB. The greatest allowable value cannot be greater than the highest version for that EIB in the production store. You can view the latest version in the WorkInstruction Document Management dialog box in the WorkInstruction Editor. Default Value: 0. Description: EIB Version Number from the DMS (Document Management System), Microsoft Visual SourceSafe.
STEP_NO	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A unique ID that identifies the step in an EIB.

EWI_SIGNATURE Table	
Attribute	**Attribute Domain**
OCCURRENCE	Data Type: Integer. Length: 10 digits. Null Support: Nulls not allowed. Allowable Values: Sequential number that can range from 0 to 2147483647. Default Value: None. Description: A counter that defines the number of times that a logic step or an EWI (electronic work instruction) step executes. This counter increases by one for each EWI step completion, logic step completion, skip step, or not run event. The re-execution of an EWI step does not increase the counter. You cannot re-execute a logic step. An EWI or logic step can be executed more than once by logic step looping. A loop is treated as a new occurrence.
SIGNATURE_TYPE	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: The type of signature: • Performed • Performed By • Verified By (implies Performed By and Verified By) • Acquire • Release

EWI_SIGNATURE Table

Attribute	**Attribute Domain**
PERFORMED	Data Type: String. Length: 255 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: If the signature type is Performed, this field is the default signature of the EWI (computer name or login user name). Else, this attribute is Null.
PERFORMEDBY	Data Type: String. Length: 255 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: If this is a Performed By or Verified By signature type, then this field is the name of the operator who performed and signed the EWI. Otherwise, if this is an Acquire or Release signature type, then this field is the name of the operator who acquired or released the EIB.
VERIFIEDBY	Data Type: String. Length: 255 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscore (_). Default Value: None. Description: If the signature type is Verified By, then this field is the name of the supervisor who approved and signed the EWI. Otherwise, this field is NULL.

EWI_SIGNATURE Table	
Attribute	**Attribute Domain**
PERFORMED_TIME_STAMP	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp. Default Value: Current date and time. Description: Date and time the event was performed. This attribute is Null if it represents a Performed By, Verified By, Acquire, or Release signature type.
PERFORMEDBY_TIME_STAMP	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp. Default Value: Current date and time. Description: Date and time the event was performed and signed.
VERIFIEDBY_TIME_STAMP	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp. Default Value: Current date and time. Description: Date and time the event was verified and signed.
PHASEPROC_ID	Data Type: String. Length: 40 characters. Null Support: Nulls allowed. Allowable Values: a-z, A-Z, 0-9, and underscores (_). Default Value: None. Description: The recipe phase name.

EWI_SIGNATURE Table	
Attribute	**Attribute Domain**
PERFORMED_TIME_STAMP_UTC	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp, in UTC format.. Default Value: Current date and time, in UTC format.. Description: Date and time, in UTC format, that the event was performed. This attribute is Null if it represents a Performed By, Verified By, Acquire, or Release signature type.
PERFORMEDBY_TIME_STAMP_UTC	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp, in UTC format. Default Value: Current date and time, in UTC format. Description: Date and time, in UTC format, that the event was performed and signed.
VERIFIEDBY_TIME_STAMP_UTC	Data Type: Date/time. Length: Not applicable. Null Support: Nulls allowed. Allowable Values: Valid time stamp, in UTC format. Default Value: Current date and time, in UTC format. Description: Date and time, in UTC format, that the event was verified and signed.

Resolving Client/Server Connection Issues

When connecting a WorkInstruction Client to a WorkInstruction Server you may encounter connection issues that are related to local and remote security settings. Two types of connection issues commonly encountered include:

- The WorkInstruction Client running the EWI ActiveX control fails to connect to the remote WorkInstruction Server; an access denied message appears when you try to connect to the WorkInstruction Server.
- The WorkInstruction Client connects to the remote WorkInstruction Server, but does not receive any updates. The EIBs that you acquired or released from the WorkInstruction Server computer do not display an updated status on the remote WorkInstruction Client computer running the EWI ActiveX control.

For more information on how to troubleshoot the issues listed above, see the System Configuration manual.

WorkInstruction Editor Dialog Boxes

The Work Instruction Editor application includes the following dialog boxes (listed in alphabetical order):

- Add FieldCode Dialog Box
- Database Info For Audit Trail Dialog Box
- Database Info For SqlTest Dialog Box
- Date Time Page Dialog Box
- Default Signature Template Dialog Box
- Edit Page Dialog Box
- EIB Header Dialog Box
- EWI Editor Dialog Box
- EWI Logic Expression Builder Dialog Box
- Field Code Editor Dialog Box
- FieldCode IDs Dialog Box
- History Dialog Box
- Log On Information Dialog Box
- Logic Step Editor Dialog Box
- ODBC Data Sources Dialog Box
- Performed By Dialog Box
- Radio Button Page Dialog Box

- SourceSafe Configuration Dialog Box
- Step Selection Dialog Box
- Verification Status Results Dialog Box
- Windows Security Groups Dialog Box
- WorkInstruction Document Management System Dialog Box

Add FieldCode Dialog Box

The Add FieldCode dialog box displays the following items:

Type

Select the type of field code that you want to add: unit tag, phase parameter, report parameter, recipe step, SQL statement, hyperlink or user defined.

ID

Enter the ID value for the field code.

Description

Enter a description for the field code.

Data Link

Enter the data link value for the field code.

Data Type

Select a data type for the field code from the drop-down list: String, Real, Integer, Enumeration, or Boolean. Some notes on data types include:

- Use the Boolean data type only with User Defined field codes.
- Use the String data type for unit tags or phase parameters that are defined as Enumeration data types in the Proficy Batch Execution WorkSpace.
- Only use the Enumeration data type for field codes that you want to manually type in the instruction text field in the EWI Editor dialog box or add in the EWI Logic Expression Builder that is launched from the Logic Step Editor dialog box.

Database Info For Audit Trail Dialog Box

The Database Info For Audit Trail dialog box displays the following items:

Use Signature Verification for DMS commands

Select this check box if you want the user to enter a user name and password every time he performs a DMS function (check in, check out, and so on). Each time a user enters a user name and password from the WorkInstruction Database Management System dialog box, it is tracked in the Audit Trail.

Data Source Name

Enter the name of the ODBC data source DSN.

User Name

Enter the user name for the ODBC data source DSN.

Password

Enter the password for the user.

Database Info For SqlTest Dialog Box

The Database Info For SqlTest dialog box displays the following items:

Data Source Name

Enter the name of the ODBC data source DSN.

User Name

Enter the user name for the ODBC data source DSN.

Password

Enter the password for the user.

Date Time Page Dialog Box

The Date Time Page dialog box displays the following items:

Automatic Entry

Select this option for the date to be automatically set to the system time.

Manual Entry

Select this option to allow the user to select the date.

Default Signature Template Dialog Box

The Default Signature Template dialog box displays the following items:

Use Signature Template Defaults

Click this check box to enable default signature requirements to appear when configuring your EIBs, EWIs, and Logic Steps.

