Wonderware® FactorySuite™ InTrack

Runtime Developer's Guide

Revision E
June 2001

Wonderware Corporation

100 Technology Drive
Irvine, CA 92618
U.S.A.
(949) 727-3200
http://www.wonderware.com

Trademarks

.

Contents

CHAPTER 1

Introduction

This user's guide contains information for developing your Wonderware FactorySuite InTrack runtime graphical user interface (GUI). The approach you use when developing an InTrack application determines how transactions against the relational database created by Wonderware FactorySuite InTrack ModelMaker will be performed in runtime. Processing a transaction can be performed manually by the operator using ActiveX controls, or automatically by the InTrack system by creating InTouch QuickScripts which use InTrack OLE automated scripts created by the application developer.

This chapter describes the object types supported by InTrack and how transactions can be processed against your database. It is assumed that you have already created and configured your database using ModelMaker.

For more information on how to create and configure the your InTrack database, see your *InTrack Getting Started User'sGuide.*

For more information on using InTouch, see your *InTouch Runtime User'sGuide.*

Contents

Objects

The concept of objects applies to both the development and runtime environments of an InTrack application. However, each of these two environments uses a separate type of object. Structural objects are the static objects created at development using the FactorySuite ModelMaker. Activity objects are the dynamic objects created, tracked, or modified via transactions at runtime.

Structural Objects

Structural objects are objects created during development in ModelMaker. Structural objects are, for the most part, static components of the manufacturing model. When used together, structural objects provide the "structure," or rules, that support the flow of WIP lots, inventory lots, bulk inventory, machines and their related data in a manufacturing enterprise. The following is a list of all structural objects:

- Calendars
- Customer specifications
- Data set templates
- Disposition codes
- Failure reasons
- Failure symptoms
- Locations
- Machines
- Machine tasks
- Machine types
- Materials
- Operations
- Privileges
- Routes
- Security groups
- Setpoint templates
- Tables
- Users
- User certifications
- Work instructions

For more information on structural objects, see your *InTrack ModelMaker User's Guide*.

Activity Objects

Activity objects are objects created and manipulated at runtime through transactions or OLE scripts. Transactions can be issued either manually by an operator or automatically by the InTrack system. Activity objects are dynamic components of the manufacturing model in that they can move between the different operations of a manufacturing route. The following is a list of all activity objects:

- Work In Process (WIP) lots
- Inventory lots
- Sublots
- Data set samples

When you create a WIP lot, transfer an inventory lot, ship a bulk inventory, or open a machine repair, you are manipulating an activity object. For example, when a quantity of the material Bottles, a structural object, is received into inventory and assigned a lot ID, it becomes an activity object. The inventory of bottles is now tracked by the system as a unit, identified by a lot ID.

The highlighted row of bottles in the Inventory Lot Selector, Bottle-0402-001, represents an activity object:

Inventory Lot Selector

		Lot ID	Material	Material Ver	Create Date	Quantity	Units	Location	Customer Spec	Customer Ver	Hold Status
1		Bottle-0402-001	Bottles	NONE	06/04/96	999,354.0000		Storeroom	NONE	NONE	RELEASED
2		Bottle-0402-002	Bottles	NONE	06/04/96	345.0000		Storeroom	NONE	NONE	RELEASED
3		Bottle-0415-001	Bottles	NONE	06/04/96	250.0000		Storeroom	NONE	NONE	RELEASED
4		Caps-0401-001	BottleCaps	NONE	06/04/96	74,360.0000		Storeroom1	NONE	NONE	RELEASED
5		Caps-0402-002	BottleCaps	NONE	06/04/96	350.0000		Storeroom1	NONE	NONE	RELEASED
6		Caps-0403-003	BottleCaps	NONE	06/04/96	4,000.0000		Storeroom1	NONE	NONE	RELEASED
7		CW-0411-001	WonderWater	1	06/04/96	4,000.0000		Storeroom2	1	1	RELEASED
8		CW-0411-002	WonderWater	1	06/04/96	4,000.0000		NONE	1	1	RELEASED
9		CW-0411-003	WonderWater	1	06/04/96	1,000.0000		NONE	1	1	RELEASED
10		CW-0411-004	WonderWater	1	06/04/96	1,000.0000		NONE	1	1	RELEASED

Operators can manipulate an activity object at runtime by selecting the activity object in its appropriate selector and creating a dialog box (e.g., ship, move, adjust, etc.) through an ActiveX control or OLE script.

Transactions

A transaction is a collection of one or more OLE script statements that read and/or write information to the relational database. All transactions are processed at runtime and are performed as a single unit of work for an activity object. InTrack uses the InTouch script editors and runtime engine to implement and process InTrack transactions. In order for the system to process a transaction for an activity object, the activity object either must be selected in the appropriate selector or referenced in the syntax of the OLE script.

Manual Transactions - Selectors and Buttons

Selectors are the runtime display objects for the InTrack database tables. Operators can use these selectors to view the current status of WIP lots, inventory lots, bulk inventory, or machines and to select items to work on:

WIP Selector

		Lot ID	Operation	Operation Ver	Work State	Priority	Due Date	Material	Material Ver	Queued Qty	Started Qty
1		CWRA-0411-001	Label	NONE	QUEUED	0	12/21/96	RaspberryFlav	1	15.0000	0.0000
2		CW-0418-002	Verify	NONE	QUEUED	0	12/21/96	OrangeFlavor	2	25.0000	0.0000
3		CWNF-0411-001	Label	NONE	IN_PROCESS	0	12/21/96	WonderWate	1	0.0000	25.0000
4		CWNF-0411-002	Package	NONE	IN_PROCESS	0	12/21/96	WonderWate	1	0.0000	15.0000
5		CWNF-0411-003	Cap	NONE	QUEUED	0	12/21/96	WonderWate	1	2.0000	0.0000
6		CWOR--0418-001	Fill	NONE	IN_PROCESS	0	12/21/96	OrangeFlavor	2	0.0000	1.0000
7		CW-0418-001	Verify	NONE	QUEUED	0	12/21/96	OrangeFlavor	2	20.0000	0.0000
8		CWRA-0411-002	Fill	NONE	QUEUED	0	12/21/96	RaspberryFla	2	45.0000	0.0000
9		CWRA-0411-003	Fill	NONE	QUEUED	0	12/21/96	RaspberryFla	2	50.0000	0.0000
10		CWRA-0411-004	Fill	NONE	IN_PROCESS	0	12/21/96	RaspberryFla	1	10.0000	10.0000
11		PCNF-0411	Label	NONE	QUEUED	0	12/21/96	WonderWate	1	15.0000	0.0000
12		PCNF-0411-001	Label	NONE	IN_PROCESS	0	12/21/96	WonderWate	1	0.0000	50.0000
13		PCOR-0411-002	Package	NONE	IN_PROCESS	0	12/21/96	OrangeFlavor	2	0.0000	15.0000

Each selector has an ActiveX control that supports it. An ActiveX control, when invoked, is processed on the item currently selected in the applicable selector.

When processing manually, the operator must select an activity object from the appropriate selector, then execute the OLE script for the selected object, usually through an ActiveX control. The OLE script executed by the ActiveX control does not specify a particular activity object as part of its syntax; the script function processes for the currently selected item in the selector.

Automated Transactions - OLE Scripts

An automated transaction processes one or more InTrack-specific OLE scripts against the database based on an event, action, or condition. A transaction can be processed based on changing of a PLC, an operator invoking an event, a particular condition in the system, and so on.

Note: It is recommended that you create InTouch QuickScripts that use the automated OLE transactions. Automated transactions through command scripts are supported in InTrack.

For more information on OLE scripts, see your *InTrack OLE Reference User's Guide*.

The following table identifies the five types of scripts supported by the InTrack system and the triggering mechanisms for each:

Script Name	Types	Triggering Mechanism
ActiveX Event Scripts	Based on ActiveX Control	Based on ActiveX Control
QuickFunction Scripts	Based on Script	Based on Script
Application Script	On Startup	Runs once during application startup
	While Running	Based on time frequency
	On Shutdown	Runs once during application shutdown
Window Script	On Show	Runs once when window is opened
	While Showing	Based on time frequency
	On Hide	Runs once when a window is closed
Key Script	On Key Down	Runs once when key is pressed
	While Down	Based on time frequency
	On Key Up	Runs once when key is released
Condition Script	On True	Runs once when condition turns true
	While True	Based on time frequency
	On False	Runs once when condition turns false
	While False	Based on time frequency
Data Change Script	Changing Value	Runs when a tag's value changes

For more information on QuickFunction Scripts, see your *InTouch User'sGuide.*

CHAPTER 2

Working with InTrack ActiveX Controls

ActiveX controls, originally known as OLE controls or OCXs, are standalone software components that perform specific functions in a standard way. They define standard interfaces for reusable components. ActiveX controls are not separate applications. Instead, they are servers that are placed into a control container. To use ActiveX controls, they must be placed in an ActiveX container. InTouch is the ActiveX container for FactorySuite. InTouch WindowMaker is the development environment used to configure your ActiveX controls.

InTrack ActiveX controls are designed to be used in the development of a runtime client application for InTrack. They provide information needed to execute InTrack database transactions. For example, to create a WIP lot, a Material and a Route must be specified. The ITQueryList is used to present a list of InTrack Materials to the user. Other controls are used to support other InTrack transactions. Each query control is based on a query provided through the InTrack Automation Server. All InTrack ActiveX controls provide an interface to InTrack OLE object classes. Therefore, the properties and methods of the OLE object class are provided and selected by the ActiveX control during development.

For more information on OLE References, see your *InTrack OLE Reference*.

Contents

Working with ActiveX Controls

There are three main components of ActiveX controls: *properties*, *methods* and *events*. Properties are very similar to variables that you can modify, for example, calendar day, control height, and so on. Methods are similar to script function calls that you can call from the container.

For example, **Browser.Navigate("http://www.wonderware.com")**, **Engine.start()**.

Events occur through the ActiveX container. For example, **Control.click (shift)**. **FileViewer.DoubleClick (name)**, and so on.

InTouch allows you to access ActiveX control properties, methods and events. You can associate these properties with InTouch tagnames or you can access them through InTouch QuickScripting.

Note: In order for an ActiveX Event script to function properly the ActiveX control for which the script was created, must be loaded into memory. If the window containing an ActiveX control is closed, its ActiveX Event scripts, or any other InTouch QuickScripts containing script functions associated with that ActiveX control, will not execute properly.

You can use one or more ActiveX controls in your InTouch application. InTouch allows you to easily select and paste an ActiveX control into any application window and to add them to your **Wizards/ActiveX Toolbar**. You can also import ActiveX Event scripts from one application to another.

➢ To use an ActiveX control in InTouch:

1. Install the ActiveX control(s) you want to use.
2. Select and paste the ActiveX control into a WindowMaker window.
3. Configure the ActiveX control's properties and assign them to tagnames.
4. Associate ActiveX events to ActiveX Event scripts.
5. Call ActiveX methods and set ActiveX control properties in ActiveX Event scripts, or other InTouch QuickScripts.

➢ To edit an ActiveX control in InTouch:

The following WindowMaker edits can be made to an ActiveX control:

- An ActiveX control's size can be changed, if sizing is supported by the control.
- ActiveX controls can be duplicated, cut, copied, pasted and deleted.
- All aligning commands (left, right, top, bottom, centerpoint) can be applied to an ActiveX control.
- ActiveX controls can be added to the **Wizards/ActiveX Toolbar**.
- ActiveX controls can be included with other objects when creating a cell.
- The WindowMaker menu commands and their equivalent toolbar tools can be used to directly modify many ActiveX properties. For example, Reduce Font, Line Color, Fill Color, and so on.

InTouch does not support the following types of ActiveX controls:

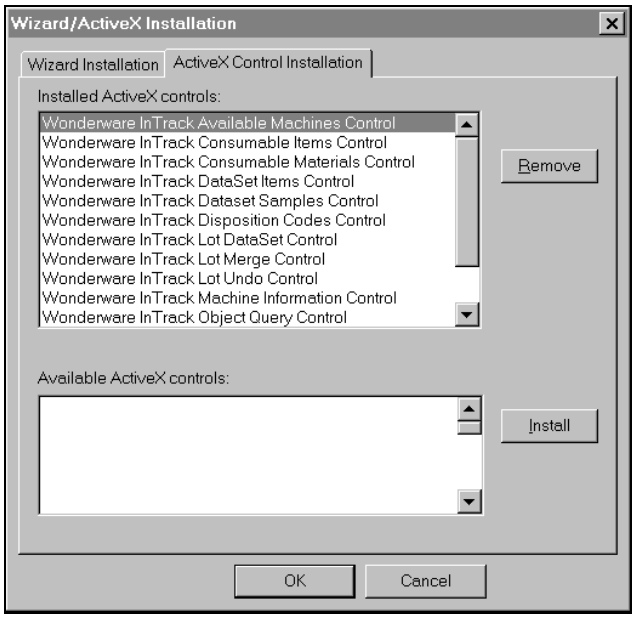
- Windowless Controls
- Simple Frame Site (Group Box)
- Containers
- Data Controls
- Dispatch Objects
- Arrays, Blobs, Objects, Variant Types

➢ To install or remove an ActiveX control:

1. On the **Special** menu, point to **Configure**, then click **Wizard/ActiveX Installation**, or in the Application Explorer, double-click **Wizard/ActiveX Installation**. The **Wizard/ActiveX Installation** dialog box appears.

   In the Application Explorer, you can also right-click **Wizard/ActiveX Installation**, then click **Open**.

2. Click the **ActiveX Control Installation** tab to activate the **ActiveX Installation** property sheet:



3. In the **Installed ActiveX controls** list, select the control(s) that you want to remove from your application, then click **Remove**. An interactive message box will appear asking you to confirm the deletion.

   To select a group of controls, click your first selection, hold down the SHIFT key and select your last selection. All controls in between will be selected as well. To select multiple controls that are not consecutively listed, click the first control, then hold down the CTRL key as you click another.

4. Click **Yes** to remove the control(s). The removed control(s) is moved to the **Available ActiveX controls** list.

   When you remove a control, it is not deleted. However it is no longer loaded into memory. Therefore, it will not function properly.

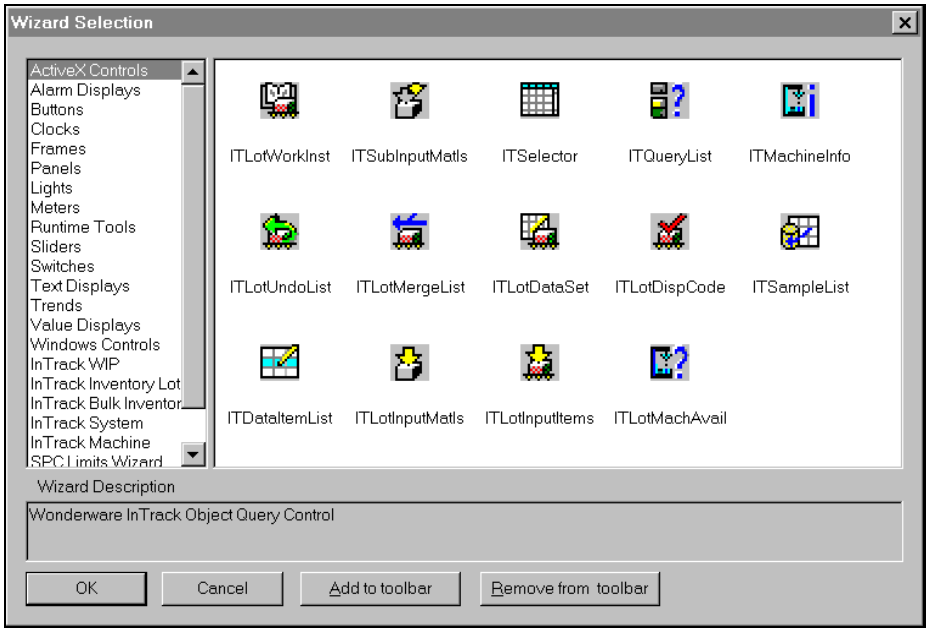
5. To install ActiveX controls, select them in the **Available ActiveX controls** list, then click **Install**.

   The **Install** button is active only when controls are displayed in **Available ActiveX controls** list.

6. Click **Close**.

➢ **To place an ActiveX control in a window:**

1. Click the Wizard Dialog tool in the **Wizards/ActiveX Toolbar**. The **Wizard Selection** dialog box appears:



2. In the list of wizards, click the **ActiveX Controls** category. All available ActiveX controls will be shown the display area.

3. Select the ActiveX control that you want to use, then click **OK**, or double-click the control. The dialog box will close and your window will reappear.

   To add the ActiveX control to the **Wizards/ActiveX Toolbar**, click **Add to toolbar**. After you add a control to the **Wizards/ActiveX Toolbar**, you can select it and paste it into your open window at any time.

   Note The number of ActiveX controls that you can add to the toolbar is limited to your system resources.

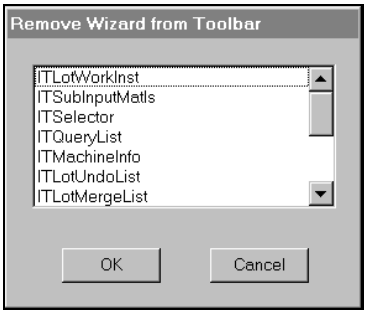
4. The cursor will change to the corner symbol, , when you return to the window. Click the location in the window where you want to paste the ActiveX control.

5. Double-click the control to configure it properties.

For more information on the WindowMaker toolbars, see your *InTouch User's Guide*.