EIB Signing Group

The following table lists the contents of the EIB Signing group:

Item	**Description**
EIB Signature Requirements	Select this check box if you require a default Performed By, Verified By, or Performed By/Verified By electronic signatures to complete the EIB.
Performed By check box	Select this check box if you require a default Performed By electronic signature to complete the EIB.
Performed By text box	Enter the default Windows group name to which the operators (Performed By group) belong, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character. During execution of an EIB, only users who are members of this group can electronically sign the EIB as Verified By. ***NOTE:*** *The same user can not sign both the Performed By and Verified By signature requirements. The full user name for Windows is used when comparing signatures.*

Item	**Description**
Verified By check box	Select this check box if you require a default Verified By electronic signature to complete the EIB.
Verified By text box	Enter the default Windows group name to which the supervisors belong, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character. During execution of an EIB, only users who are members of this group can electronically sign the EIB as Verified By. ***NOTE:*** *The same user can not sign both the Performed By and Verified By signature requirements. The full user name for Windows is used when comparing signatures.*
Release Override check box	Select this check box if you require the operator to enter a default electronic signature before performing an emergency release of this EIB.
Release Override text box	Enter the default Windows group name to which the users who can perform emergency overrides belong, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Default Name for Unchecked Properties	Select the default name (Windows login name or computer name) that is captured if no other signatures are required for the EIB.
Acquire check box	Select this check box if you require the operator to enter a default electronic signature to acquire this EIB.
Acquire text box	Enter the default Windows group name to which the users who can acquire EIBs belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Release check box	Select this check box if you require the operator to enter a default electronic signature to release this EIB.
Release text box	Enter the default Windows group name to which the users who can release EIBs belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.

Item	Description
Deviation check box	Select this check box if you require the operator to enter a default electronic signature to perform deviations on this EIB.
Deviation text box	Enter the default Windows group name to which the users who can perform deviations belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Comments check box	Select this check box if you require the operator to enter a default electronic signature to add comments to this EIB.
Comments text box	Enter the Windows group name to which the default users who can enter comments for EIBs belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Skip check box	Select this check box if you require the operator to enter a default electronic signature to skip this EIB.
Skip text box	Enter the default Windows group name to which the users who can perform the skip function belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
ReExecute check box	Select this check box if you require the operator to enter a default electronic signature to re-execute this EIB.
ReExecute text box	Enter the default Windows group name to which the users who re-execute EIBs belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.

EWI Signing Group

The following table lists the contents of the EWI Signing group:

Item	Description
EWI Signature	Select this check box if you require a default Performed By, Verified By, or Performed By/Verified By electronic signatures to complete the EWI.

Item	Description
Performed By check box	Select this check box if you require a default Performed By electronic signature to complete the EWI.
Performed By text box	Enter the default Windows group name to which the operators belong, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character. During execution of an EIB, only users who are members of this group can electronically sign the EIB as Performed By . ***NOTE:*** *The same user can not sign both the Performed By and Verified By signature requirements. The full user name for Windows is used when comparing signatures.*
Verified By check box	Select this check box if you require a default Verified By electronic signature to complete the EWI.
Verified By text box	Enter the default Windows group name to which the supervisors belong, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character. During execution of an EIB, only users who are members of this group can electronically sign the EIB as Verified By. ***NOTE:*** *The same user can not sign both the Performed By and Verified By signature requirements. The full user name for Windows is used when comparing signatures.*
Default Name for Unchecked Properties	Select the default name (Windows login name or computer name) that is captured if no other signatures are required for the EWI.
Deviation check box	Select this check box if you require the operator to enter a default electronic signature to perform deviations on this EWI.
Deviation text box	Enter the default Windows group name to which the users who can perform deviations to EWIs belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Comments check box	Select this check box if you require the operator to enter a default electronic signature to add comments this EWI.

Item	Description
Comments text box	Enter the default Windows group name to which the users who can add comments to EWIs belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.

Logic Step Signing Group

The following table lists the contents of the Logic Step Signing group:

Item	Description
Default Name for Unchecked Properties	Select the default name (Windows login name or computer name) that is captured if no other signatures are required for the Logic Step.
Deviation check box	Select this check box if you require the operator to enter a default electronic signature to perform deviations to this Logic Step.
Deviation text box	Enter the default Windows group name to which the users who can perform deviations to Logic Steps belong, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Comments check box	Select this check box if you require the operator to enter a default electronic signature to add comments to this Logic Step.
Comments text box	Enter the default Windows group name to which the users who can add comments to Logic Steps belong, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.

Edit Page Dialog Box

The Edit Page dialog box displays the following items:

Data Type Group

The following table lists the contents of the Data Type group:

Item	Description
Type	Select a data type from the drop-down list.
EGU Enabled	Select this check box to enable the operator to enter the engineering units (EGUs) with the data entry value in run-time.

Intelligent Verification Objects Group

The following table lists the contents of the Intelligent Verification Objects group:

Item	Description
Precision	Select this check box if you want to enter a mask for the data entry.
Mask	If you select Real or Integer for the data entry type, you can keep the default mask or specify up to 12 characters for the data entry mask. If you select Real for the data type, you can also enter a decimal point. The character positions in the data entry mask map to a placeholder for a digit that the user can input. For example: if you select Real, the mask is set to ####.### by default. Each # symbol represents a digit the user can enter.
Compare	Click this check to enable data comparison for the data entry.
Operator	Select a comparison operator from the drop-down list. For an Alphanumeric values you can only select the equals (=) sign. For Real or Integer values you can select =, <>, <, >, <= or >=.
Value	Enter the value that you want to compare the data entry value with.
Compare Value Case Sensitive	Select this check box if the alpha numeric comparison is case sensitive.

Item	Description
Range	Select this check box if you want to enter limits to constrain the data entered by the operator. Data values that are outside this range cause a deviation to be generated at run-time.
Low	If you enabled a range for the data value, enter the low value for the range. You can enter up to 20 digits, including 0-9, the decimal point, and the dash (-) character. When the user enters a value at run-time, WorkInstruction checks if the value is within the high-low range. If the value is outside that range, an automatic deviation is generated and a dialog box appears.
High	If you enabled a range for the data value, enter the high value for the range. You can enter up to 20 digits, including 0-9, the decimal point, and the dash (-) character. When the user enters a value at run-time, WorkInstruction checks if the value is within the high-low range. If the value is outside that range, an automatic deviation is generated and a dialog box appears.

Intelligent Display Objects Group

The following table lists the contents of the Intelligent Display Objects group:

Item	Description
Type	Select the type of value that you want the operator to enter for the intelligent display object.
Type Edit Text	Enter a value.
Test SQL	Use this button to check the validity of the SQL statement you entered in the Dynamic IDO field in the Edit Page dialog box.
Read	Click to enter a readable value.
Write	Click to enter a value that you can write to.
ReadWrite	Click to enter a value that you can read or write to.

Item	**Description**
Static Values	Click Insert to add static IDO values. The format of the values entered is dependent up on whether you selected Alphanumeric, Integer, or Real for the data type. An IDO is a suggested value. You can enter up to 50 characters, including A-Z, 0-9, and the underscore character. At run-time an operator can use the suggested value or enter another value in this data entry field.
Insert	Use this command to add a value.
Modify	Use this command to modify a static IDO value.
Delete	Use this command to delete a value.