➢  **To remove ActiveX controls from the toolbar:**

1. Click the Wizard Dialog tool in the **Wizards/ActiveX Toolbar**. The **Wizard Selection** dialog box will appear.
2. Click **Remove from toolbar**. The **Remove Wizard from Toolbar** dialog box appears:



3. Select the ActiveX control(s) that you want to remove from the toolbar.
4. Click **OK**.

Configuring an ActiveX Control

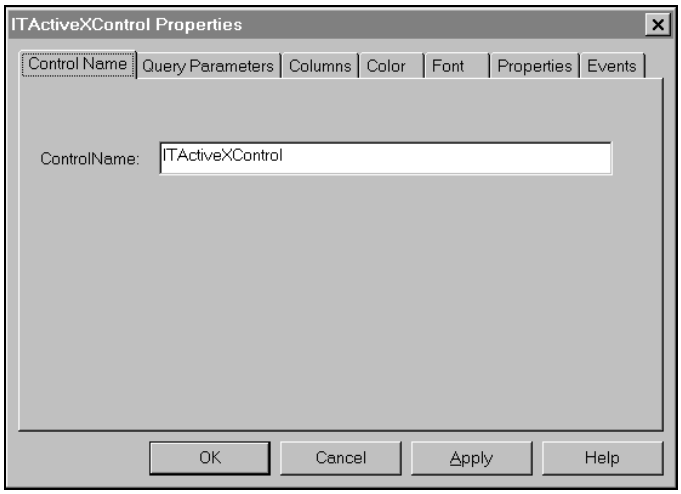
When you paste an ActiveX control into an InTouch window, you must configure its properties to interact with InTouch. Each control must be named for reference from InTouch QuickScripts. A default control name such as, ITActiveXControl, will be generated when you paste the ActiveX control. (This control name will be global within your InTouch application.)

The ActiveX control's properties must be assigned to InTouch tagnames. Each property type must be assigned to an equivalent InTouch tagname type.

➢ To name an ActiveX control:

1. Paste the ActiveX control into your WindowMaker window.
2. Double-click the control, or right-click the control, then click **Properties**. The control's respective **Properties** dialog box will appear.

Note Each ActiveX control's **Properties** dialog box is unique to the control. The number of tabs displayed is based upon the properties of the particular control. Some ActiveX controls may require you to configure more properties than others do. For example some controls may require you to configure their **Colors** and **Fonts**, while others may not have these properties. However, for all ActiveX controls, InTouch adds three tabs; **Control Name, Properties** and **Events**.



3. Click the **Control Name** tab, then type a unique name for the ActiveX control in the **ControlName** box.

   You must define a unique name for each ActiveX control used in your InTouch application. The Control Name is used in script functions to identify the control. For example:

   ```
   #ITActiveXControl.day = Tag1;
   #ITActiveXControl.year = 1997;
   ```

Note If you use the default Control Name, for example, ITActiveXControl1, and you subsequently duplicate the ActiveX control, InTouch will automatically increment the Control Name. In this case, the duplicate ActiveX control's name would be ITActiveXControl2.

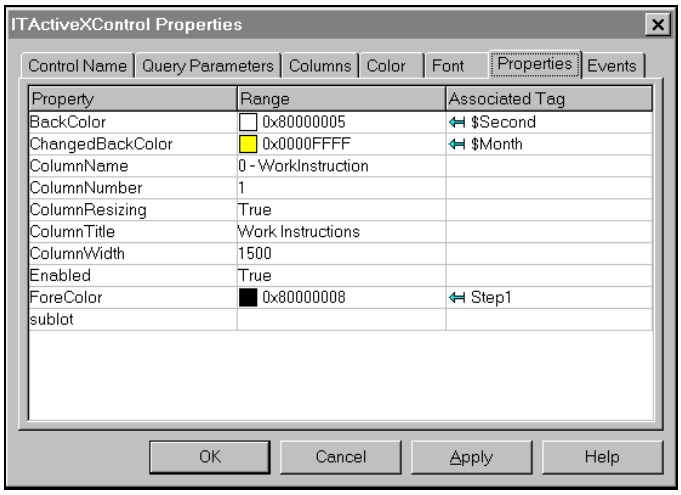
Configuring ActiveX Control Properties

The properties that you can configure for a particular ActiveX control are determined by the ActiveX control designer. Each ActiveX control's **Properties** property sheet displays three columns: **Property**, **Range** and **Associated Tag**. The **Property** and **Range** columns are read-only. The **Associated Tag** column is used to associate InTouch tagnames with the respective property in the **Property** column.

Note When you click certain items in the **Range** column an arrow will appear that you can click to view the list of possible values for the item. The items in the list for viewing purposes only and cannot be changed.

➢ **To configure an ActiveX control's properties:**

1. Click the **Properties** tab in the ActiveX control's **Properties** dialog box to activate the **Properties** property sheet:



2. Click in the middle of each cell in the **Associated Tag** column, then type a tagname for the respective property.

   If you type in a tag name that is not defined in the Tagname Dictionary, you will be prompted to define it now.

   If you double-click a blank cell, or click the  button, the Tag Browser will appear displaying the tagnames for the selected tag source. Double-click the tagname that you want to use, or select it, then click **OK**. The tagname is automatically inserted into the cell.

For more information on the Tag Browser, see your *InTouch User's Guide*.

3. After you specify the tagname, double-click in the cell to the left of the tagname to select the association direction for the tagname to its respective property. (Continuously double-clicking will cycle you through the various association direction choices. The association direction choices are described below.)

   There are two fields in each cell in the **Associated Tag** column. The association direction selection and the tagname entry. The ActiveX control determines the association direction and the property type determines the tagname type that must be used.

   You can select one directional or bi-directional association. However, if the association direction you select is not valid for the property or tagname, the control will automatically change it accordingly. For example, if you select ⇦, when the tagname's value changes, its associated property is changed accordingly. Select the appropriate association symbol as follows:

   - ⇦ The tagname sets the value of the associated property.
   - ⊢ This symbol indicates that the property is read-only and the tagname cannot change the property's value.
   - ⇨ The property sets the value of the associated tagname.
   - ⊣ This symbol indicates that the tagname is read-only and the property cannot change the tagname's value.
   - ⇔ Value can be set from both the tagname or the property. (Tagname takes precedence.)
   - ⊢⊣ The tagname and the property are both read-only.
   - ⇦⊣ The tagname can change the property's value, but the property cannot change the tagname's value. The property cannot change the tagname's value because the property is non-bindable, or the tagname is read-only.
   - ⊢⇨ The property can change the tagname's value, but the tagname cannot change the property's value. The tagname cannot change the property's value because the property is read-only.

4. Click **OK**.

Note: You can also access or change properties through ActiveX Event scripts and/or other InTouch QuickScripts. All ActiveX script functions are qualified by the # (pound) sign. The valid syntax to access ActiveX properties is:

```
#ControlName.PropertyName
```

Examples:

```
#ITActiveXControl.Day = 29;

Tag1 = #ITActiveXControl.year;
```

For more information, see your *InTouch User's Guide*.

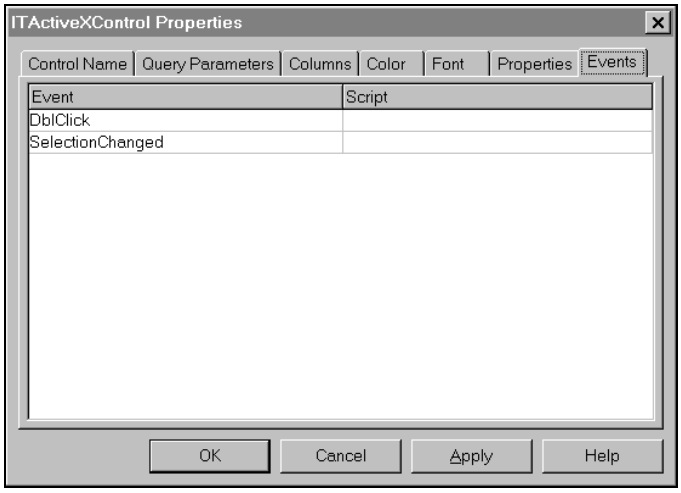
Using ActiveX Control Methods

ActiveX control methods are similar to ActiveX control properties. You can activate methods in runtime (WindowViewer). ActiveX control methods are accessed through ActiveX Event scripts and/or other InTouch QuickScripts.

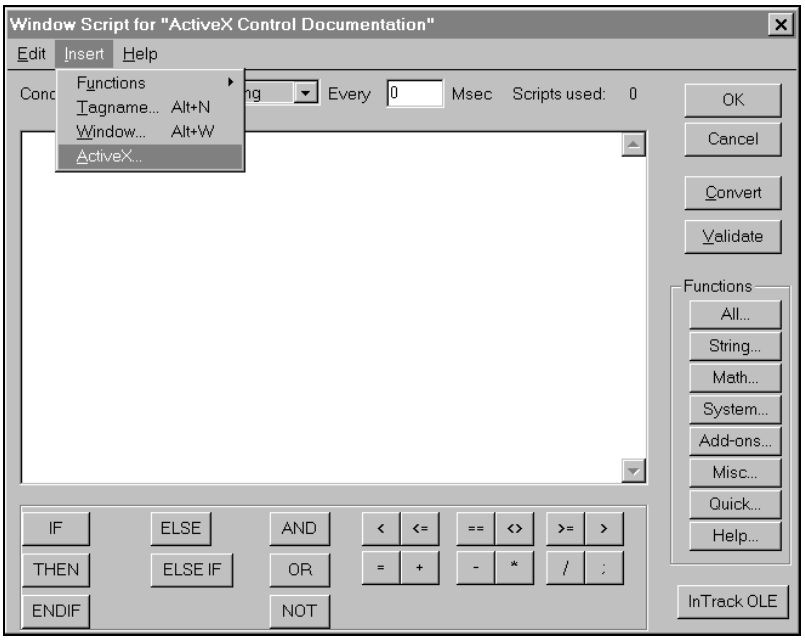
Note: In order for an ActiveX Event script to function properly the ActiveX control for which the script was created, must be loaded into memory. If the window containing an ActiveX control is closed, its ActiveX Event scripts, or any other InTouch QuickScripts containing script functions associated with that ActiveX control, will not execute properly.

➢ **To use ActiveX methods and/or properties:**

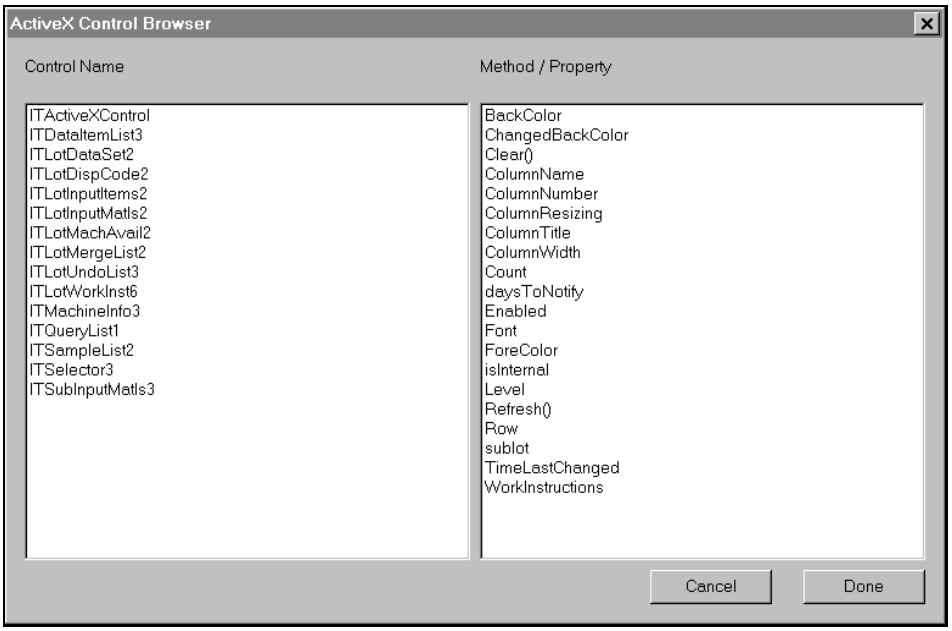
1. In the ActiveX control's **Properties** dialog box, click the **Events** tab to activate the **Events** property sheet:



2. Double-click a blank cell in the **Script** column. The **ActiveX Event Scripts** editor appears:



3. On the **Insert** menu, click **ActiveX**. The **ActiveX Control Browser** appears:



4. In the **Control Name** list, select the ActiveX control whose methods or properties you want to access.

   The names of all ActiveX controls currently being used in your application will be listed.

Note If you select **This Control** instead of the actual **Control Name**, the methods and properties displayed will be those for the ActiveX control currently selected in your application. By selecting **This Control** instead of the actual **Control Name**, you can create generic ActiveX Event script functions. You can then copy and paste these functions into any other ActiveX Event script, or any other InTouch QuickScript without having to change the **Control Name** in the new script. For example:

```
#ThisControl.Navigate  (“http:\\www.wonderware.com”);

#ThisControl.Navigate(URL);   { where URL is a tagname}
```

This Control is accessible only through ActiveX Event scripts. It is not accessible through any other type of InTouch QuickScript.

---

5. In the **Method / Property** list, select the method or property that you want to use in your script.

   Properties are the items in the list that include parenthesis. For example, **Display()**.

6. Click **Done**. The selected control name and method or property are automatically inserted into your script.

   ActiveX control's methods and properties are also accessed through the **Insert** menu in all other InTouch QuickScripts types.

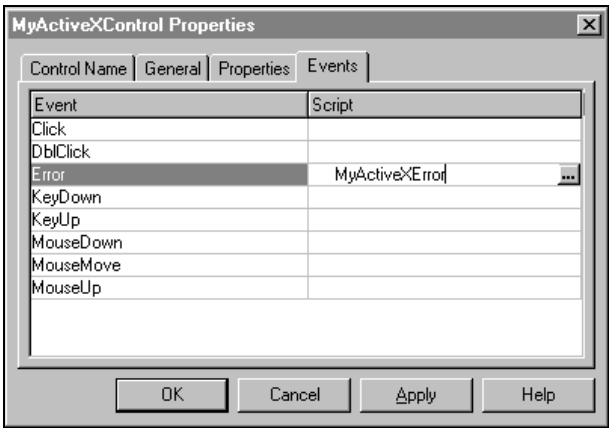
Using ActiveX Control Event Parameters

You can execute ActiveX control events in runtime (WindowViewer) by designing a particular action and associating it to the event. For example, if your ActiveX control has an error event handler, you could create a ActiveX Event script that displays and a window with an error message when an error occurs or any other InTouch QuickScript. ActiveX Event Scripts are provided to support event actions. You can associate a named event script to each event.

Note In order for an ActiveX Event script to function properly the ActiveX control for which the script was created, must be loaded into memory. If the window containing an ActiveX control is closed, its ActiveX Event scripts, or any other InTouch QuickScripts containing script functions associated with that ActiveX control, will not execute properly.

➢ **To use ActiveX event parameters:**

1. Double-click the ActiveX control for which you want to create an ActiveX Event script. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the **Events** tab to activate the **Events** property sheet:

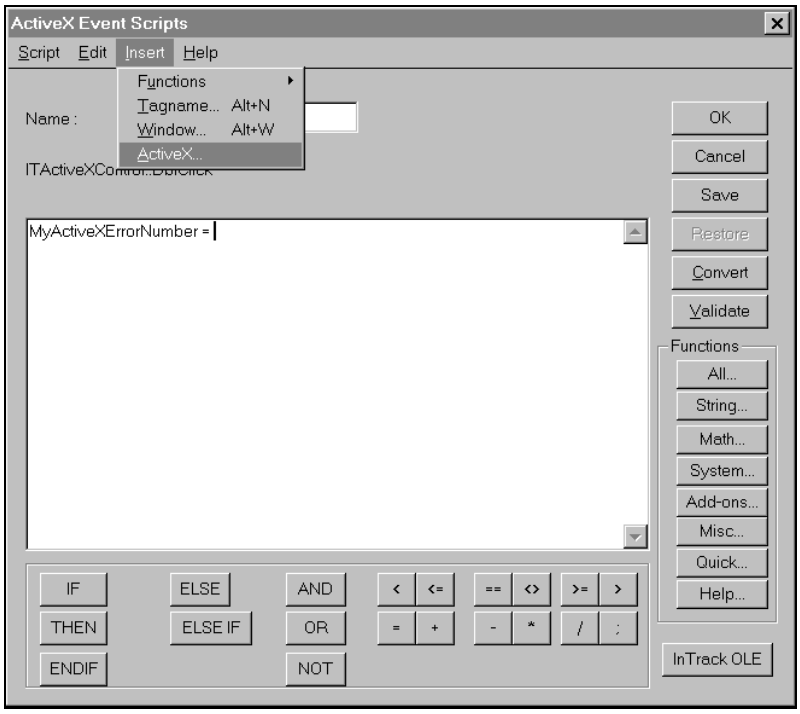


3. In the **Event** column select the event to which you want to associate an ActiveX Event Script.
4. In the respective cell in the **Script** column, type a unique name for the ActiveX Event Script that you want to create, then double-click the name, or click **OK**. The following message box appears:

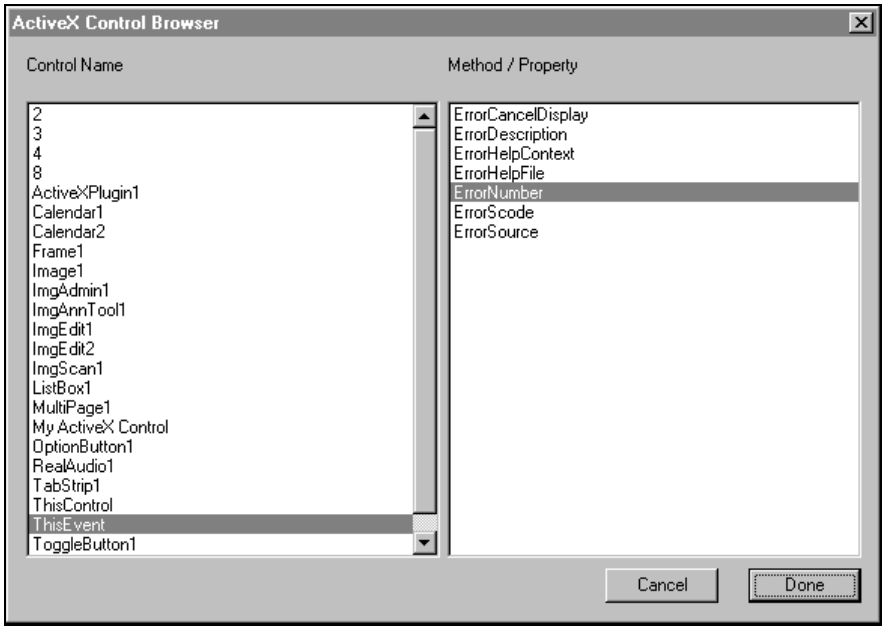
Click **OK**. The ActiveX Event script editor will appear displaying the name that you typed in the **Name** input box (see example below). If you double-click a blank **Script** cell, when the ActiveX Event script editor appears, you must then type a name for the ActiveX Event script.

If the ActiveX Event script that you want to use already exists, click the ... button. The Choose ActiveX Script dialog box will appear listing all existing ActiveX Event scripts in your application.

For more information, see your *InTouch User's Guide*.



5. On the **Insert** menu, click **ActiveX**. The **ActiveX Control Browser** appears:



6. In the **Control Name** list, select **This Event** to access the parameters for the selected event. In this case, the selected event is, **Error**.

Note **This Event** is accessible only through ActiveX Event scripts. It is not accessible through any other type of InTouch QuickScript. You must select **This Event** to access the event parameters for an ActiveX control.

Events may or may not pass parameters in runtime. Event parameters can be accessed by using the **ThisEvent** keyword. For example:

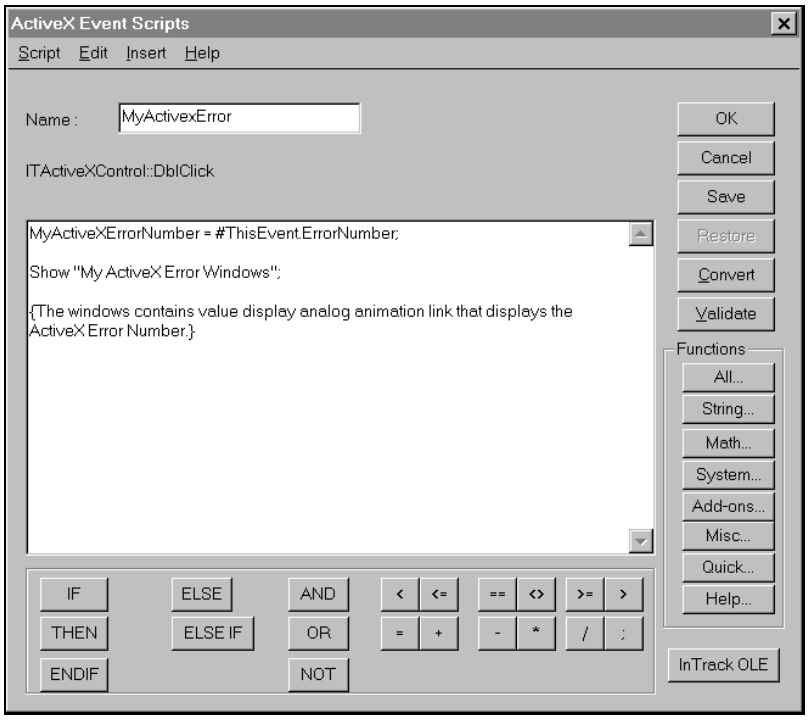
```
MyActiveXErrorNumber = #ThisEvent.ErrorNumber;
```

Where: **#** indicates that this is an ActiveX script function. **ThisEvent** relates to the event selected in the ActiveX control's **Event** property sheet, and **ErrorNumber** is the parameter passed by the selected event.

7. In the **Method / Property** list, select the event that you want to use in your ActiveX Event script.

8. Click **Done**.

   The selected control name, in this case, **This Event**, and selected event parameter are both automatically inserted into your script at the cursor location. For example:



9. Click **OK** to save your ActiveX Event script and close the script editor. The ActiveX control's **Properties** dialog box reappears.

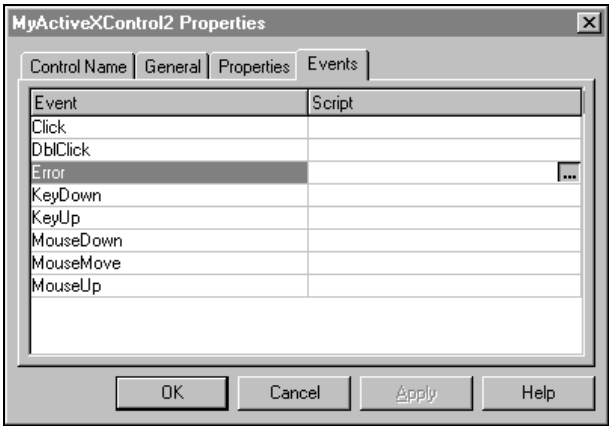
10. Click **OK** to close the **Properties** dialog box, or continue to create ActiveX Events scripts.

Reusing ActiveX Event Scripts

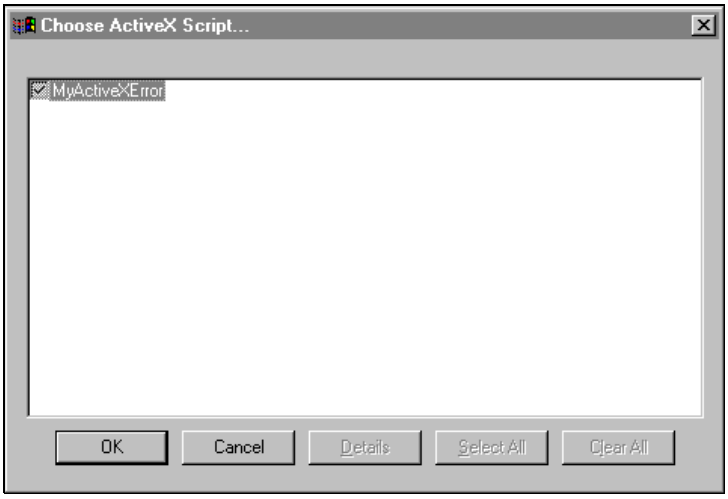
ActiveX Event scripts can only be reused for the same event for the same kind of ActiveX control. For example, the mouse down event may be a stock event on hundreds of ActiveX controls. However, an ActiveX Event script written for mouse down on ActiveX ControlA cannot be reused for mouse down on ActiveX ControlB unless the two controls are the same type.

➢ **To reuse an ActiveX Event script:**

1. Double-click the ActiveX control for which you want to reuse an existing ActiveX Event script. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the **Events** tab to activate the **Events** property sheet:



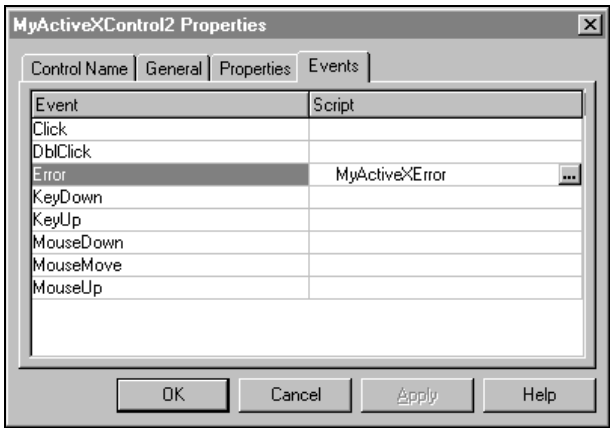
3. In the **Script** column for the respective event, click the ... button. The **Choose ActiveX Script** dialog box appears:



Note This dialog box will only display the ActiveX Event scripts that were written for the same type of ActiveX control and the same selected event.
For example, let's assume that you are creating an ActiveX Event script for a second ITActiveXControl control's "Click" event. You have already created two other ActiveX Event scripts named Click1 and Click2 in your application. Click1 was created for a different ITActiveXControl control's "Click" event, and Click2 was created for an ActiveX InSQLTrend control's "Click" event. When you click the ... button and the **Choose ActiveX Script** dialog box appears, it will only display the Click1 script since was created for the same type of ActiveX control and the same event.

4. Select the ActiveX Event script that you want to use, then click **OK**.

   The name of the selected script is automatically inserted into the **Script** cell where you previously clicked the ... button. For example:



5. Click **OK** to close the **Properties** dialog box, or continue to create ActiveX Events scripts.

Importing ActiveX Event Scripts

Importing ActiveX Event scripts from one InTouch application to your current application, can save you a considerable amount of development time. When you move ActiveX Event scripts from one InTouch application to another, you must use the **Import** command on the WindowMaker **File** menu.

Note When you import ActiveX Event scripts, from one application to another, all ActiveX Events scripts are imported. Additionally, in order for an imported ActiveX Event script to function properly in the new application, the same ActiveX control and the same event for which the script was originally created, must also be used in the new application, and it must be loaded into memory. If the window containing an ActiveX control is closed, its ActiveX Event scripts, or any other InTouch QuickScripts containing script functions associated with that ActiveX control, will not execute properly.

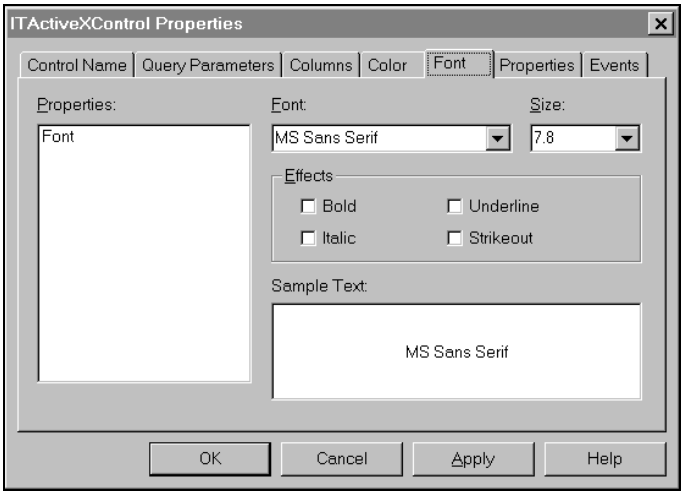
For more information on importing scripts, see your *InTouch User's Guide.*

Common Font Properties of InTrack ActiveX Controls

The InTrack ActiveX controls have properties that allow you to change the appearance of the control. Each InTrack ActiveX control has provisions to change the font attributes for the runtime display. Each InTrack ActiveX control can have a different font.

➢ **To view or change the font attributes of an InTrack ActiveX control:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the **Font** tab to activate the Font property sheet:



When selected, the **Font** dialog will display a list of available fonts on your Windows system. A TrueType logo next to a font name indicates that it is a TrueType font. Click on the desired font to select it.

Note It is possible to select a font at design time that may not be available in runtime. Windows will attempt to provide a substitute font that may produce an undesirable display.

The font size (in points) may be selected with the **Size** dialog. A list will display available font sizes, or a font size may be manually entered.

Any desired effects (bold, italic, underline or strikeout) may be selected in the **Effects** window.

The selected font, size and effects are displayed in the **Sample Text** window.

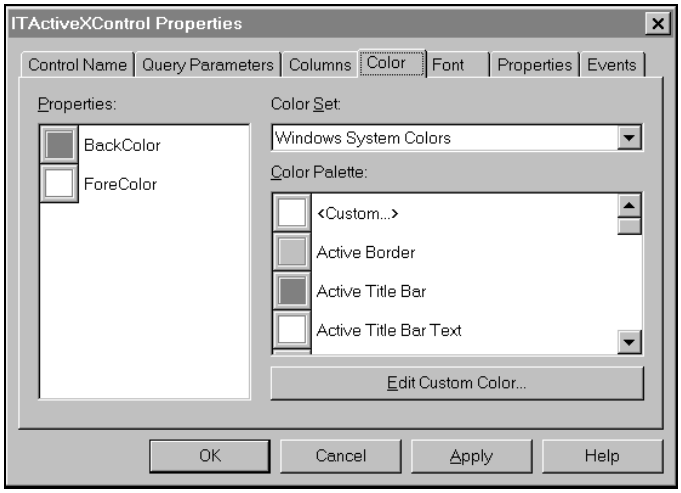
After you have assigned the desired InTrack ActiveX control font attributes, click **OK** to close the **Properties** dialog or **Apply** to continue working with other property tabs.

Common Color Properties of InTrack ActiveX Controls

The InTrack ActiveX controls have properties that allow you to change the appearance of the control. Each InTrack ActiveX control has provisions to change to the **BackColor** (The background color) and the **ForeColor** (the foreground, or text, color) attributes. Additional color attributes that can be changed are listed in the description of that InTrack ActiveX control. Each InTrack ActiveX control can have a different color scheme.

➢ **To view or change the color attributes of an InTrack ActiveX control:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the **Color** tab to activate the Color property sheet:



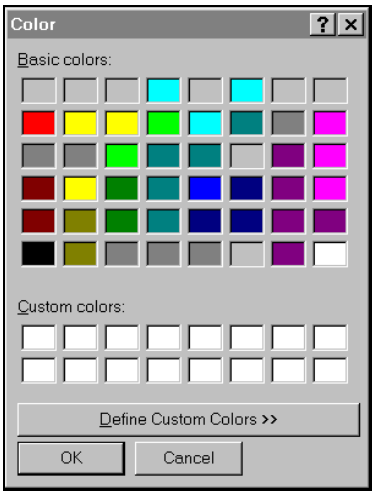
There are two **Color Sets** available for use with InTrack ActiveX controls:

- **Standard Colors** - The standard Windows 16 color palette, plus one definable custom color. The custom color may be changed for each attribute. For example, a custom color may be defined for BackColor, and another for ForeColor.
- **Windows System Colors** - This palette will allow you to assign the InTrack ActiveX control colors to the current Windows system colors. For example, the BackColor may be assigned to the Windows Active Title Bar color and the ForeColor may be assigned to the Windows Active Border color. When the Windows system colors are changed through Control Panel, the InTrack ActiveX control colors will change to match. One definable custom color is available as well. The custom color may be changed for each attribute. For example, a custom color may be defined for BackColor, and another custom color may be defined for ForeColor.

Note It is possible to assign the InTrack ActiveX control color attributes to Windows system colors that could result in the BackColor and ForeColor (and any additional attributes, if available) being the same, thereby rendering the InTrack ActiveX control unreadable.

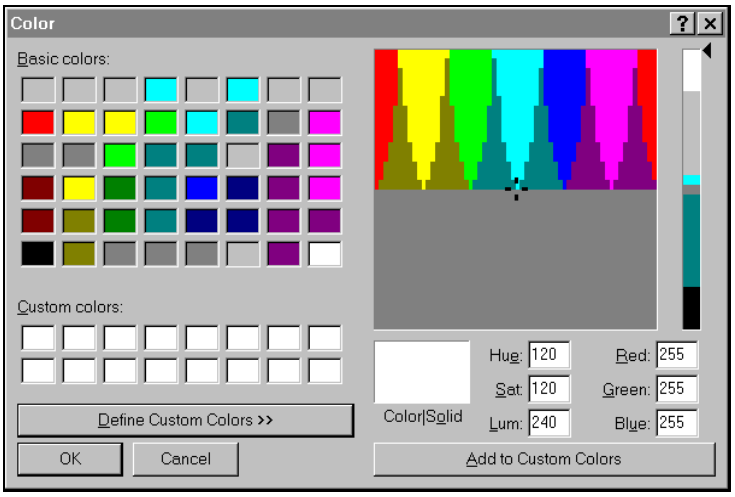
3. To select a color set, click on the **Color Set** dialog, then select the desired color set.

4. To select a custom color, click the **Edit Custom Color** button to open the **Color** dialog box:



The **Color** dialog box displays the 48 basic colors as well as 16 user definable custom colors. Click on the desired color and click **OK**.

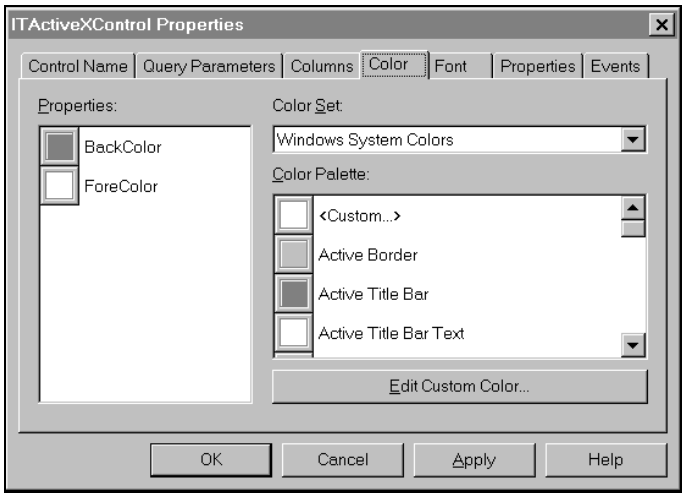
5. To define a custom color, click the **Define Custom Colors** button and the **Color** dialog box will expand:



The custom color is defined by placing the target cursor on the desired color of the palette. The attributes of the custom color (hue, saturation and luminosity) are selected with the slider at the right of the palette. When the custom color has been defined, click the **Add to Custom Colors** button. The defined color is then added to the **Custom colors** display and may be selected in the same manner as the basic colors.