EIB Header Dialog Box

The EIB Header dialog box displays the following items:

Identification Group

The following table lists the contents of the Identification group:

Item	**Description**
EIB ID	Displays the file name of the EIB, as a read-only field.
EIB Version	Displays the version of the EIB, as a read-only field.
Document Control Number	Enter the document control number for the EIB, if required. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
EIB Title	Enter the title of the EIB. You can enter up to 40 characters, including A-Z, 0-9, and the underscore character.

Revision Control Group

The following table lists the contents of the Revision Control group:

Item	Description
Author - First Name	Enter the first name of the EIB author. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Author - Middle Initial	Enter the middle initial of the EIB author. You can enter up to 3 characters, including A-Z, 0-9, and the underscore character.
Author - Last Name	Enter the last name of the EIB author. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Verified By - First Name	Enter the first name of the user required to verify completion of the EIB. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Verified By - Middle Initial	Enter the middle initial of the user required to verify completion of the EIB. You can enter up to 3 characters: A-Z, 0-9, or the underscore character.
Verified By - Last Name	Enter the last name of the user required to verify completion of the EIB. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Approved By - First Name	Enter the first name of the user required to approve completion of the EIB. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Approved By - Middle Initial	Enter the middle initial of the user required to approve completion of the EIB. You can enter up to 3 characters: A-Z, 0-9, or the underscore character.
Approved By - Last Name	Enter the last name of the user required to approve completion of the EIB. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Effective From Date	Select this check box if you want the EIB to be effective starting on a particular date.
Effective Until Date	Select this check box if you want the EIB to be effective only until the specified date.

Item	Description
Document Control Status	Enter the document control status, if required. For example: working, released, or obsolete. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.

Signature Requirements Group

The following table lists the contents of the Signature Requirements group:

Item	Description
EIB Signature Requirements	Select this check box if you require a Performed By, Verified By, or both Performed By and Verified By signatures to complete this EIB.
Performed By check box	Select this check box to require a Performed By electronic signature.
Performed By text box	Enter the Windows group name to which the operators belong, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character. During execution of an EIB, only users who are members of this group can electronically sign the EIB as Performed By and enter operator comments. ***NOTE:*** *The same user can not sign both the Performed By and Verified By signature requirements. The full user name for Windows is used when comparing signatures.*
Verified By check box	Enter the Windows group name to which the users who can release EIBs belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.

Item	Description
Verified By text box	Enter the Windows group name to which the supervisors belong, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character. During execution of an EIB, only users who are members of this group can electronically sign the EIB as Verified By and enter operator comments. ***NOTE:*** *The same user can not sign both the Performed By and Verified By signature requirements. The full user name for Windows is used when comparing signatures.*
Release Override check box	Select this check box if you require the operator to enter an electronic signature to perform an emergency release of this EIB.
Release Override text box	Enter the Windows group name to which the users who can perform emergency releases belong to, or click Browse to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Default Name for Unchecked Properties	Select the default name (Windows login name or computer name) that is captured if no other signatures are required for the EIB.
Acquire check box	Select this check box if you require the operator to enter an electronic signature to acquire this EIB.
Acquire text box	Enter the Windows group name to which the users who can acquire EIBs belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Release check box	Select this check box if you require the operator to enter an electronic signature to release this EIB.
Release text box	Enter the Windows group name to which the users who can release EIBs belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Deviation check box	Select this check box if you require the operator to enter an electronic signature to add a deviation to this EIB, EWI, or Logic Step.

Item	Description
Deviation text box	Enter the Windows group name to which the users who can perform deviations on EIBs belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Comments check box	Select this check box if you require the operator to enter an electronic signature to add a comment to this EIB, EWI, or Logic Step.
Comments text box	Select this check box if you require the operator to enter an electronic signature to add a comment to this EIB, EWI, or Logic Step.
Skip check box	Select this check box if you require the operator to enter an electronic signature to skip this EIB.
Skip text box	Enter the Windows group name to which the users who can skip EIBs belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
ReExecute check box	Select this check box if you require the operator to enter an electronic signature in order to re-execute this EIB.
ReExecute text box	Enter the Windows group name to which the users who can re-execute EIBs belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.

Description

Optionally, enter a description for the EIB. You can enter up to 1024 characters, including A-Z, 0-9, and the underscore character.

EWI Editor Dialog Box

The EWI Editor dialog box displays the following items:

Title

Enter the title for the EWI. You can enter up to 80 characters, including A-Z, 0-9, and the underscore character.

Step

Read-only field that describes the number of the EWI in the sequence.

Group

Describes the group number to which the EWI or Logic Step belongs.

Instruction Text

The following table lists the contents of the Instruction text box:

Item	**Description**
Instruction Text	Enter instructional text. To change the font of all or part of the instructional text, select the text and click on the Font button below this edit box. To add a field code to the instructional text, select a field code from the drop-down and click the Add below this edit box. ***NOTE:*** *Although the WorkInstruction Editor allows you to format text within a field code tag, do not format only part of the tag text. Either format the whole tag including the % symbols, or none of the tag.*
Field Codes	Select a field code from the drop-down list and click add to include the field code in the instructional text. You can also right click anywhere in the text box to bring up the Field Code IDs dialog box. ***NOTE:*** *Although the WorkInstruction Editor allows you to format text within a field code tag, do not format only part of the tag text. Either format the whole tag including the % symbols, or none of the tag.* Make sure that you manually enter Phase Parameters or Unit Tags that are Enumeration types directly into the text box. For example, enter UnitTag.ABC or PhaseParameter.ABC as the value. You cannot use the Field Code IDs dialog box.

Item	Description
Add	Click this button to add a field code for use within the EIB.
Font	Click this button to change the font of the selected text in the Instruction Text edit box.

DataEntry

The following table lists the contents of the DataEntry group:

Item	Description
Type	Select a type for the data entry: check box, radio button, date/time, edit box, or no data entry.
Details	Click to enter data entry requirements for the edit box, radio button, or date/time fields.
Label	Enter the label that you want to appear with the data entry field.

RunTask

The following table lists the contents of the RunTask group:

Item	Description
Executable	Enter the name and location of the executable file that allow the EWI operator to launch, or click the Browse (…) button to select a file.
Runtime Caption	Enter the name that you want to appear on the button that launches the run task executable.

Signature Requirements

The following table lists the contents of the Signature Requirements group:

Item	**Description**
EWI Signature	Select this check box to require a Performed By, Verified By, or both Performed By and Verified By signatures to complete the EWI.
Performed By check box	Select this check box if you require a Performed By electronic signature to complete this EWI.
Performed By text box	Enter the Windows group name to which the operators belong, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character. During execution of an EIB, only users who are members of this group can electronically sign the EIB as Performed By and enter operator comments. ***NOTE:*** *The same user can not sign both the Performed By and Verified By signature requirements. The full user name for Windows is used when comparing signatures.*
Default Name for Unchecked Properties	Select the default name (Windows login name or computer name) that is captured if no other signatures are required for the EIB.
Verified By check box	Select this check box to require a Verified By electronic signature to complete the EWI.
Verified By text box	Enter the Windows group name to which the supervisors belong, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character. During execution of an EIB, only users who are members of this group can electronically sign the EIB as Verified By and enter operator comments. ***NOTE:*** *The same user can not sign both the Performed By and Verified By signature requirements. The full user name for Windows is used when comparing signatures.*
Comment check box	Select this check box if you require the operator to enter an electronic signature to add comments to this EWI.
Comment text box	Enter the Windows group name to which the users who can add comments to the EWI belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.

Item	Description
Deviation check box	Select this check box if you require the operator to enter an electronic signature to perform a deviation on this EWI.
Deviation text box	Enter the Windows group name to which the users who can perform deviations to the EWI belong to, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.

Previous

Click this button to go back to the previous EWI or Logic Step.

Next

Click this button to proceed to the next EWI or Logic Step in the list.

EWI Logic Expression Builder Dialog Box

The EWI Logic Expression Builder dialog box displays the following items:

Current Transition Status

Displays the Boolean condition of the current transition. To change the condition, edit the text in the field.

Buttons Group

The following table lists the contents of the Buttons group:

Item	Description
+	Click this button to add a plus sign (+) to the Boolean condition.
-	Click this button to add a minus sign (-) to the Boolean condition.
*	Click this button to add a multiplication sign (*) to the Boolean condition.

Item	Description
/	Click this button to add a division sign (/) to the Boolean condition.
<	Click this button to add a less than sign (<) to the Boolean condition.
<=	Click this button to add a less than or equal sign (<=) to the Boolean condition.
>	Click this button to add a greater than sign (>) to the Boolean condition.
>=	Click this button to add a greater than or equal sign (>=) to the Boolean condition.
=	Click this button to add an equal sign (=) to the Boolean condition.
<>	Click this button to add a not equal to sign (<>) to the Boolean condition.
(	Click this button to add a left parenthesis [( ] to the Boolean condition. Expressions within parentheses are evaluated before expressions outside of parentheses.
)	Click this button to add a right parenthesis [ )] to the Boolean condition. Expressions within parentheses are evaluated before expressions outside of parentheses.
Not	Click this button to add a NOT operator to the Boolean condition.
And	Click this button to add an AND operator to the Boolean condition.
Or	Click this button to add an OR operator to the Boolean condition.
Paste	Click this button to paste text from the clipboard into the Boolean condition.
Undo	Use this button to undo the changes you have made to the transition.

Category List

Displays the categories of items or specifies parameters/attributes you can add to a Boolean condition. Double-click the parameter or attribute to add it to the Boolean condition.

Operator List

Displays the specific operators or the type of data selected. Double-click an item in the list box to add it to the Boolean condition.

Field Code Editor Dialog Box

The Field Code Editor dialog box displays the following items:

Field Code List

This grid area lists all of the defined field codes. The following table lists the contents of each row in the field code list:

Item	**Description**
Fieldcode Type	Lists the type of field code that you want to add: unit tag, phase parameter, report parameter, recipe step, SQL statement, hyperlink or user defined.
ID	Lists the ID value for the field code.
Description	Lists a description for the field code.
Data Link	Lists the data link value for the field code.
Data Type	Lists a data type for the field code from the drop-down list: String, Real, Integer, Enumeration, or Boolean.

Add

Click to add a new field code to the list for this EIB.

Modify

Click this button to edit a field code. You cannot edit a reserved field code.

Delete

Click this button to delete a field code. You cannot delete a reserved field code.

FieldCode IDs Dialog Box

The FieldCode IDs dialog box displays the following item:

Field Code IDs List Box

Select a field code ID from the list.

NOTE: *You cannot access Phase Parameters or Unit Tags that are Enumeration types, directly from this dialog box. Instead, you must manually enter these IDs into the text box or edit field. For example, enter UnitTag.ABC or PhaseParameter.ABC as the value in the text box or edit field.*

History Dialog Box

The History dialog box displays the following items:

Revision History List Box

Displays the revision history for an EIB in production

Log On Information Dialog Box

The Log On Information dialog box displays the following items:

User Name

Enter the user name that you want to log in to the DMS.

Password

Enter the password for the user.

Logic Step Editor Dialog Box

The Logic Step Editor dialog box displays the following items:

Title

Enter the title for the Logic Step. You can enter up to 80 characters, including A-Z, 0-9, and the underscore character.

Step

Read-only value that indicates the numerical order of the step in the list.

Group

Describes the group number to which the EWI or Logic Step belongs.

Logic Step Group

The following table lists the contents of the Logic Step group:

Item	**Description**
Expression	Read-only field that displays the expression evaluated in the logic step.
Build Expression	Click to open the Expression Editor.
Action On True	Select the action from the drop-down list that you want to perform the when the logic expression evaluates as true. You can also select No Action from the drop-down list.
Action On True - Step	Enter a value for the action to use when it evaluates the expression as true.
Action On False	Select the action from the drop-down list that you want to perform the when the logic expression evaluates as false. You can also select No Action from the drop-down list.
Action On False - Step	Enter a message for the action to use when it evaluates the expression as false.

Security Groups

The following table lists the contents of the Security Groups:

Item	**Description**
Default Name for Unchecked Properties	Select the default name (Windows login name or computer name) that is captured if no other signatures are required for the EIB.
Comments	Select this check box if you require the operator to enter an electronic signature to add a comment to this EIB, EWI, or Logic Step.

Item	Description
Comments Text	Enter the Windows group name to which the operators belong, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character. During execution of an EIB, only users who are members of this group can electronically sign the EIB as Performed By and enter operator comments. ***NOTE:*** *The same user can not sign both the Performed By and Verified By signature requirements. The full user name for Windows is used when comparing signatures.*
Deviation	Select this check box if you require the operator to enter an electronic signature to add a deviation to this EIB, EWI, or Logic Step.
Deviation Text	Enter the Windows group name to which the supervisors belong, or click the Browse (…) button to select a Windows security group. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character. During execution of an EIB, only users who are members of this group can electronically sign the EIB as Verified By and enter operator comments. ***NOTE:*** *The same user can not sign both the Performed By and Verified By signature requirements. The full user name for Windows is used when comparing signatures.*

Previous

Click this button to go back to the previous EWI or Logic Step.

Next

Click this button to proceed to the next EW0I or Logic Step in the list.

ODBC Data Sources Dialog Box

The ODBC Data Sources dialog box displays the following item:

Data Source List

Select a data source from the list and click OK.

Performed By Dialog Box

The Performed By dialog box displays the following items:

Action

The name of the action being performed: Enable Authority Check.

User Name

The name of the user performing the action. The name you supply here is from the iEsigAdministrators Windows user group.

Password

The password for the user performing the action. The name you supply here is from the iEsigAdministrators Windows user group.

Comment

Optionally, enter comments about the action.