Another way to define a custom color is to manually enter the hue, saturation, luminosity, red, green and blue values. The **Color|Solid** window will display the defined color.

6. To assign a color to an InTrack ActiveX attribute, select the attribute in the **Properties** window on the **Color** property sheet and the desired color in the **Color Palette** window:



7. After you have assigned the desired InTrack ActiveX control color attributes, click **OK** to close the **Properties** dialog or **Apply** to continue working with other property sheets.

InTrack Grid-Based ActiveX Controls

All InTrack grid-based ActiveX controls share the same **Columns** tab in the **Properties** dialog box as well as **Font** and **Color** tabs that will vary with each control. Additionally, all ActiveX controls used with InTouch share **Control Name**, **Properties**, and **Events** tabs. Tabs or properties that are unique to individual grid-based ActiveX controls are explained in their individual section.

- For more information on the **Columns** property sheet, see Common Properties of InTrack Grid-Based ActiveX Controls.
- For more information on the **Font** property sheet, see Common Font Properties of InTrack ActiveX Controls.
- For more information on the **Color** property sheet, see Common Color Properties of InTrack ActiveX Controls.
- For more information on the **Control Name** property sheet, see Using ActiveX Control Methods.
- For more information on the **Properties** property sheet, see Using ActiveX Control Methods.
- For more information on the **Events** property sheet, see Using ActiveX Control Methods.

The following InTrack grid-based ActiveX controls are described in this section:

- ITDataItemList
- ITLotDataSet
- ITLotDispCode
- ITLotInputItems
- ITLotInputMatls
- ITLotMachAvail
- ITLotUndoList
- ITLotWorkInst
- ITSubInputMatls
- ITSelector

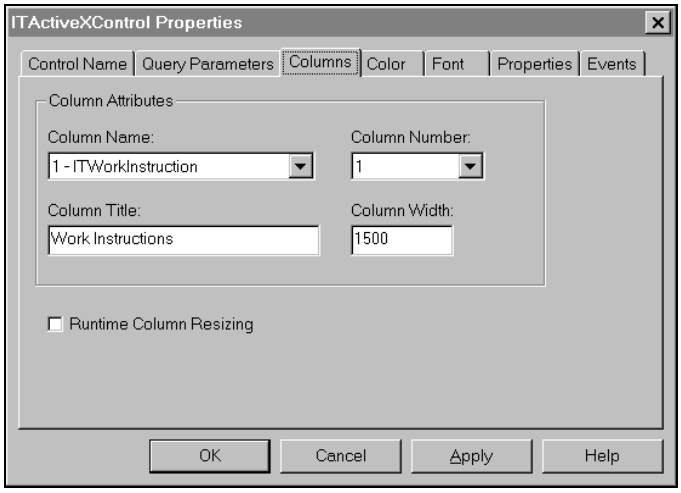
Common Properties of InTrack Grid-Based ActiveX Controls

The grid-based InTrack ActiveX controls have properties that allow you to change the appearance of the control's columns. Each control has a different set of columns, but the mechanism for configuring the columns is the same. The following attributes of an InTrack grid-based ActiveX control can be changed:

- Column Title
- Column Number
- Column Width
- Runtime Column Resizing

➢ **To view or change the columns attributes of an InTrack grid-based ActiveX control:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the **Columns** tab to activate the Columns property sheet:



3. Each attribute is changed by selecting the **Column Name**. The column name is an enumerated list, one for each column in the control. Each column name will correspond with a property of the object class represented by the ActiveX control and the total number of columns will vary to mach the number of properties of the object class.

The **Column Title** box displays the column header display. This attribute can be changed from the default title provided by the ActiveX control.

The **Column Number** list allows you to change the order that the columns are displayed in runtime. Column 1 is the leftmost column.

The **Column Width** box allows you to set a default width for the column. Setting this attribute to 0 will hide the associated column. The column width is measured in TWIPS (1440 TWIPS per inch.)

The **Runtime Column Resizing** attribute enables or disables the ability to change the widths of the columns at runtime. When column resizing is enabled, the mouse changes to a column resize pointer when placed over the column boundaries in the column header of the grid. You can then drag the mouse to change the width of the column.

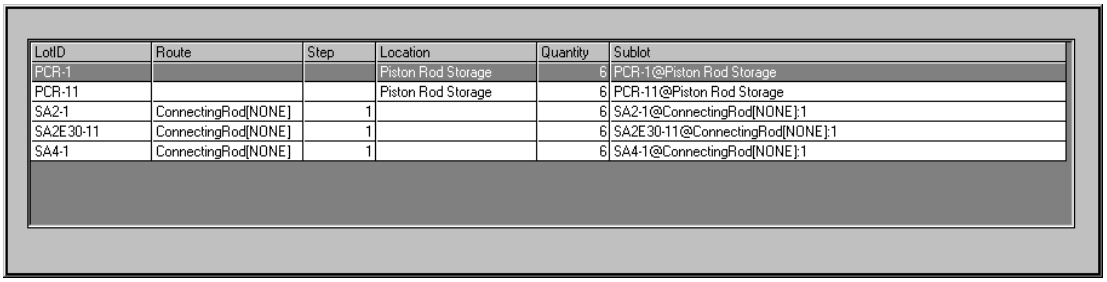
4. After you have assigned the desired InTrack grid-based ActiveX control column attributes, click **OK** to close the **Properties** dialog or **Apply** to continue working with other property tabs.

Runtime View of InTrack Grid-Based ActiveX Controls

In runtime, the InTrack Grid-Based ActiveX controls share a common appearance. The column order, width and titles in each column heading will appear as configured during development. If the **Runtime Column Resizing** option is selected during development for an ActiveX control, the runtime operator will be able to resize the column widths by placing the mouse over the column vertical gridlines, where it will change to a resize cursor, allowing for variable adjustment of the column width. Any columns to the right of the adjusted column will automatically adjust their widths to compensate for the available space.

Only ActiveX Event scripts assigned to events at design time will occur when that event is performed at runtime. No other events are included in the ActiveX control.

Note Due to the above, it is the responsibility of the application developer to supply runtime help for any InTrack ActiveX controls used in the application.



LotID	Route	Step	Location	Quantity	Sublot
PCR-1			Piston Rod Storage	6	PCR-1@Piston Rod Storage
PCR-11			Piston Rod Storage	6	PCR-11@Piston Rod Storage
SA2-1	ConnectingRod[NONE]	1		6	SA2-1@ConnectingRod[NONE]:1
SA2E30-11	ConnectingRod[NONE]	1		6	SA2E30-11@ConnectingRod[NONE]:1
SA4-1	ConnectingRod[NONE]	1		6	SA4-1@ConnectingRod[NONE]:1

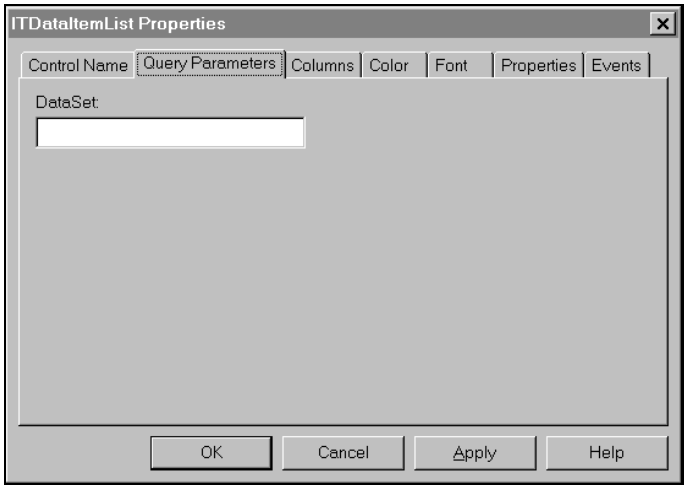
ITDataItemList

This InTrack grid-based ActiveX control provides an interface for the InTrack **Query.DataSetItem** object class. The results of this query are read-only and presented in a grid-based display.

➢ **To view or change the attributes of the ITDataItemList grid-based ActiveX control:**

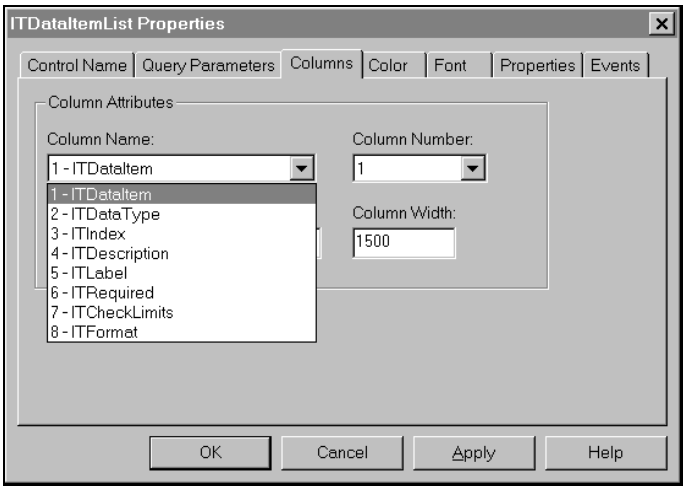
1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

In addition to the InTouch Control Name, Properties and Events, as well as the InTrack grid-based ActiveX control Columns property sheets, the Intrack ITDataIemList ActiveX control has one unique property sheet, **Query Parameters**:



The **DataSet** box must be completed with the VersionKey identifying the data set template for which to retrieve data items.

The results of the query are presented in 8 columns corresponding to the Query.DataSetItem object class:



For more information on OLE References, see your *InTrack OLE Reference*.

The ITDataItemList grid-based ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are RequiredBackColor in addition to the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITDataItemList control are double-click and selection changed, either of which may have Event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

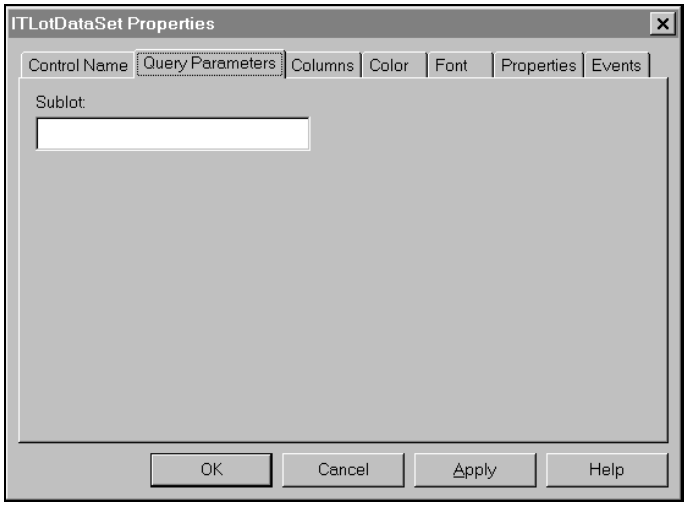
ITLotDataSet

This InTrack grid-based ActiveX control provides an interface for the InTrack **Query.LotDataSet** object class. The results of this query are read-only and presented in a grid-based display.

➢ **To view or change the attributes of the ITLotDataSet grid-based ActiveX control:**

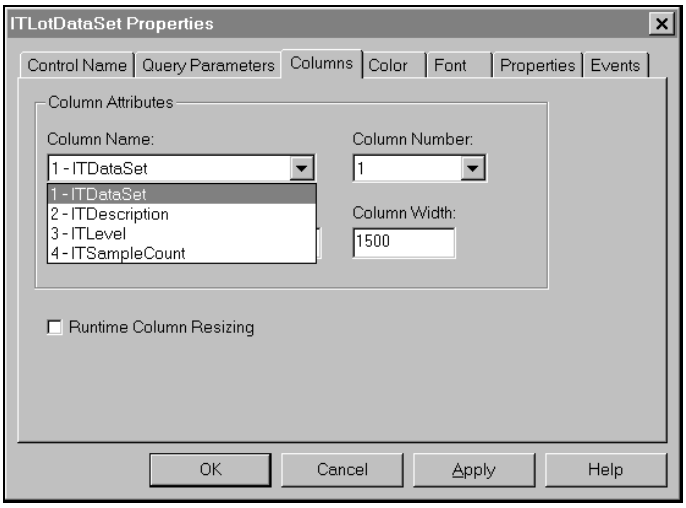
1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

In addition to the InTouch Control Name, Properties and Events, as well as the InTrack grid-based ActiveX control Columns property sheets, the Intrack ITLotDataSet ActiveX control has one unique property sheet, **Query Parameters**:



The **Sublot** box must be completed with the SublotKey identifying the sublot for which to retrieve data set templates.

The results of the query are presented in 4 columns corresponding to the Query.LotDataSet object class:



For more information on OLE References, see your *InTrack OLE Reference*.

The ITLotDataSet grid-based ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITLotDataSet control are double-click and selection changed, either of which may have event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

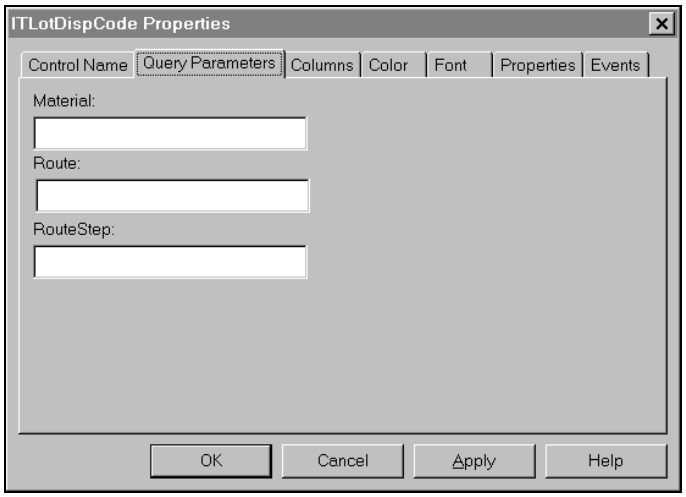
ITLotDispCode

This InTrack grid-based ActiveX control provides an interface for the InTrack **Query.Disposition** object class. The results of this query are read-only and presented in a grid-based display.

➢ **To view or change the attributes of the ITLotDispCode grid-based ActiveX control:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

In addition to the InTouch Control Name, Properties and Events, as well as the InTrack grid-based ActiveX control Columns property sheets, the Intrack ITLotDispCode ActiveX control has one unique property sheet, **Query Parameters**:

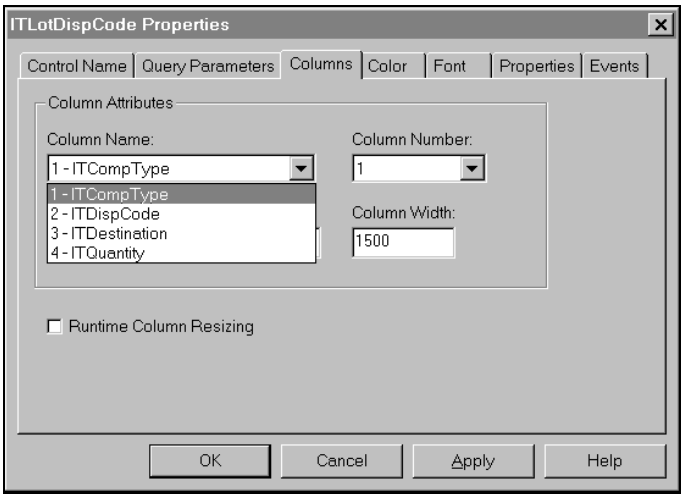


The **Material** box must be completed with the VersionKey identifying a target material associated with the route.

The **Route** box must be completed with the VersionKey identifying the route for which to retrieve disposition codes.

The **RouteStep** box must be completed with the NameKey identifying a route step associated with the route.

The results of the query are presented in four columns corresponding to the Query.Disposition object class:



For more information on OLE References, see your *InTrack OLE Reference*.

The ITLotDispCode grid-based ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITLotDispCode control are double-click and selection changed, either of which may have event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

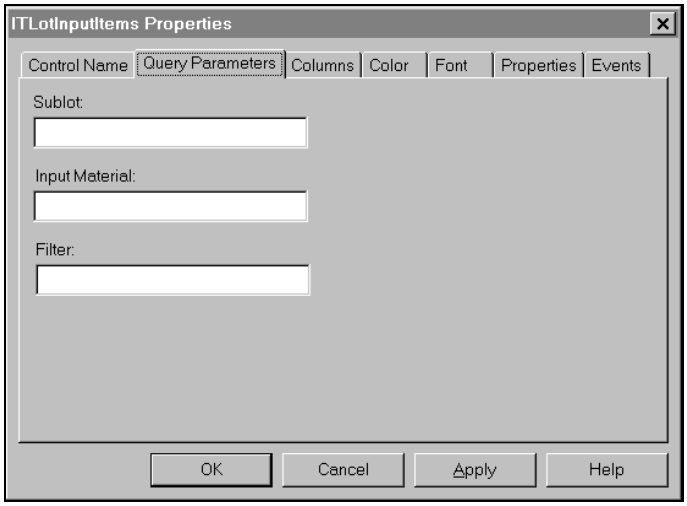
ITLotInputItems

This InTrack grid-based ActiveX control provides an interface for the InTrack **Sublot.GetConsumeItems** object class. The results of this query are read-only and presented in a grid-based display.

➢ **To view or change the attributes of the ITLotInputItems grid-based ActiveX control:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

In addition to the InTouch Control Name, Properties and Events, as well as the InTrack grid-based ActiveX control Columns property sheets, the Intrack ITLotInputItems ActiveX control has one unique property sheet, **Query Parameters**:

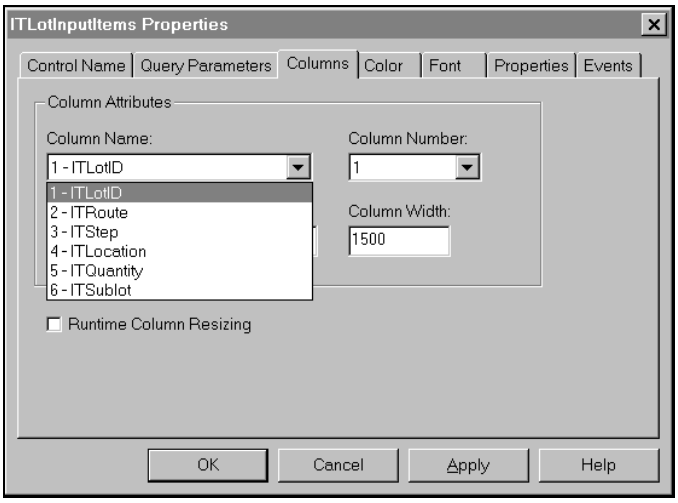


The **Sublot** box must be completed with the SublotKey identifying the sublot on which to perform transactions.

The **Input Material** box must be completed with the VersionKey identifying the material (for example, bottles[2]); use NONE if the material is not versioned.

The **Filter** box must be completed with the the SQL predicate that serves as the basis for filtering (for example, a WHERE clause).

The results of the query are presented in six columns corresponding to the Sublot.GetConsumeItems object class:



For more information on OLE References, see your *InTrack OLE Reference*.

The ITLotInputItems grid-based ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITLotInputItems control are double-click and selection changed, either of which may have event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

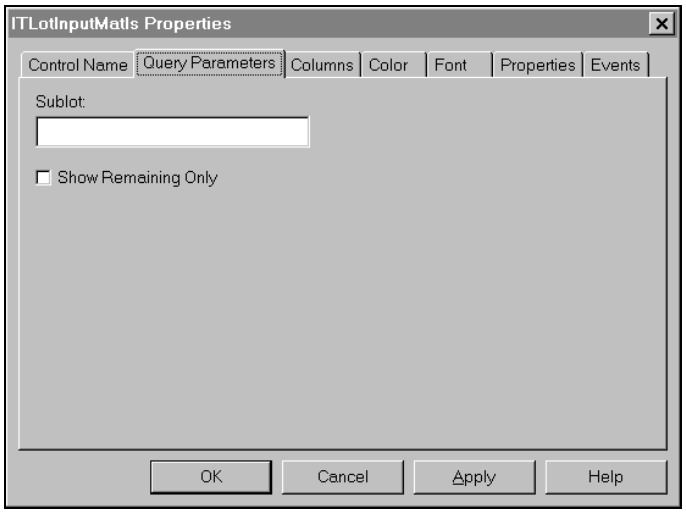
ITLotInputMatls

This InTrack grid-based ActiveX control provides an interface for the InTrack **Sublot.GetConsumableMaterials** method. The results of this query are read-only and presented in a grid-based display.

➢ **To view or change the attributes of the ITLotInputMatls grid-based ActiveX control:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

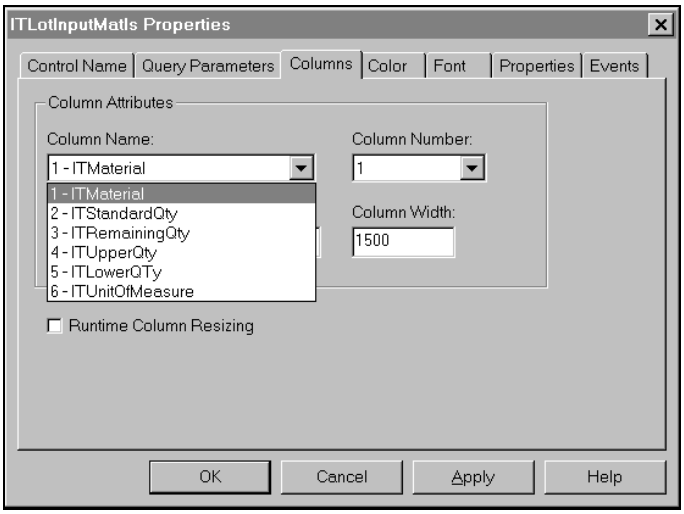
In addition to the InTouch Control Name, Properties and Events, as well as the InTrack grid-based ActiveX control Columns property sheets, the Intrack ITLotInputMatls ActiveX control has one unique property sheet, **Query Parameters**:



The **Sublot** box must be completed with the SublotKey identifying the sublot on which to perform transactions.

Show Remaining Only may be selected as an option.

The results of the query are presented in six columns corresponding to the Sublot.GetConsumableMaterials method:



For more information on OLE References, see your *InTrack OLE Reference*.

The ITLotInputMatls grid-based ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITLotInputMatls control are double-click and selection changed, either of which may have event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

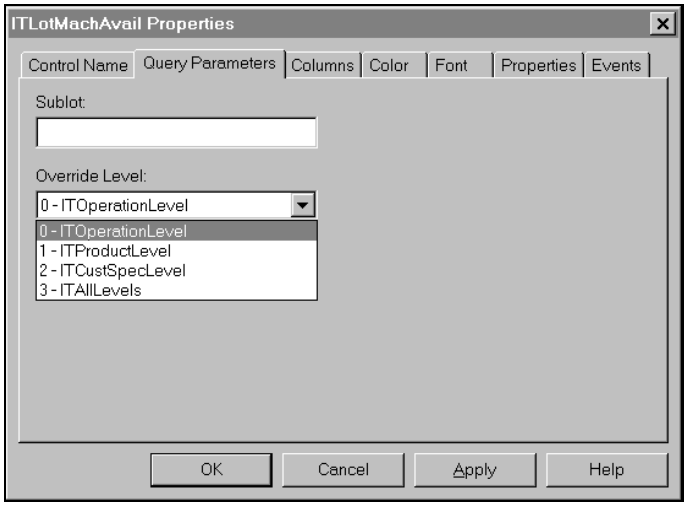
ITLotMachAvail

This InTrack grid-based ActiveX control provides an interface for the InTrack **Query.LotMachine** object class. The results of this query are read-only and presented in a grid-based display.

➢ **To view or change the attributes of the ITLotMachAvail grid-based ActiveX control:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

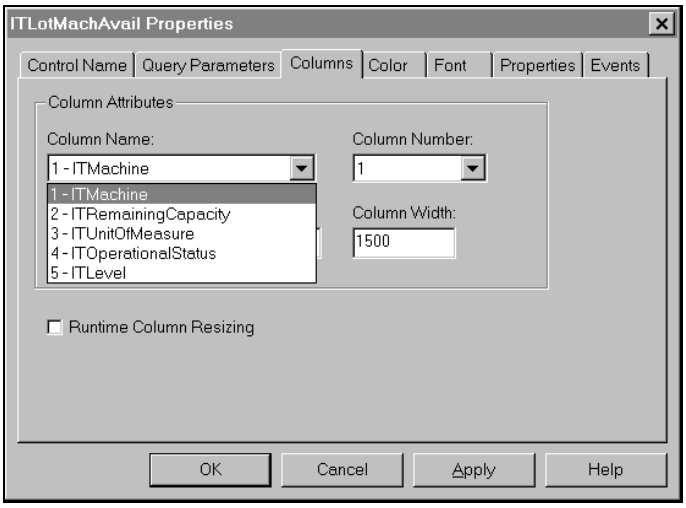
In addition to the InTouch Control Name, Properties and Events, as well as the InTrack grid-based ActiveX control Columns property sheets, the Intrack ITLotMachAvail ActiveX control has one unique property sheet, **Query Parameters**:



The **Sublot** box must be completed with the SublotKey identifying the sublot for which to retrieve machine information.

An **Override Level** must be selected.

The results of the query are presented in five columns corresponding to the Query.LotMachine object class:



For more information on OLE References, see your *InTrack OLE Reference*.

The ITLotMachAvail grid-based ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITLotMachAvail control are double-click and selection changed, either of which may have event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

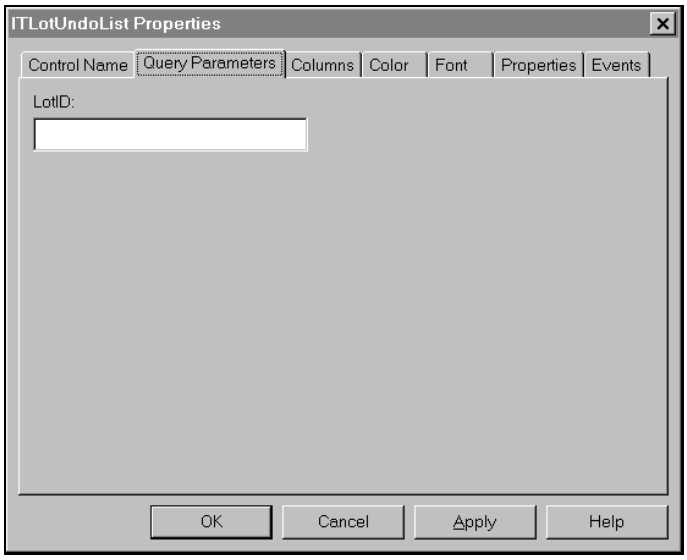
ITLotUndoList

This InTrack grid-based ActiveX control provides an interface for the InTrack **Query.LotUndo** object class. The results of this query are read-only and presented in a grid-based display.

➢ **To view or change the attributes of the ITLotUndoList grid-based ActiveX control:**

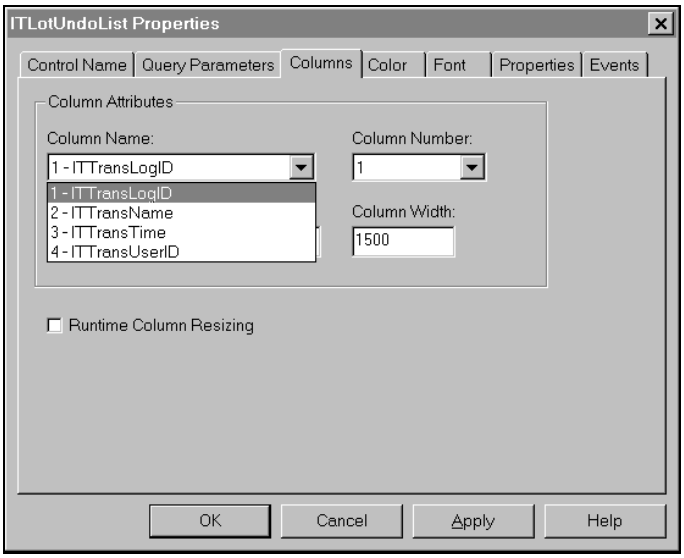
1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

In addition to the InTouch Control Name, Properties and Events, as well as the InTrack grid-based ActiveX control Columns property sheets, the Intrack ITLotUndoList ActiveX control has one unique property sheet, **Query Parameters**:



The **LotID** box must be completed with the LotKey identifying the lot for which to retrieve undo transaction information.

The results of the query are presented in four columns corresponding to the Query.LotUndo object class:



For more information on OLE References, see your *InTrack OLE Reference*.

The ITLotUndoList grid-based ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITLotUndoList control are double-click and selection changed, either of which may have event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

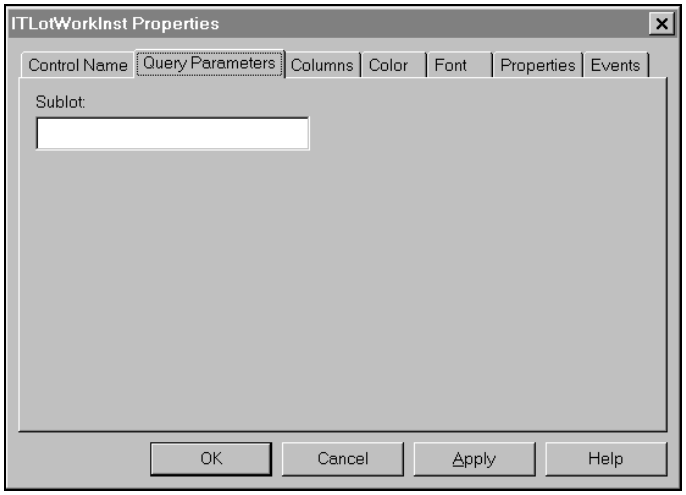
ITLotWorkInst

This InTrack grid-based ActiveX control provides an interface for the InTrack **Query.LotWorkInstructions** object class. The results of this query are read-only and presented in a grid-based display.

➢ **To view or change the attributes of the ITLotWorkInst grid-based ActiveX control:**

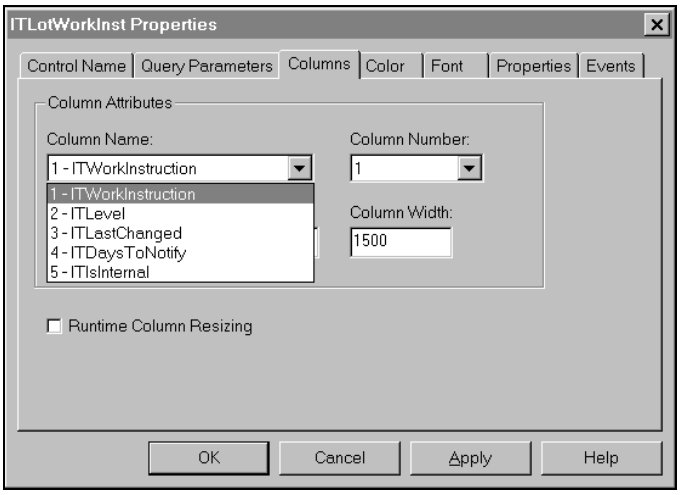
1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

In addition to the InTouch Control Name, Properties and Events, as well as the InTrack grid-based ActiveX control Columns property sheets, the Intrack ITLotWorkInst ActiveX control has one unique property sheet, **Query Parameters**:



The **Sublot** box must be completed with the SublotKey identifying the sublot for which to retrieve work instructions.

The results of the query are presented in five columns corresponding to the Query.LotWorkInstructions object class:



For more information on OLE References, see your *InTrack OLE Reference*.

The ITLotWorkInst grid-based ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are ChangedBackColor in addition to the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITLotWorkInst control are double-click and selection changed, either of which may have event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

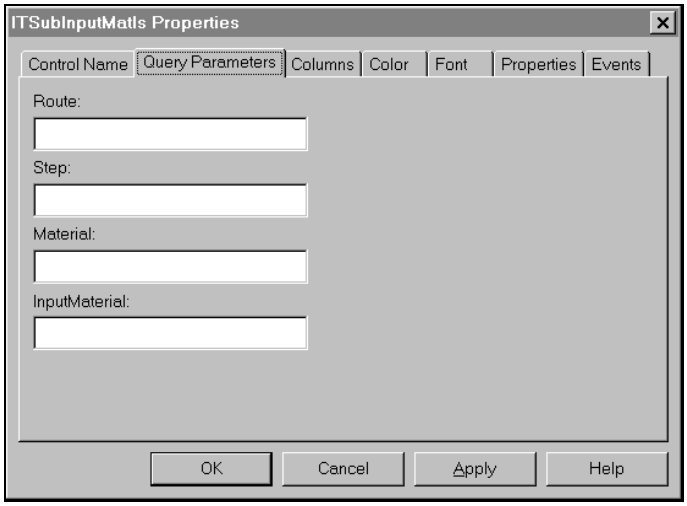
ITSubInputMatls

This InTrack grid-based ActiveX control provides an interface for the InTrack **Query.SubstituteMaterialInput** object class. The results of this query are read-only and presented in a grid-based display.

➢ **To view or change the attributes of the ITSubInputMatls grid-based ActiveX control:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

In addition to the InTouch Control Name, Properties and Events, as well as the InTrack grid-based ActiveX control Columns property sheets, the Intrack ITSubInputMatls ActiveX control has one unique property sheet, **Query Parameters**:



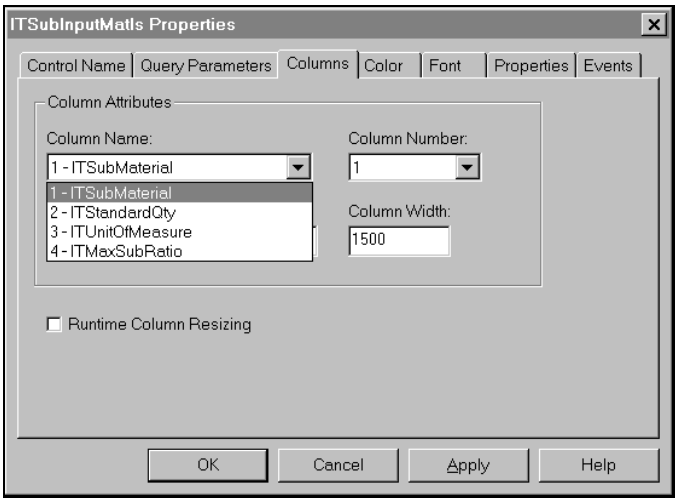
The **Route** box must be completed with the VersionKey identifying the route for which to retrieve a substitute input material.

The **Step** box must be completed with the NameKey identifying a route step associated with the route.

The **Material** box must be completed with the VersionKey identifying a target material associated with the route

The **InputMaterial** box must be completed with the VersionKey identifying an input material for which to retrieve a substitute input material.