Radio Button Page Dialog Box

The Radio Button Page dialog box displays the following items:

Number of Buttons

Use the spin control next to this field to select the number of radio buttons that you want to display within the EWI.

Labels Group

The following table lists the contents of the Labels group:

Item	Description
1	Enter the label that you want to appear next to the first radio button. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
2	Enter the label that you want to appear next to the second radio button. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.

Item	Description
3	Enter the label that you want to appear next to the third radio button. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
4	Enter the label that you want to appear next to the fourth radio button. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
5	Enter the label that you want to appear next to the fifth radio button. You can enter up to 30 characters, including A-Z, 0-9, and the underscore character.
Browse button (...)	Click this button to add a field code, phase parameter, phase report parameter, unit tag, or SQL command.

SourceSafe Configuration Dialog Box

The SourceSafe Configuration dialog box displays the following items:

Work Directory

Displays the local directory path where WorkInstruction stores the EIBs that you are working on.

Source Safe Path

Enter the path where your Microsoft SourceSafe database resides. For example, for the EWI demo the SourceSafe Path is:

```
C:\Program Files\Proficy\Proficy Batch
Execution\Projects\EWIDEMO\EWISS\srcsafe.ini
```

Step Selection Dialog Box

The Step Selection dialog box displays the following items:

EWI Step

Select this option to create an EWI Step.

Logic Step

Select this option to create a Logic Step.

Verification Status Results Dialog Box

The Verification Status Results dialog box displays the following item:

Results Area

Lists the results of the verification.

Windows Security Groups Dialog Box

The Windows Security Groups dialog box displays the available Windows security groups. Each row in the table includes the following items:

Location

The computer name.

Groups

The name of the Windows group.

Description

A description of the specified Windows group.

WorkInstruction Document Management System Dialog Box

The WorkInstruction Document Management System dialog box displays the following items:

Work Group

The following table lists the contents of the Work group:

Item	Description
EIB List	This box displays a list of the current versions of all the EIBs in the EWIWork project in the Document Management System (DMS).
Read	Click this button to read down a copy of the selected EIB to your computer from the DMS (Document Management System).
Check In	Click this button to check the selected EIB into the Document Management System (DMS).

Item	Description
Check Out	Click this button to check out the selected EIB from the Document Management System (DMS).
Add	Use this command to add an EIB (.XML file) to the Work Project into the Document Management System.
Delete	Use this command to delete a value.
Undo Check Out	Click this button to undo the check-out of the selected file from the Document Management System (DMS).
Copy to Production	Select an EIB in the work list and click this button to copy it to the EWIProduction project in the Document Management System (DMS).
Refresh	Click this button to refresh the working EIB list.

Production Group

The following table lists the contents of the Production group:

Item	Description
EIB List	This box displays a list of all the EIBs in the EWIProduction project in the Document Management System (DMS).
Move to Obsolete Project	Select an EIB in the production list and click this button to move it to the EWIObsolete project in the Document Management System (DMS).
History	Click this button to display the history for the selected EIB.
Refresh	Click this button to refresh the EIBs in production list.

Show Obsolete

Click this button to display the EIBs in the EWIObsolete project in the Document Management System (DMS).

Obsolete Group

The following table lists the contents of the Obsolete group:

Item	Description
EIB List	This box displays a list of all the EIBs in the EWIObsolete project in the Document Management System (DMS).
Delete	Click this button to delete the selected EIB from the EWIObsolete project in the Document Management System (DMS).
Refresh	Click this button to refresh the list of obsolete EIBs.

Hide Obsolete

Click this button to hide the list of EIBs in the EWIObsolete project in the Document Management System (DMS).

Local Working Directory

Displays the local directory path where WorkInstruction stores the EIBs that you are working on.

EIB Server Manager Dialog Boxes

The EIB Server Manager application includes the following dialog boxes for managing the EIB Server:

- EIB Server Choose Boot Method Dialog Box
- EIB Server Manager Dialog Box

EIB Server Choose Boot Method Dialog Box

The EIB Server Choose Boot Method dialog box displays the following items:

Warm

Click this button to return the EIB Server to its last known state. EWIs and EIBs are returned to the state that existed prior to the termination of the server. Only use the warm boot method to recover data after an unexpected loss of the EIB Server.

Cold

Click this button to start the EIB Server with all EWIs and EIBs in the Idle state. This is the preferred method of starting the EIB Server.

Cancel

Click to close this dialog box without starting the EIB Server.

EIB Server Manager Dialog Box

The EIB Server Manager dialog box displays the following items:

EIB Server Control Group

The following table lists the contents of the EIB Server Control group:

Item	**Description**
Status	Displays the current status of the server: Stopping, Stopped, Starting, Running.
Connections	Displays the number of client connections to the EIB server.
Icon	Shows the status of the Batch Server.
Start Server	Click to perform a cold start of the EIB Server.
Stop Server	Click this button to stop the EIB Server. If you close the EIB Server Manager without clicking Stop, the EIB Server will remain running, even with the application window closed.
Help	Click to access the online help for the EIB Server.

Message Box

Lists status messages about client connections to the server.

Clear Messages

Click this button to clear the server messages shown in the list box.

How Do I...

The following sections explain how to work with the WorkInstruction Editor:

- Using the WorkInstruction Editor
- Managing EIBs in the DMS
- Using the EIB Server Manager

Using the WorkInstruction Editor

For information on using WorkInstruction Editor, refer to the following sections:

- Configuring a Logic Step
- Configuring an EWI Step
- Configuring radio buttons
- Configuring the date/time data entry
- Configuring the Edit box
- Defining default signature requirements
- Defining the local working directory
- Selecting a data source
- Selecting a step type
- Working with EIBs
- Working with Field Codes

Defining the Local Working Directory and SourceSafe Path

▶To define the local working directory:

1. In the WorkInstruction Editor, on the Options menu, click SourceSafe Configuration. The SourceSafe Configuration dialog box appears.
2. In the Work Directory field, enter or browse to the local directory (outside of SourceSafe) that you want to store the EIBs that you are working on.
3. In the Source Safe Path field, enter or browse to the directory where your Microsoft SourceSafe database resides.
4. Click OK. You return to the SourceSafe Configuration dialog box.
5. Click OK.

Selecting a Data Source

▶To select a data source:

1. In the WorkInstruction Editor, open the Database Info dialog box. To open this dialog box:
    - If you are configuring a DSN for the audit trail, on the Audit Trail menu, click the Login Info option. The Database Info for Audit Trail dialog box appears.
    - If you are configuring a DSN to use for field codes or dynamic IDOs, on the Options menu, click the Database Login Info option. The Database Info for SqlTest dialog box appears.
2. In the Data Source Name field, enter the name of the database source field or click the Browse (…) button to select a database.
3. Enter a user name and a password.
4. If you are configuring a data source for audit trail, select the Use Signature Verification for DMS Commands option.

    The Performed By dialog box appears requesting a user name and password. Enter a user name and password from the iESigAdministrators Windows security group.
5. Click OK.