The results of the query are presented in four columns corresponding to the Query.SubstituteMaterialInput object class:



For more information on OLE References, see your *InTrack OLE Reference*.

The ITSubInputMatls grid-based ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITSubInputMatls control are double-click and selection changed, either of which may have event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

ITSelector

This InTrack grid-based ActiveX control provides an interface for the InTrack **Query.SQL** object class. The results of this query are read-only and presented in a grid-based display. Data displayed is not automatically updated.

Note The following database table columns must be updated by the application prior to viewing data by the ITSelector ActiveX control:

- Sublot/EstimatedCompletion
- Sublot/CriticalRatio
- Sublot/DataCollectDone
- Sublot/ConsumptionDone

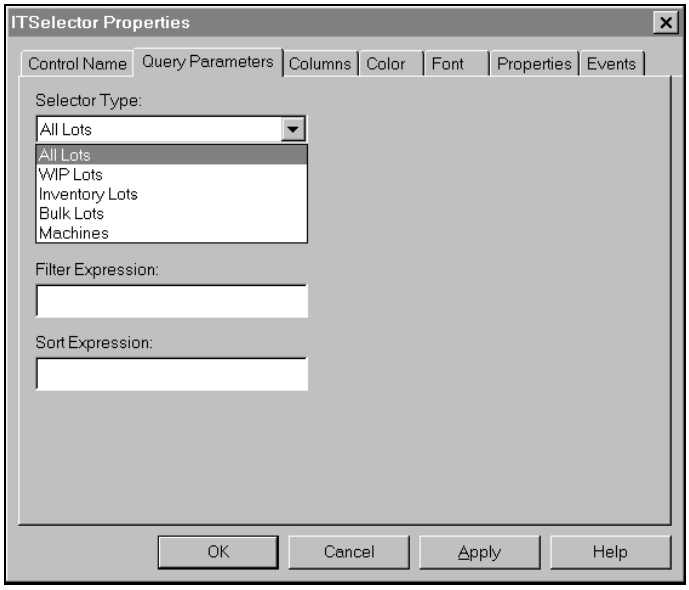
➢ **To view or change the attributes of the ITSelector grid-based ActiveX control:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

In addition to the InTouch Control Name, Properties and Events, the Intrack ITSelector ActiveX control has two unique property sheets, **Query Parameters** and **Column Select**. The Column Select property sheet replaces the standard Column Name property sheet for grid-based InTrack ActiveX controls.

➢ **To view or change the attributes of the ITSelector Query Parameters property sheet:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the **Query Parameters** tab to activate the Query Parameters property sheet



The **Selector Type** must be selected from the list.

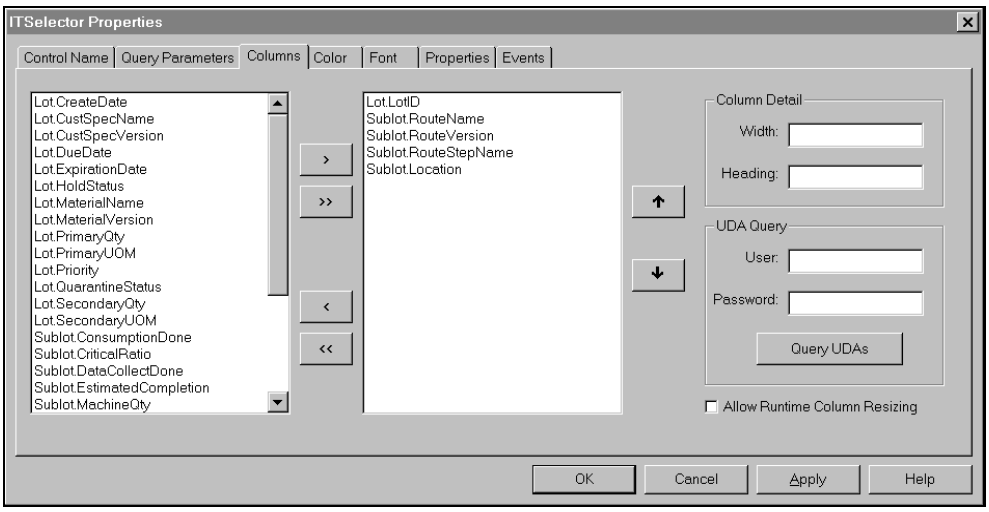
The **Filter Expression** box may be used to provide the name of the SQL WHERE predicate that serves as the basis for filtering the list of rows.

Note When searching multiple tables, the predicate should contain the join condition(s).

The **Sort Expression** box may be filled with an Order By SQL clause.

➢ To view or change the attributes of the ITSelector Column Select property sheet:

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the **Column Select** tab to activate the Column Select property sheet:



The left two-thirds of the Column Select property sheet consists of two windows. The left window displays available query items. The right window displays selected query items. Individual items may be selected in either window and transferred to the other window with single arrow keys. The double arrow keys are used to transfer the entire window contents.

The up and down arrows to the right of the item windows are used to move a selected query item up or down the list. This will affect the columns displayed in runtime, with the topmost item shown at the far left of the grid.

The right third of the Column Select property sheet consists of two boxes. The upper box, **Column Detail**, allows you to select the column width and heading for the runtime display of the grid.

The lower box allows you to enter your user name and password for the InTrack database to access user-defined attributes (UDAs). Clicking the **Query UDAs** button will cause a query of the Intrack database for UDAs, with the results added to any query items in the left window for inclusion in your runtime grid.

For more information on OLE References, see your *InTrack OLE Reference*.

The ITSelector grid-based ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are ClosedColor, IdleColor, InProcessColor, InQuarantineColor, InUseColor, NormalColor, OfflineColor, OnHoldColor and QueuedColor in addition to the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITSelector control are double-click and selection changed, either of which may have event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

InTrack Listbox ActiveX Controls

All InTrack listbox ActiveX controls share the same **Font** and **Color** tabs in the **Properties** dialog box. Additionally, all ActiveX controls used with InTouch share **Control Name**, **Properties**, and **Events** tabs. Tabs that are unique to individual listbox ActiveX controls are explained in their individual section.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.
- For more information on the **Control Name** property sheet, see *Using ActiveX Control Methods*.
- For more information on the **Properties** property sheet, see *Using ActiveX Control Methods*.
- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

The following InTrack listbox ActiveX controls are described in this section:

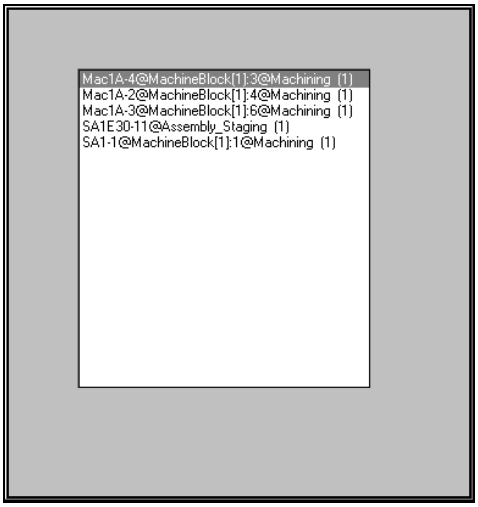
- ITLotMergeList
- ITMachineInfoList
- ITSampleList
- ITQueryList

Runtime View of InTrack Listbox ActiveX Controls

In runtime, the InTrack Listbox ActiveX controls share a common appearance. The queried items will be displayed in a single column listbox. There is no runtime adjustment of the listbox size.

Only ActiveX Event scripts assigned to events at design time will occur when that event is performed at runtime. No other events are included in the ActiveX control.

Note: It is the responsibility of the application developer to supply runtime help for any InTrack ActiveX controls used in the application.



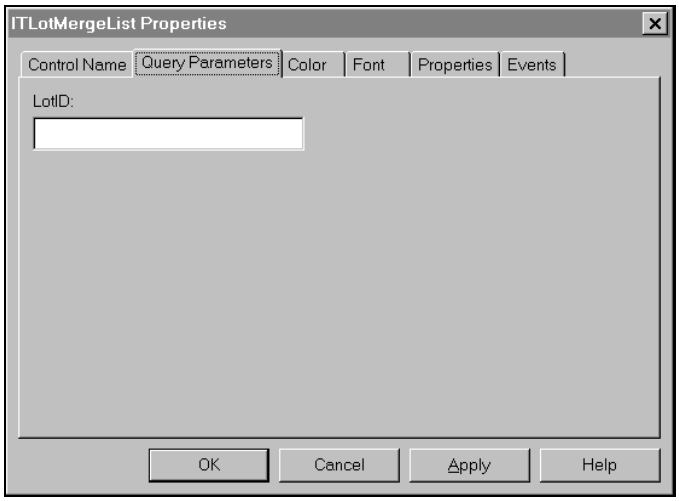
ITLotMergeList

This InTrack listbox ActiveX control provides an interface for the InTrack **Query.MergeLot** object class. The results of this query are read-only and presented in a standard windows listbox display.

➢ **To view or change the attributes of the ITLotMergeList listbox ActiveX control:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

In addition to the InTouch Control Name, Properties and Events, the InTrack ITLotMergeList ActiveX control has one unique property sheet, **Query Parameters**:



The **LotID** box must be completed with the LotKey identifying the lot for which to retrieve sublots eligible to merge.

For more information on OLE References, see your *InTrack OLE Reference.*

The ITLotMergeList ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITLotMergeList control are double-click and selection changed, either of which may have event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods.*

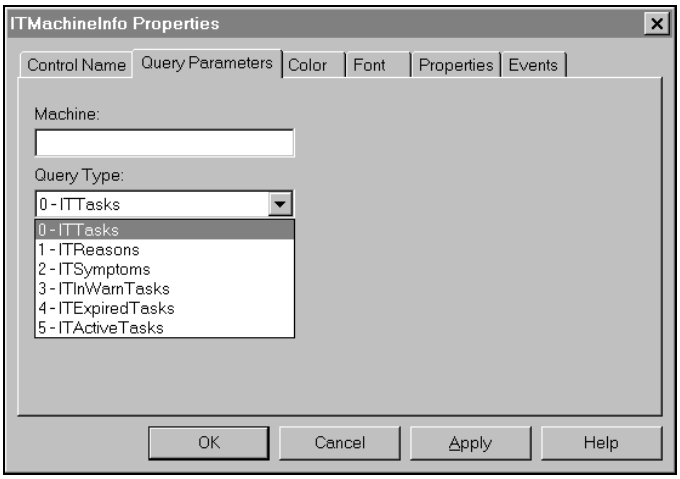
ITMachineInfoList

This InTrack listbox ActiveX control provides an interface for the InTrack **Machine**, **Query.MachineTask**, **Query.ExpiredMachineTask** and **Query.InWarningMachineTask** object classes. The results of this query are read-only and presented in a standard windows listbox display.

➢ **To view or change the attributes of the ITMachineInfoList listbox ActiveX control:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

In addition to the InTouch Control Name, Properties and Events, the InTrack ITMachineInfoList ActiveX control has one unique property sheet, **Query Parameters**:



The **Machine** box must be completed with the identifier for the machine to be Queried.

A **Query Type** must be selected.

For more information on OLE References, see your *InTrack OLE Reference*.

The ITMachineInfoList ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITMachineInfoList control are double-click and selection changed, either of which may have event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

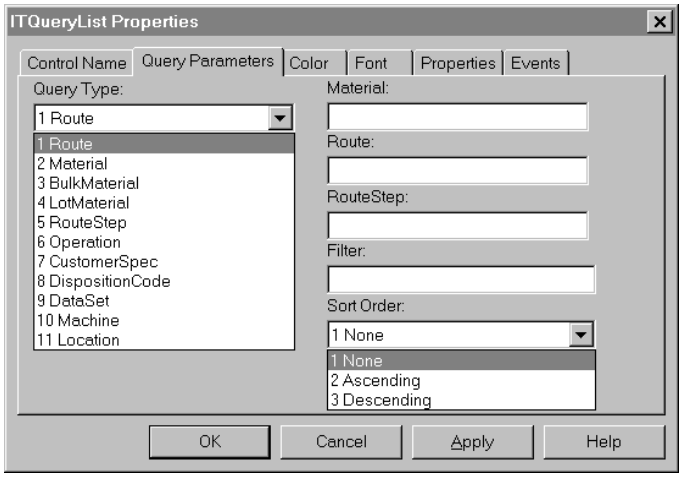
ITQueryList

This InTrack listbox ActiveX control provides an interface for the InTrack **Query.SQL** object class. The results of this query are read-only and presented in a standard windows listbox display.

➢ **To view or change the attributes of the ITQueryList listbox ActiveX control:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

In addition to the InTouch Control Name, Properties and Events, the Intrack ITQueryList ActiveX control has one unique property sheet, **Query Parameters**:



The **Query Type** must be selected from the list.

The **Material** box must be completed with the VersionKey identifying the target material.

The **Route** box must be completed with the VersionKey identifying the route for which to retrieve a substitute input material.

The **Route Step** box must be completed with the NameKey identifying a route step associated with the route.

The **Filter** box may be used to provide the name of the SQL WHERE predicate that serves as the basis for filtering the list.

The **Sort Order** must be selected from the list.

For more information on OLE References, see your *InTrack OLE Reference*.

The ITQueryList ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITQueryList control are double-click and selection changed, either of which may have event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

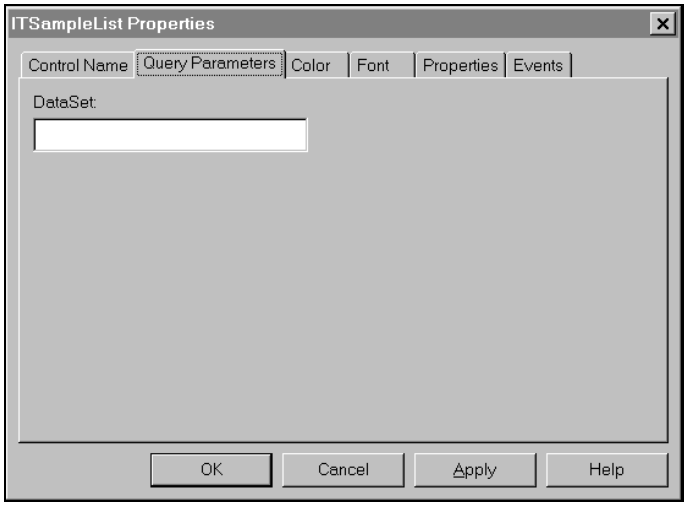
ITSampleList

This InTrack listbox ActiveX control provides an interface for the InTrack **Query.DataSetSample** object class. The results of this query are read-only and presented in a standard windows listbox display.

➢ **To view or change the attributes of the ITSampleList listbox ActiveX control:**

1. Double-click the ActiveX control. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the appropriate tab to activate the desired property sheet

In addition to the InTouch Control Name, Properties and Events, the Intrack ITSampleList ActiveX control has one unique property sheet, **Query Parameters**:



The **DataSet** box must be completed with the VersionKey identifying the data set template for which to retrieve data samples

For more information on OLE References, see your *InTrack OLE Reference*.

The ITSampleList ActiveX control uses the standard InTrack ActiveX control **Font** and **Color** property sheets. The supported Color attributes are the standard BackColor and ForeColor attributes.

- For more information on the **Font** property sheet, see *Common Font Properties of InTrack ActiveX Controls*.
- For more information on the **Color** property sheet, see *Common Color Properties of InTrack ActiveX Controls*.

The two **Events** supported by the ITSampleList control are double-click and selection changed, either of which may have event scripts attached during development. Data from the database tables are not automatically updated, they must be updated by an Event script.

- For more information on the InTouch Events property sheet, see *Using ActiveX Control Methods*.

CHAPTER 3

The InTrack TagList Editor

This chapter describes how to create, delete, and modify TagLists using the TagList Editor. Information on TagList templates, parameters, and tagnames also is included.

Note: This chapter assumes that you know how to create and use InTouch tagnames.

For more information on InTouch tagnames, see your *InTouch User's Guide*.

Contents

TagLists

A TagList is a collection of tagnames referenced by one or more labels or names. For example, the TagList "List1" might contain three tags, "TagA," "TagB," and "TagC." This TagList can be used by one or more InTrack System commands or OLE object methods that will link the tagnames in the list to command or method parameters so that the system can automatically perform data acquisition and processing. Thus, when an InTrack System command is issued or an OLE object method is called using that TagList, all the parameters required to process that command or method will be retrieved from the tagnames in the TagList.

Important: I/O tags used in a TagList must be active before an InTrack System command or OLE object method using them can be issued. A tag is active if it is: (1) displayed in a WindowViewer window, (2) used in a script, (3) being alarmed or event logged, (4) displayed in a real time trend or historically logged, (5) the I/O topic is marked as Advise All.

TagLists are created using the InTrack TagList Editor. When you create a TagList, you first specify a TagList type and an available template for the TagList. After a specific template is chosen and the TagList is created, all the parameters for processing an InTrack System command or an OLE object method using that template, and their required data types, will appear in the TagList Editor. Some parameters are required to be defined, some have default values, and others are optional.

Command TagLists

A Command TagList is a group of tagnames used when issuing a System command.

The Command TagList type is supported for InTrack applications that have been migrated from previous versions of InTrack.

Data Collection TagLists

A Data Collection TagList is a group of tagnames used to read/write values for data collection items. Data collection items are the individual data points to be collected at runtime for an operation or a machine. Data items are defined for data set templates in the ModelMaker. Any defined data set template can be used to create a Data Collection TagList. If a Data Collection TagList is used, the operator is not required to manually enter all the data item values for the data collection. Data is collected automatically from the tagnames defined in the Data Collection TagList and then entered into the database.