Defining the Default Signature Requirements

▶To define the default signature requirements:

1. In the WorkInstruction Editor, on the Options menu, click Default Signature Templates. The Default Signature Template dialog box appears.
2. Select the Use Signature Template Defaults check box.
3. In the EIB, EWI, and Logic Step Signing group boxes:
    - Select the check boxes of the commands for which you want to define default signature requirements.
    - Select the security groups for these commands.
    - Select LoginUserName or ComputerName from Default Name for Unchecked Properties drop-down list.
4. Click OK.

Configuring an EWI Step

▶To configure an EWI Step:

1. In the WorkInstruction Editor, add or edit an EWI Step to display the EWI Editor dialog box.
2. In the EWI Editor dialog box, in the Title field, enter a title.
3. In the Instructional Text field, enter instructional text:
    - To change the font of all or part of the instructional text, select the text and click on the Font button below this edit box.

- To add a field code to the instructional text, select a field code from the drop-down and click the Add below this edit box. Although the WorkInstruction Editor allows you to format text within a field code tag, do not format only part of the tag text. Either format the whole tag including the % symbols, or none of the tag.

3. Select the type of data entry.
4. Click the Details button to configure the data entry fields, if applicable.
5. Optionally, enter a task to run when the EWI is run, and a label.
6. Select the signature requirements: Performed, Performed By, or Performed By/Verified By.
7. Enter the Windows security groups for the operators (Performed By group) and supervisors (Verified By group).
8. In the Default Name for Unchecked Properties drop-down list, select the Windows login user name or computer name.
9. Optionally, select the Comment or Deviation check box if you want a signature to be captured when either action is performed.
10. Enter the Windows security group from which you want to capture an electronic signature for the comment or deviation.
11. Click OK.

Configuring a Logic Step

▶To configure a Logic Step

1. In the WorkInstruction Editor, add or edit a Logic Step to display the Logic Step Editor dialog box.
2. In the Logic Step Editor dialog box, in the Title field, enter a title.
3. Click the Build Expression button to enter an expression.
4. Select an action or no action to perform when the expression evaluates as true (from the Action on True drop-down list).
5. Select an action or no action to perform when the expression evaluates as false (from the Action on False drop-down list).
6. If an action is selected in step 4 or 5, enter any values required in the text box next to each field.
7. Select the Windows login user name or computer name in the Default Name for Unchecked Properties drop-down list.
8. Optionally, select the Comment or Deviation check box if you want a signature to be captured when either action is performed.
9. Enter the Windows security group from which you want to capture an electronic signature for the comment or deviation.
10. Click OK.

Configuring an Expression in the Expression Editor

▶To configure an expression:

1. In the WorkInstruction Editor, create a Logic Step.
2. In the Logic Step dialog box, click the Build Expression button. The EWI Logic Expression Builder dialog box appears.
3. In the expression group box, type your expression.
4. Use the row of buttons in the middle of the dialog box to add operators to a condition.
5. Use the list boxes at the bottom of the dialog box to select the items you want to add from a list. When using these list boxes:
    - You select an item from the left list box.
    - The categories of the selected item appear in the middle list box.
    - The items in the selected category appear in the right list box.
    - You can add an item to the condition at the top of the dialog box by double-clicking the item from the right list box.
    - If you see a plus sign next to a folder in the left list box, there are items inside the folder.
6. Click OK when you are finished creating your expression.

Configuring Radio Buttons

▶To configure radio buttons:

1. In the WorkInstruction Editor, add or edit an EWI Step to display the EWI Editor dialog box.
2. In the EWI Editor dialog box, in the Data Entry field, select Radio Buttons.
3. Click Details. The Radio Button Page dialog box appears.
4. Use the spin control next to the Number of Buttons field to select the number of radio buttons that you want to display within the EWI.

    The number of available Label fields updates as you change the number that displays in this field.
5. In the Labels group box, enter a label for each radio button.
6. Click OK.

Configuring the Date/Time Data Entry

▶To configure the date/time data entry:

1. In the WorkInstruction Editor, add or edit an EWI Step to display the EWI Editor dialog box.
2. In the EWI Editor dialog box, in the Data Entry field, select Date Time.
3. Click Details. The Date Time Page dialog box appears.
4. Select Manual or Automatic Entry for the date/time selection option.
5. Click OK.

Configuring the Edit Box

▸To configure the Edit box:

1. In the WorkInstruction Editor, add or edit an EWI Step to display the EWI Editor dialog box.
2. In the EWI Editor dialog box, in the Data Entry field, select Edit Box.
3. Click Details. The Edit Page dialog box appears.
4. From the Type drop-down list, select a data type.
5. If you want to allow the operator to enter engineering units at runtime, select the EGU check box.
6. Optionally, in the Intelligent Verification objects group box, configure Precision, Comparison, and Range properties, as required.
7. Optionally, in the Intelligent Display objects group box, select the type of IDO from the Type drop-down list and configure the values.
   - If you select Static IDO, insert the static values using the buttons to the right.
   - Otherwise, define the Dynamic IDO, Unit Tag, Phase Parameter, or Report Parameter below the Type drop-down list.

   NOTE: *You cannot browse for Phase Parameters or Unit Tags that are enumeration types directly from the FieldCode IDs dialog box. Instead, you must manually enter these IDs into the edit field. For example, enter UnitTag.ABC or PhaseParameter.ABC as the value in the edit field.*
8. Click OK.

Selecting a Step Type

▸To select a step type:

1. In the WorkInstruction Editor, double-click an empty row in the spreadsheet. The Step Selection dialog box appears.
2. Select either the EWI Step or Logic Step option.
3. Click OK.

Viewing the Audit Log

▸To view the audit log:

1. In the WorkInstruction Editor, on the Options menu, click Audit Reporter. The Proficy Batch Execution Audit Reporter application appears.
2. From the Proficy Batch Execution Audit Reporter, on the Tools menu, click Options to select a DSN. The Application Options dialog box appears.
3. Enter the DSN name, user name, and password.
4. Click OK.
5. In the Audit Reporter, open the Report Template dialog box to configure the columns that you want to display.

6. In the Audit Reporter, on the Report menu, click Run. The WorkInstruction audit trail appears in the Audit Report application.
7. Look for WorkInstruction Editor in the AuditApplication column.

   NOTE: *In order to view audit trail information for the WorkInstruction Editor, the Use Signature Verification for DMS Commands option must be enabled in the Database Info for Audit Trail dialog box (from the Audit Trail menu in the WorkInstruction Editor). Once it is enabled, you can begin to track DMS activities.*

Working with EIBs

For information on working with EIBs in the WorkInstruction Editor, refer to the following sections:

- Adding an EWI or Logic Step
- Entering EIB header information
- Rearranging the order of the EWIs within an EIB
- Running two EWIs to run in parallel
- Updating an EIB
- Verifying an EIB

Adding an EWI or Logic Step

▶To add an EWI or Logic Step:

1. In the WorkInstruction Editor, on the Step menu, click Add EWI Step or Logic Step. The EWI Editor dialog box or Logic Step Editor dialog box appears.
2. Configure the EWI or Logic step.
3. Click OK.

Entering EIB Header Data

▶To enter EIB header information:

1. In the WorkInstruction Editor, on the EIB menu, click Header. The EIB Header dialog box appears.
2. In the Document Control Number field, enter a document control number.
3. In the EIB Title field, enter a title.
4. Enter revision control information. Place the cursor in the field and press F1 for more information on the field.
5. Select the desired signature requirements, and enter the Windows group name where appropriate. Place the cursor in the field and press F1 for more information on the field.
6. Select the Windows login user name or computer name in the Default Name for Unchecked Properties drop-down list.
7. Optionally, in the Description field, enter a description for the EIB.
8. Click OK.

Rearranging the Order of the EWIs within an EIB

▶To rearrange the order of the EWIs within an EIB:

1. In the WorkInstruction Editor, in the spreadsheet, select the EWI you want to move.
2. Use the up and down arrows in the toolbar to move the EWI to the desired location.
3. Repeat steps 1-2 for each EWI you want to move.
4. On the Step menu, click Rearrange Groups to refresh the group numbers listed in the main screen.

Running Two EWIs in Parallel

▶To run two EWIs in parallel:

1. In the WorkInstruction Editor, in the spreadsheet, double-click one of the EWIs that you want to run in parallel. The EWI Editor screen appears.
2. Enter a group number in the Group field in the top right corner of the screen.
3. Click OK.
4. Double-click the next EWI that you want to run in parallel.
5. Click OK.
6. On the Step menu, click Rearrange Groups to refresh the group numbers listed in the main screen.

Updating an EIB that is in the Work Project

▶To update an EIB:

1. Check out the EIB from the DMS.
2. Make any changes to the EIB Header fields.
3. Double-click an EWI of Logic Step in the grid. The EWI Editor or Logic Step Editor dialog box appears.
4. Make your changes to your EWI or Logic steps.
5. Click OK.
6. Check in the EIB back into the DMS.

Verifying an EIB

▶To verify an EIB:

1. In the WorkInstruction Editor, on the EIB menu, click Verify. The Verification Status Results dialog box appears.
2. Click OK.

Working with Field Codes

For information on working with field codes in the WorkInstruction Editor, refer to the following sections:

- Adding a field code
- Editing a field code

Adding a Field Code

▶To add a field code:

1. In the WorkInstruction Editor, on the Field Codes menu, click Field Code Editor. The Field Code Editor dialog box appears.
2. Click the Add. The Add Field Code dialog box appears.
3. Select a tag from the drop-down list: unit tag, phase parameter, report parameter, user defined, step, SQL statement, or hyperlink.

   NOTE: *When configuring a unit tag, do* not *use the name of the unit tag. Make sure that you use the unit tag's class name.*

4. Enter an ID.
5. Enter a description for the field code.
6. Select a data type from the Data Type drop-down list: Real, Integer, String, Enumeration, or Boolean.
7. Click OK.

NOTES:

- *Use the Boolean data type only with User Defined field codes. The Proficy Batch Execution product does not support Boolean data types.*
- *Use the String data type for unit tags or phase parameters that are defined as Enumeration data types in the Proficy Batch Execution WorkSpace. The Edit Page dialog box in the WorkInstruction Editor does not support Enumeration data types. From the Edit Page, you need to either manually type in the unit tag or phase parameter, or use the String data type to define the Enumeration in the Field Code Editor.*
- *If you add an Enumeration data type field code to the field codes list, you can only use it in the instruction text field in the EWI Editor dialog box and in the EWI Logic Expression Builder launched from the Logic Step Editor dialog box.*

Editing a Field Code

▶To edit a field code:

1. In the WorkInstruction Editor, on the Field Codes menu, click Field Code Editor. The Field Code Editor dialog box appears.
2. Select the field code that you want to change.
3. Click the Modify. The Add Field Code dialog box appears.

4. Change the required fields.
5. Click OK.

Managing EIBs in the DMS

For information on managing EIBs in the DMS in the WorkInstruction Editor, refer to the following sections:

- Logging in to DMS
- DMS Functions

Logging in to DMS

▶To log in to DMS:

1. In the WorkInstruction Editor, on the DMS menu, click Document Management System. The Log On Information dialog box appears.
2. Enter a user name and password.
3. Click OK. The WorkInstruction Document Management System window opens.

   NOTE: *If you cannot connect, check that the settings in the SourceSafe Configuration dialog box in the WorkInstruction Editor are correct.*

DMS Functions

For information on DMS functions, refer to the following sections:

- Adding an EIB into the DMS
- Checking in an EIB from the DMS
- Checking out an EIB from the DMS
- Copying an EIB from the DMS
- Copying an EIB into production
- Deleting an EIB from the DMS
- Deleting an Obsolete EIB
- Displaying the Revision History for an EIB in Production
- Moving an EIB out of Production
- Undoing a Check Out of an EIB from the DMS

Adding an EIB into the DMS

▶To add an EIB into the DMS:

1. In the WorkInstruction Editor, on the DMS menu, click Document Management System. The

Log On Information dialog box appears.

2. Enter a user name and password.
3. Click OK. The WorkInstruction Document Management System dialog box appears.
4. From the Document Management System dialog box, click Add. The Open dialog box appears.
5. Select a file and click Open.
6. If the Use Signature Verification for DMS Commands option is enabled in the Database Info for Audit Trail dialog box, you will also be prompted to enter a user name and password (from the iWorkUsers Windows security group) to proceed.

Checking in an EIB from the DMS

▶To check in an EIB from the DMS:

1. In the WorkInstruction Editor, on the DMS menu, click Document Management System. The Log On Information dialog box appears.
2. Enter a user name and password.
3. Click OK. The WorkInstruction Document Management System dialog box appears.
4. From the Document Management System dialog box, select the file in the Work project.
5. Click Check In.

   NOTE: *If the Use Signature Verification for DMS Commands option is enabled in the Database Info for Audit Trail dialog box, you will also be prompted to enter a user name and password (from the iWorkUsers Windows security group) to proceed.*

Checking out an EIB from the DMS

▶To check out an EIB from the DMS:

1. In the WorkInstruction Editor, on the DMS menu, click Document Management System. The Log On Information dialog box appears.
2. Enter a user name and password.
3. Click OK. The WorkInstruction Document Management System dialog box appears.
4. From the Document Management System dialog box, select the file in the Work project.
5. Click Check Out.

   NOTE: *If the Use Signature Verification for DMS Commands option is enabled in the Database Info for Audit Trail dialog box, you will also be prompted to enter a user name and password (from the iWorkUsers Windows security group) to proceed.*

Copying an EIB into Production

▶To copy an EIB into production:

1. In the WorkInstruction Editor, on the DMS menu, click Document Management System. The Log On Information dialog box appears.

2. Enter a user name and password.
3. Click OK. The WorkInstruction Document Management System dialog box appears.
4. From the Document Management System dialog box, select the file in the Work project.
5. Click Copy To Production Project.

Copying an EIB from the DMS

To copy an EIB from the DMS:

1. In the WorkInstruction Editor, on the DMS menu, click Document Management System. The Log On Information dialog box appears.
2. Enter a user name and password.
3. Click OK. The WorkInstruction Document Management System dialog box appears.
4. From the Document Management System dialog box, select the file in the Work project.
5. Click Read.