Object Details (UDAs) TagLists

An Object Details (UDAs) TagList is a group of tagnames used to return various details, or attributes, for a particular activity object. Object details include defined characteristics of an object, such as the name of the object, the quantity, its quarantine status, any user-defined attributes, etc. UDAs (user-defined attributes) are fields, or columns, created in the database that are usually specific to the manufacturing process (e.g., cost, machine ID).

The Object Details (UDAs) TagList type is supported for InTrack.

Diagnostic TagLists

A Diagnostic TagList is a group of tagnames used to return values indicating the status of any InTrack transaction. A transaction is a request to the InTrack system to find, enter, change, or return information about an activity or structural object in the relational database.

Query TagLists

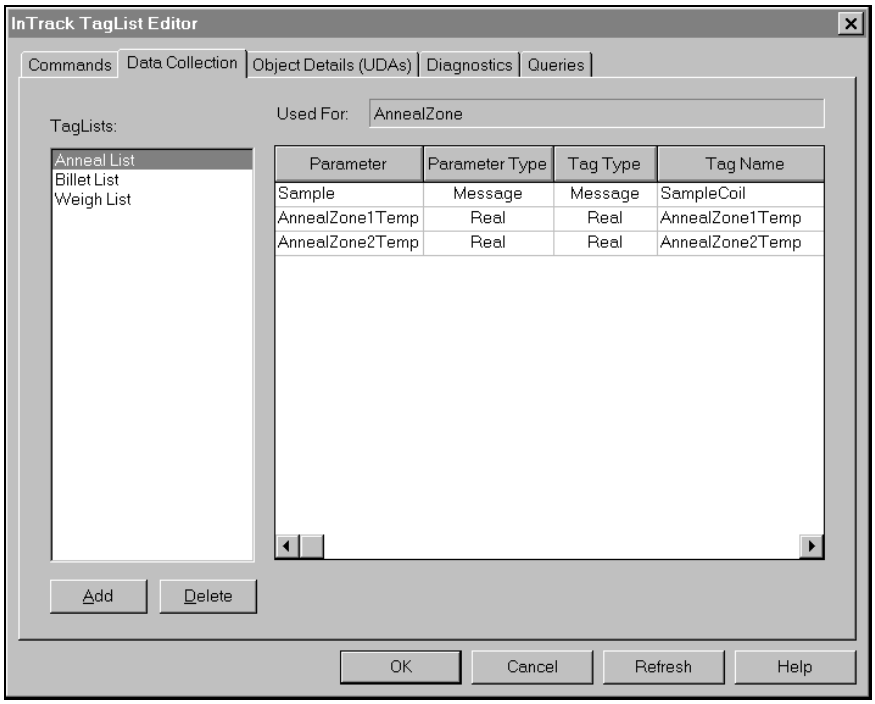
A Query TagList is a group of tagnames used when issuing a database query or when getting results for a database query. A query is a request to the InTrack system to find all the objects in the relational database based on specific criteria.

InTrack TagList Editor

All TagLists are created using the InTrack TagList Editor.

➢ **To access the TagList Editor:**

1. Start WindowMaker.
2. In WindowMaker, choose the **Special,** InTrack, TagLists command to open the InTrack TagList Editor:

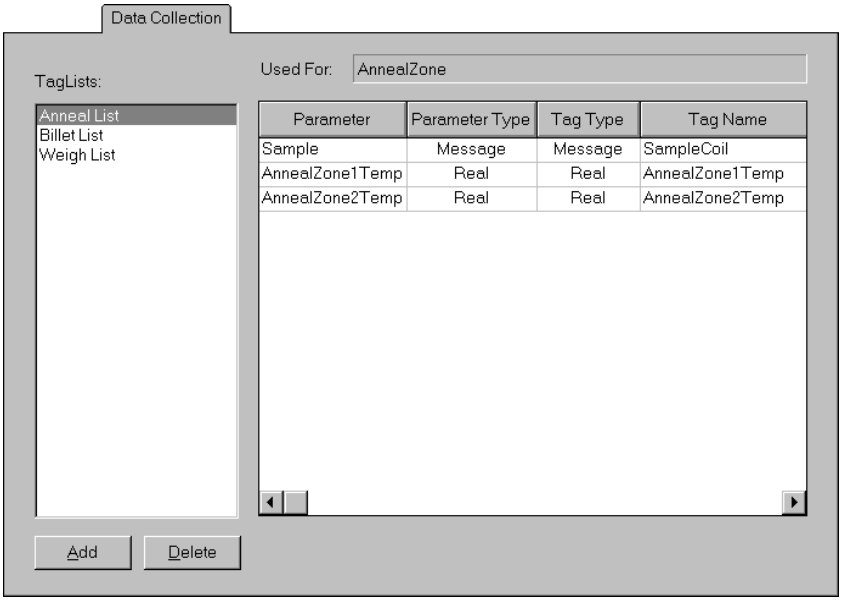


Details for all TagLists of a particular type are displayed when you click the corresponding tab in the TagList Editor. For example, to create or view Data Collection TagLists, click the Data Collection tab.

Note: Although there are tabs for five types of TagLists in the InTrack TagList Editor, only one type can be defined in InTrack - Data Collection. The other four TagList types - Command, Object Details (UDAs), Diagnostic, and Query - are no longer supported in InTrack.

TagList Tab Features

Features of the TagList Editor tab are described in this section. Each Data Collection TagList has the same layout. For example, the layout for the TagList "CoolerTemp" is shown below:



All defined TagLists appear in the **TagLists:** window. To modify a TagList name or delete an existing TagList, the name must be highlighted in this window.

The name of the template for the TagList selected in the TagLists window appears in the **Used For:** field.

All possible input and output parameters for a selected TagList template automatically appear in the **Parameter** column. Parameters are the individual informational items required to complete a transaction or the values returned as a result of a transaction.

All default data types for each parameter automatically appear in the **Parameter Type** column. The parameter type specifies how the parameter value is stored in the InTrack database. Parameter types cannot be changed.

Tag types for all default parameters for a selected TagList template automatically appear in the **Tag Type** column. The tag type specifies how the tagname value is handled by InTouch. Tag types cannot be changed.

Tagnames are linked to the TagList parameters in the **Tag Name** column. Existing tagnames can be linked to the parameters, or new tagnames can be created..

TagList Operations

To add a TagList for the selected tab, click **Add**. The Add TagList dialog box will appear.

To delete a TagList for the selected tab, click **Delete**.

To save the input and close the TagList Editor, click **OK**.

To close the TagList Editor and cancel the input, click **Cancel**.

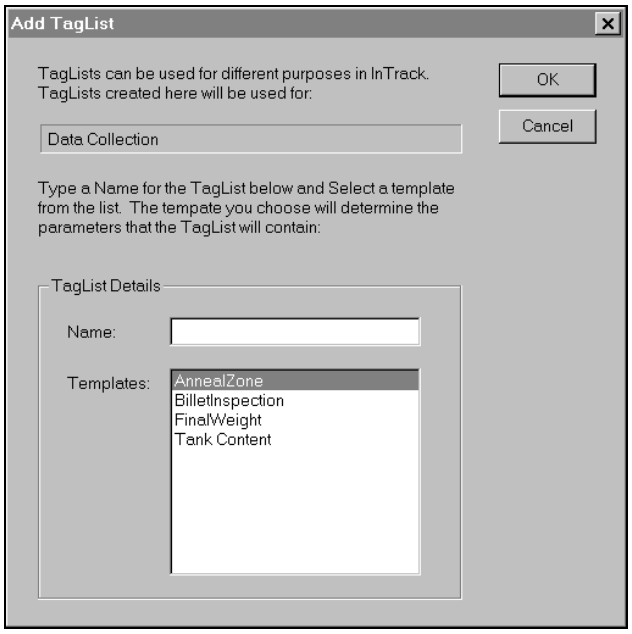
To refresh the display, click **Refresh**. You must refresh the display after any changes are made to the manufacturing model for the changes to be reflected in the TagList Editor.

To obtain online Help for the TagList Editor, click **Help**.

Creating a TagList

➢ **To create a TagList:**

1. Click the **Data Collection** tab.
2. Click **Add** on the TagList tab to open the **Add TagList** dialog box:



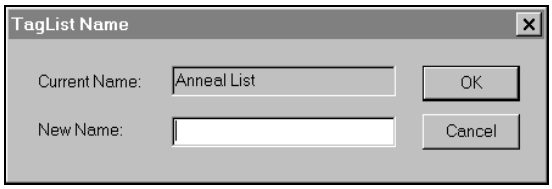
3. In the **Name** field, type the name of the new TagList.
4. In the **Templates** list box, choose a template for the new TagList. The template you choose will determine the input and output parameters for the TagList.
5. Click **OK**. The parameters for the template will appear in the spreadsheet of the TagList Editor.
6. Link tagnames to the parameters.
7. When you are finished, click **OK** to save the changes.

Changing a TagList Name

A TagList name can be changed at any time; however, if you change a TagList name, you also must change the TagList name for all scripts that reference it. Calling an InTrack System command or an OLE object method with an obsolete TagList name as a parameter will cause the command or method to fail.

➢ **To change a TagList name:**

1. Click the **Data Collection** tab.
2. In the **TagLists** field, double-click the name of the TagList. The **TagList Name** dialog box appears, with the current TagList name listed in the **Current Name** field:



3. Enter the new name of the TagList in the **New Name** field and click **OK**.

Deleting a TagList

A TagList can be deleted at any time; however, if you delete a TagList, you must also delete the TagList in all scripts that reference it. Calling an OLE object method with an obsolete TagList as a parameter will cause the method to fail.

➢ **To delete a TagList:**

1. Click the **Data Collection** tab.
2. In the **TagLists** field, highlight the name of the TagList.
3. Click **Delete**. A message box will appear, prompting you to confirm the deletion:

TagList Templates

A TagList template is a predefined set of input and/or output parameters. Input parameters define the values needed to complete a transaction. Output parameters define the values to be returned from the database after the transaction is complete. TagList templates are provided to simplify application development; when you choose a template, all the required parameters to complete the OLE object method using the TagList will appear in the spreadsheet of the TagList Editor.

Each data set template structural object in the InTrack database has a corresponding TagList template displayed in the Templates list box of the Add TagList dialog box.

An unlimited number of TagLists can be created using a single template.

TagList Parameters

TagList parameters define the values to be written to or returned from the database. All the parameters required to complete a specific OLE object method and return output values (if any) are grouped into sets, called templates.

Any tagname can be linked to any TagList parameter, provided that the tagname is of the correct type (e.g., integer, message, etc.). When tagnames are linked to TagList parameters, all the values required to call an OLE object method are automatically supplied by the current values in the tags, eliminating the need for manually entering data each time a method is called. When data is returned into a tagname for a parameter, the value of the tagname can be viewed in WindowViewer using an InTouch animation link, or used in another script that references that tagname.

Some TagList parameters have default input values that are used by the OLE object method unless a different value is specified in the script. For example, if no value is specified for the *MaterialVer* parameter, the value "NONE" will be used.

There are two types of TagList parameters: input parameters and output parameters.

Input Parameters

Input parameters define the values to be used by the OLE object method (written to the InTrack database at runtime). When an OLE object method is called, the information required to complete the method is supplied by the values in the tags that are linked to the input parameters.

Output Parameters

Output parameters specify the values to be returned from the InTrack database at runtime. When an OLE object method is called, the information found for the method is automatically returned in the tagnames linked to the output parameters.

TagLists as Parameter Values

Data Collection TagLists can be referenced as parameter values in OLE object method calls.

Parameter Types

The parameter type specifies how a parameter value is stored in the InTrack relational database. Since InTouch does not support some of the parameter types, these values must be translated into valid InTouch tagname types before they can be handled by the InTouch system.

For example, the parameter value for the *BackDate* parameter is stored in the database as a date-type value, but is handled by InTouch as a message-type value. The InTrack system automatically assigns parameter types and their corresponding tagname types, and performs all the necessary translations between them.

Parameter types and tagname types cannot be changed.

TagList Tags

The types of tags used in a TagList is determined by how OLE object methods are processed. TagList tags can either be local variables (memory type) or dedicated to actual variables supplied by factory machines (I/O type).

Note: Do not confuse tag types (e.g., memory, I/O) with data types (e.g., discrete, integer). Each tag type supports all data types.

Memory Tags

Memory tags allow you to create "global" TagLists since they reside in the local PC and can be reassigned to new values at any time during runtime. A memory tag can reference a value for a particular OLE object method, be reassigned to a different value through a script, and reference a new value for a different OLE object method. If used in this way, memory tags can be used as temporary storage for values supplied by multiple I/O tags.

For more information on memory tags, see the *InTouch User's Guide*.

I/O Tags

I/O tags are useful when creating a TagList that is specific for an InTrack System command or OLE object method. I/O tags, however, do not provide the same level of flexibility as memory tags. When a parameter is linked to a I/O tag type, the value for the parameter is a specific input/output point or variable supplied by another application such as a I/O server attached to a factory machine.

The only way that two I/O tags can read different values into a single TagList parameter is if the I/O tags are first assigned to a memory tag.

Updating Use Counts

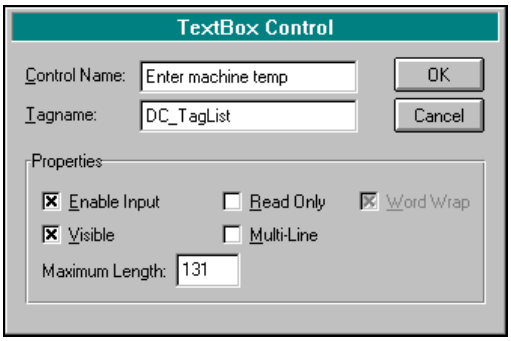
InTouch maintains a use count for tags defined in the database. A use count is the total number of times a tag is used in an animation link or script in an application. When you choose the Special/Update Use Counts command in WindowMaker, InTouch will update the count for all tags currently being used. You can delete all unused tags (those having a use count of zero) by choosing the Special/Delete Unused Tags command in WindowMaker.

However, in order for an InTrack TagList tag to have a use count, it must also be used in an InTouch animation link or script. If TagList tags are not used in animation links, they ***are not guaranteed to be updated*** (giving them a use count of zero) and ***may be deleted*** accidentally if you issue the Special/Delete Unused Tags command and click All.

User-Defined Tag Values

The value for any tag can be supplied by the runtime operator using any InTouch input animation link (e.g., a Windows Control text box). All input animation links can be assigned a tagname. When the runtime operator enters a value in the control, the value is read into the assigned tagname, and any script that references that tagname will use the entered value when processed.

For example, a Windows Control text box is created to allow user input for the temperature collected for a machine. The configuration for the text box is:



When the runtime operator enters a machine temperature into the Windows Control text box, the temperature is used for the value of the tagname "DC_TagList."

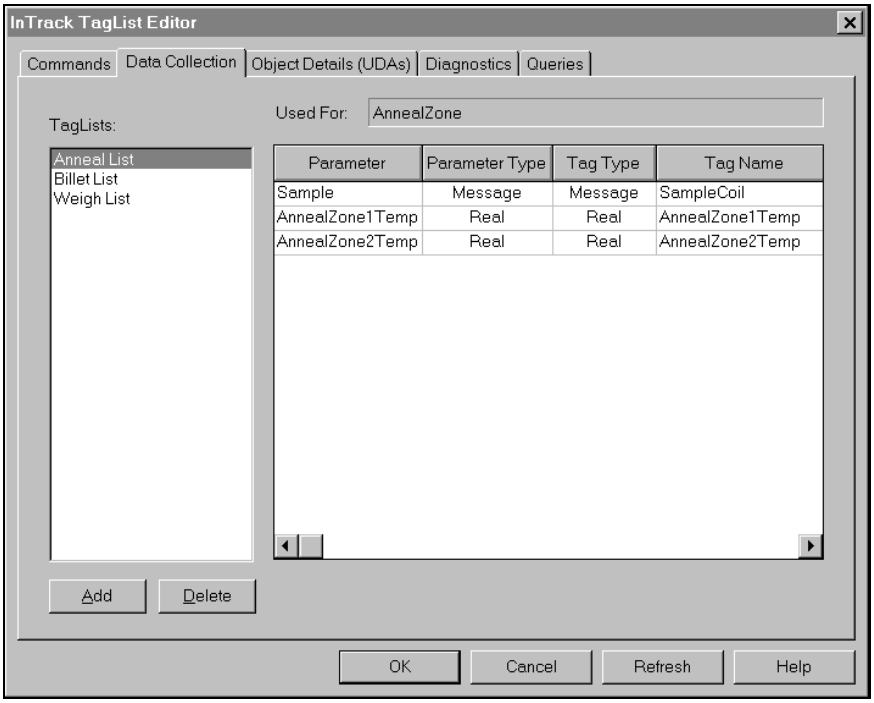
Linking Tags to TagList Parameters

Tags are linked to TagList parameters using the TagList Editor. When an InTrack System command or OLE object method is processed at runtime, the parameter values for the command or method are read from the tags to which the parameters are linked.

Some TagList parameters have default values. If parameter has a default value, and a tag is not linked to it, the default value will be used when processing the command. To use a value other than the default, a tag (memory or I/O) must be assigned to that parameter.

Important I/O tags used in a TagList must be active before an InTrack System command or OLE object method using them can be issued. A tag is active if it is: (1) displayed in a WindowViewer window, (2) used in a script, (3) being alarmed or event logged, (4) displayed in a real time trend or historically logged, (5) the I/O topic is marked as Advise All.