Deleting an EIB from the DMS

To delete an EIB from the DMS:

1. In the WorkInstruction Editor, on the DMS menu, click Document Management System. The Log On Information dialog box appears.
2. Enter a user name and password.
3. Click OK. The WorkInstruction Document Management System dialog box appears.
4. From the Document Management System dialog box, select a file from the Work project.
5. Click Delete. A message box opens asking you to confirm the deletion.
6. Click Yes.

   NOTE: *If the Use Signature Verification for DMS Commands option is enabled in the Database Info for Audit Trail dialog box, you will also be prompted to enter a user name and password (from the iWorkUsers Windows security group) to proceed.*

Deleting an Obsolete EIB

To delete an obsolete EIB

1. In the WorkInstruction Editor, on the DMS menu, click Document Management System. The Log On Information dialog box appears.
2. Enter a user name and password.
3. Click OK. The WorkInstruction Document Management System dialog box appears.
4. From the Document Management System dialog box, select the file in the Obsolete project.
5. Click Delete. A message box appears asking you to confirm the deletion.
6. Click Yes.

NOTE: *If the Use Signature Verification for DMS Commands option is enabled in the Database Info for Audit Trail dialog box, you will also be prompted to enter a user name and password (from the iWorkUsers Windows security group) to proceed.*

Displaying the Revision History for an EIB in Production

▶To display the revision history for an EIB in production:

1. In the WorkInstruction Editor, on the DMS menu, click Document Management System. The Log On Information dialog box appears.
2. Enter a user name and password.
3. Click OK. The WorkInstruction Document Management System dialog box appears.
4. From the Document Management System dialog box, select the EIB in the production list.
5. Click the History button. The History dialog box appears.
6. Review the revision history in the History dialog box.
7. Click Close.

Moving an EIB out of Production

▶To move an EIB out of production:

1. In the WorkInstruction Editor, on the DMS menu, click Document Management System. The Log On Information dialog box appears.
2. Enter a user name and password.
3. Click OK. The WorkInstruction Document Management System dialog box appears.
4. From the Document Management System dialog box, select the file in the Production project.
5. Click Move to Obsolete Project.

   NOTE: *If the Use Signature Verification for DMS Commands option is enabled in the Database Info for Audit Trail dialog box, you will also be prompted to enter a user name and password (from the iWorkUsers Windows security group) to proceed.*

Undoing a Check Out of an EIB from the DMS

▶To undo a check out EIB from the DMS:

1. In the WorkInstruction Editor, on the DMS menu, click Document Management System. The Log On Information dialog box appears.
2. Enter a user name and password.
3. Click OK. The WorkInstruction Document Management System dialog box appears.
4. From the Document Management System dialog box, select the file in the Work project.
5. Click UndoCheckOut.

NOTE: *If the Use Signature Verification for DMS Commands option is enabled in the Database Info for Audit Trail dialog box, you will also be prompted to enter a user name and password (from the iWorkUsers Windows security group) to proceed.*

Using the EIB Server Manager

The following sections explain how to work with the EIB Server Manager:

- Starting the EIB Server Manager
- Selecting the Method of Booting the EIB Server
- Setting up the EWI ActiveX Control to Communicate with the EIB Server

Starting the EIB Server Manager

▶To start the EIB Server Manager:

1. On the Start menu, point to Programs, Proficy Batch Execution, and then click EIB Server Manager. The EIB Server Manager dialog box appears.
2. Click the Start Server button. The EIB Server Choose Boot Method dialog box appears.
3. Click a button to select a boot method:
    - **Warm** – to return the EIB Server to its last known state. EWIs and EIBs are returned to the state that existed prior to the termination of the server. Only use the warm boot method to recover data after an unexpected loss of the EIB Server.
    - **Cold** – to start the EIB Server with all EWIs and EIBs in the Idle state. This is the preferred method of starting the EIB Server.
    - **Cancel** – to terminate the start of the EIB Server.

Selecting the Method of Booting the Server

▶To select a boot method for the EIB Server:

Click one of the following buttons in the EIB Server Choose Boot Method dialog box:

- **Warm** – to return the EIB Server to its last known state. EWIs and EIBs are returned to the state that existed prior to the termination of the server. Only use the warm boot method to recover data after an unexpected loss of the EIB Server.
- **Cold** – to start the EIB Server with all EWIs and EIBs in the Idle state. This is the preferred method of starting the EIB Server.
- **Cancel** – to terminate the start of the EIB Server.

Setting up the EWI ActiveX Control to Communicate with the EIB Server

▶To set up the EWI ActiveX control to communicate with the EIB Server:

1. Right-click the ActiveX control and select Properties to view the ActiveX control's properties.

2. Click the VBIS Server tab.
3. On the Server tab, if the EIB Server is remote, enter the computer name on which the EIB Server is running in the Remote Machine Name field. If the EIB Server is running on the same machine as the ActiveX control, leave the field blank.

Frequently Asked Questions About WorkInstruction

The following questions are often considered regarding WorkInstruction. Before you get started with WorkInstruction, be sure to read these questions and their answers.

Is WorkInstruction 21 CFR Part 11 compliant?

WorkInstruction is designed for 21 CFR Part 11 and when implemented with proper security and database controls can produce electronic records that meet the requirements of 21 CFR Part 11. Whether or not an implementation is considered compliant is largely determined by the specific implementation and GE makes no guarantees or warranties in this regard.

When are events recorded?

Events are recorded by the WorkInstruction EIB Server to the electronic batch record, on completion of an EWI Step, Logic Step, or EIB. Before events are recorded, they are queued in the WorkInstruction EIB Server. An EWI Step, Logic Step, or EIB is completed when you fulfill all signature requirements. If no Performed By or Verified By signature is required, the Windows computer name or login name is captured by default. Based on how you configure your EWI Step, Logic Step, or EIB, WorkInstruction captures either the Windows computer name or login name.

Why can't you use Microsoft Access with WorkInstruction?

A file-based database management system, such as Microsoft Access, is significantly slower than standard SQL-based database management systems, such as Microsoft SQL Server and Oracle. Additionally, with WorkInstruction, some of the field sizes are very large (greater than 1024 bytes) and Microsoft Access does not support this large a field size.

What part of the user name is captured for the signature?

WorkInstruction captures the Windows full user name of the operator who is executing or signs the EWI or Logic Step or completed EIB. For example, if a user account existed with a user name of JBangs, and a full user name of Joseph Bangs, WorkInstruction captures Joseph Bangs and not JBangs.

NOTE: *The full user name must be configured in Windows security, otherwise WorkInstruction does not accept the signature.*

What if I design my EWI or Logic Step not to capture a signature? Is the login name or computer name recorded?

Yes, by default, if there are no Preformed by or Verified By signatures required, the Windows login name or computer name is recorded. Based on how you configure your EWI Step, Logic Step, or EIB in the WorkInstruction Editor, either the Windows computer name or login name is captured.

Index