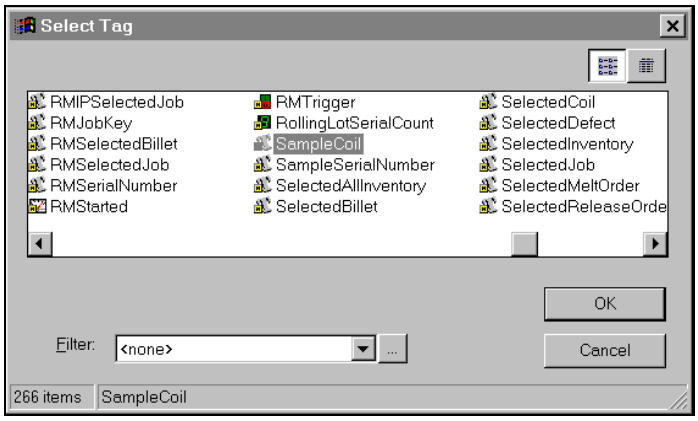
To access the parameters for a TagList, click on the name of the desired TagList in the InTrack TagList Editor:



You can either link an existing tag to a TagList parameter or define a new tag and link it to a TagList parameter.

➢ **To link an existing tag to a TagList parameter:**

1. In the Tag Name column of the TagList Editor, double-click the tagname field of the parameter for which you want to create the link. The Choose Name dialog box appears:

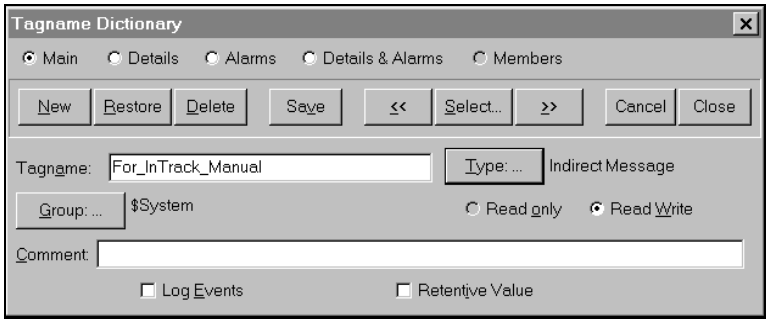


2. Double-click the desired tagname. The Choose Name dialog box will close and the selected tagname will appear in the TagList Editor.

   After a tag has been linked, you can view the tag's definition by double-clicking on the tagname in the list.

➢ **To define a new tag and link it to a TagList parameter:**

1. In the Tag Name column of the TagList Editor, click the tagname field of the parameter for which you want to create the new tag.
2. Type in the name of the new tagname and press Enter. The Dictionary - Tagname Definition dialog box appears:



3. Define the tag. When you are finished, click Done. The Dictionary - Tagname Definition dialog box will close and the defined tag will appear in the TagList Editor.

For more information on the Dictionary - Tagname Definition dialog box, see the *InTouch User's Guide*.

Changing a Tagname

➢ **To change a tagname:**

1. In the Tag Name column of the TagList Editor, click on the field containing the tag to change.
2. Press F2 to highlight the tagname.

– or –

Press the BACKSPACE key on the keyboard.

3. Make the necessary changes.
4. Press ENTER. If the tagname is not currently defined, you will be prompted to define a new tag for that tagname.

Deleting a Tagname

Deleting a tagname from a TagList does not delete it from the InTouch Tagname Dictionary.

➢ **To delete a tagname:**

1. In the Tag Name column of the TagList Editor, click on the field containing the tag to delete.
2. Press F2 to highlight the tagname and press the DELETE key.
3. Press ENTER to finalize the deletion. The cursor will return to a pointer and the field will be selected.

C H A P T E R 4

Selector and Button Wizards

This chapter describes the legacy WindowMaker tools (selector and button wizards) for creating the runtime view into the model. ActiveX controls have replaced the wizards in WindowMaker.

Note: Wizards have been included in this version of Wonderware FactorySuite for commonality with legacy applications only. ActiveX controls be should be used for new applications.

Contents

Development Tools for the Graphical User Interface

After the manufacturing model is defined, WindowMaker provides all the tools necessary for creating a runtime graphical user interface (GUI). The GUI allows operators to control the manufacturing model and to view and track the model's database information. The GUI development tools include InTrack wizards. There are two types of wizards: selector wizards and button wizards. Selector wizards are display objects for runtime, and button wizards are control objects.

All InTrack button wizards and their associated OLE scripts are grouped together based on functionality. There are five categories:

- System
- Work In Process (WIP)
- Inventory Lot
- Bulk Inventory
- Machine

With the exception of the system wizards, related wizards display a common symbol representing their functional category:

System wizards are identified by various symbols, including a computer monitor.

WIP lot wizards are identified by a lot on a conveyor belt.

Inventory lot wizards are identified by stacked boxes.

Bulk inventory wizards are identified by two barrels.

Machine wizards are identified by a machine.

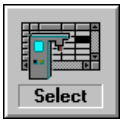
InTrack Selector Wizards

Selector wizards are the runtime display objects into InTrack database tables. There are four selector wizards: the WIP Selector, the Inventory Lot Selector, the Bulk Inventory Selector, and the Machine Selector. In WindowMaker, selector wizards are available from the Wizard Selection dialog box for the appropriate wizard category and display the graphic common to that category:

WIP

Inventory Lot

Bulk Inventory



Machine

After an AttachTo method has been used to associate a Selector object with a selector wizard, selector wizards automatically connect to the database at runtime when the operator successfully logs on. After the operator has logged on, a selector wizard lists all the current information from the database that is applicable to that selector. This information is displayed in a scrollable, spreadsheet type object where each row is a record and each column is a field. The operator then uses the selector to view the current status of WIP lots, inventory lots, bulk inventory, or machines. Operators can select which WIP lot, inventory lot, bulk inventory or machine item (record or row) to work on by clicking anywhere on the desired row.

Note: When WindowViewer is first started, all selector wizards will be blank until the operator logs on to the database using InTrack OLE System Connect script or the Connect button wizard.

WIP Selector

		Lot ID	Operation	Operation Ver	Work State	Priority	Due Date	Material	Material Ver	Queued Qty	Started Qty
1		CWRA-0411-001	Label	NONE	QUEUED	0	12/21/96	RaspberryFlav	1	15.0000	0.0000
2		CW-0418-002	Verify	NONE	QUEUED	0	12/21/96	OrangeFlavor	2	25.0000	0.0000
3		CWNF-0411-001	Label	NONE	IN_PROCESS	0	12/21/96	WonderWate	1	0.0000	25.0000
4		CWNF-0411-002	Package	NONE	IN_PROCESS	0	12/21/96	WonderWate	1	0.0000	15.0000
5		CWNF-0411-003	Cap	NONE	QUEUED	0	12/21/96	WonderWate	1	2.0000	0.0000
6		CWOR-0418-001	Fill	NONE	IN_PROCESS	0	12/21/96	OrangeFlavor	2	0.0000	1.0000
7		CW-0418-001	Verify	NONE	QUEUED	0	12/21/96	OrangeFlavor	2	20.0000	0.0000
8		CWRA-0411-002	Fill	NONE	QUEUED	0	12/21/96	RaspberryFla	2	45.0000	0.0000
9		CWRA-0411-003	Fill	NONE	QUEUED	0	12/21/96	RaspberryFla	2	50.0000	0.0000
10		CWRA-0411-004	Fill	NONE	IN_PROCESS	0	12/21/96	RaspberryFla	1	10.0000	10.0000
11		PCNF-0411	Label	NONE	QUEUED	0	12/21/96	WonderWate	1	15.0000	0.0000
12		PCNF-0411-001	Label	NONE	IN_PROCESS	0	12/21/96	WonderWate	1	0.0000	50.0000
13		PCOR-0411-002	Package	NONE	IN_PROCESS	0	12/21/96	OrangeFlavor	2	0.0000	15.0000

Each selector wizard has specific InTrack button wizards and OLE scripts that support that a selector wizard. All button wizards and OLE scripts, when invoked, are processed on the currently selected item in the applicable selector.

The optional Sort property of a Selector object is used to order the lots in the selector wizard. If you do not set the optional Sort property, the records in the selector wizard are initially displayed in order based on the Due Date (WIP Selector), Lot ID (Inventory Lot Selector), location (Bulk Inventory Selector), or machine name (Machine Selector). The first field in all selector wizards is reserved for color status.

Note: After it is created, the order of the lots or machines in a selector cannot be changed by the user at runtime unless the "Allow user runtime sorting" check box is selected in the appropriate InTrack Grid Configuration dialog box. If this is done, the selector default settings and the optional Sort property of a Selector object are overridden and the user can alphabetically resort the selector by any column simply by double-clicking the selected column heading.

Runtime Features of Selector Wizards

All selector wizards support the following runtime features:

- **Data Awareness**. Selector wizards do not need to be configured for connection to the database. At runtime, a selector wizard automatically displays database information for the InTrack database to which it is connected.
- **Sorting by Fields**. Double-click any field (column) header to sort alphabetically based on the selected field header.

Note: After it is created, the order of the lots or machines in a selector cannot be changed by the user at runtime unless the "Allow user runtime sorting" check box is selected in the appropriate InTrack Grid Configuration dialog box. If this is done, the selector default settings and the optional Sort property of a Selector object are overridden and the user can alphabetically resort the selector by any column simply by double-clicking the selected column heading.

- **Forced Refresh**. Double-click the small button in the top left corner of the selector wizard to force the selector wizard to update with current data.
- **Auto-size**. Position the cursor on the line separating two field headers and, when the cursor changes to two vertical parallel lines, double-click to auto-size the field widths.

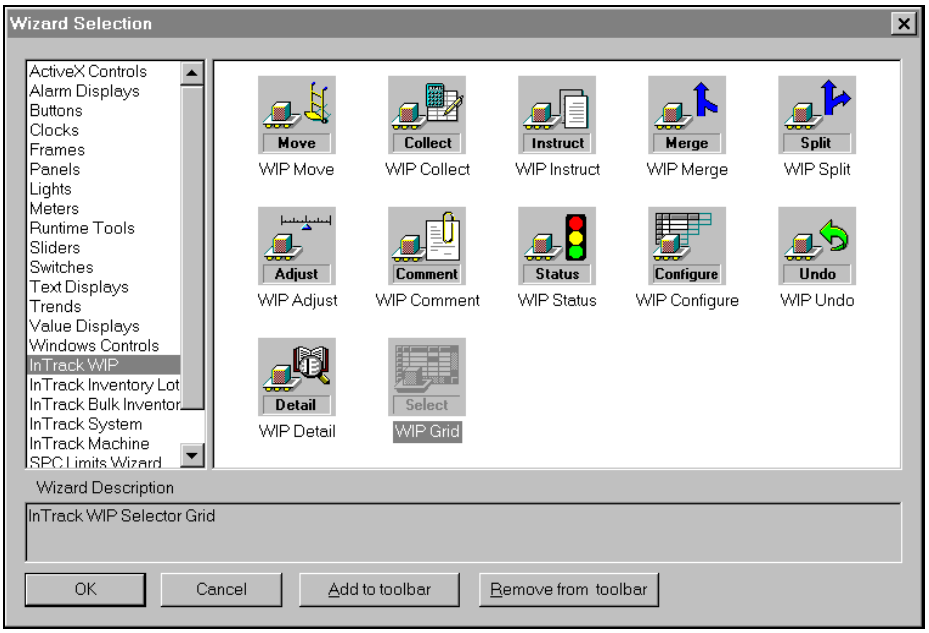
Note: Field (column) widths are persistent. After the system is restarted or the screen recalled, the widths appear as last modified.

- **Scrolling**. Selectors can be scrolled horizontally and vertically. Horizontal scrolling maintains the position of the first two headers, so that color status and lot ID (WIP Selector and Inventory Lot Selector), location (Bulk Inventory Selector), or machine name (Machine Selector) are always visible.
- **WIP Lot Details**. Only supported by the WIP Selector. Double-click anywhere on the row containing the WIP lot. A dialog box appears displaying information pertinent to that record.

Placing a Selector Wizard in a Window

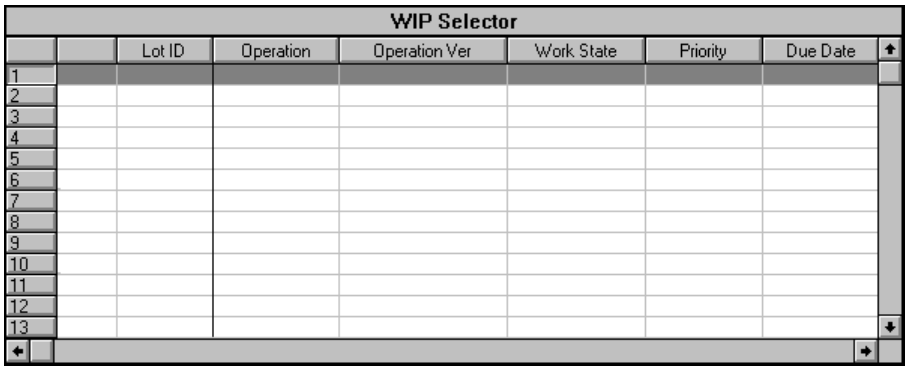
➢ **To place a selector wizard in a window, perform these steps:**

1. Start WindowMaker.
2. Choose the File/New Window... command to open a new window.
3. Click the  tool in the toolbox to access the Wizard Selection dialog box.
4. Click the InTrack category for the desired selector. For example, to place the WIP Selector Grid in the window, click the InTrackWIP category. All of the WIP wizards appear in the Wizard Selection dialog box:



5. Double-click the WIP Selector Grid wizard (Highlighted in example). The Wizard Selection dialog box will close. The WindowMaker window reappears with the cursor in the paste mode.

6. Click in the window to paste the selector wizard. The selector resembles a scrollable spreadsheet grid:



The selector is moveable, sizable, and configurable. All data displayed in the selector come from the database at runtime. Multiple selectors of the same type can be placed on the same screen.

Note: Each selector has specific InTrack button wizards and OLE scripts that create dialog boxes that support the selector. All button wizards and OLE scripts process the currently selected record (row) in the active selector. To select an item or record to work on, click anywhere on the row containing the record. A record is selected when it is displayed in reverse video.

---

7. To configure the selector, double-click it. The grid configuration dialog box for that selector appears.

Configuring a Selector

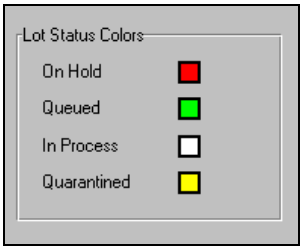
A grid configuration dialog box exists for each of the four types of selectors. This dialog box is used to configure the colors for lot, inventory, or machine status, the selector title, the selector name, and the column heading names. It also provides for user sorting of machines and lots during runtime. When you double-click a WIP Selector, the InTrack WIP Grid Configuration dialog box appears:

InTrack WIP Grid Configuration Dialog Box

The default selector title (WIP Selector, Inventory Lot Selector, Bulk Inventory Selector, or Machine Selector) appears in the **Title** field. To change the selector title displayed in the title bar, enter a new title in the field.

The default selector name appears in the **Name** field (A WIP Selector, An Inventory Lot Selector, An Inventory Bulk Selector, or Machine Selector). The selector name binds a Selector object to a specific selector wizard. To attach a Selector object to a selector wizard, the selector name in this field must be the same as the selector name used in the selectorName parameter of the Selector object' AttachTo method. To change the selector name in this field, enter a new name.

Select **Allow user runtime sorting** to let the user double-click a column heading in the selector wizard at runtime and alphabetically reorder the lots or machines by the selected column.



To change a status color, click its adjacent colored square. The color palette appears:

Click the desired color. The palette closes and the selected color now appears in the square in the InTrack WIP Grid Configuration dialog box. (To close the palette without changing the color, either click the same color or click the configuration dialog box behind the palette.)

For a WIP Selector, status color can be configured for WIP lots:

- On Hold
- Queued
- In Process
- Quarantined

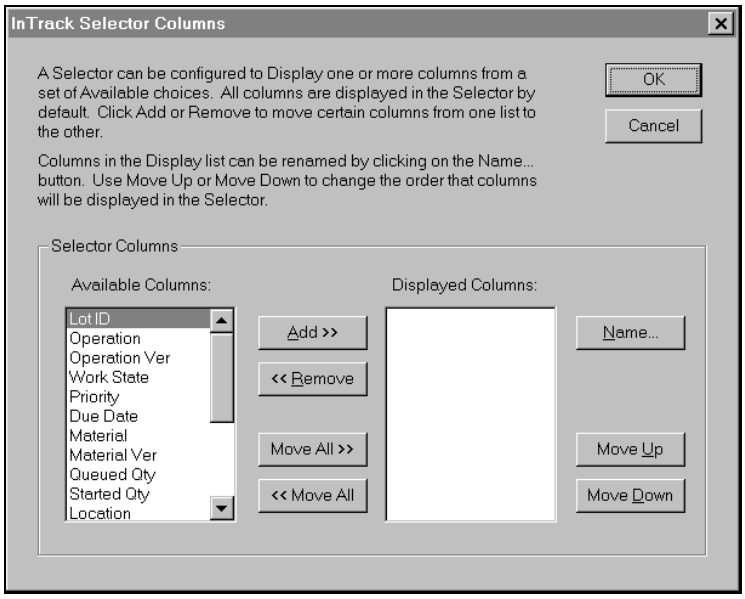
For an Inventory Lot Selector or a Bulk Inventory Selector, status color can be configured for inventory:

- Normal
- On Hold
- Quarantined

For a Machine Selector, status color can be configured for machine status:

- In Use
- Idle
- Offline

To configure column headings for the selector, click the **Columns** button. The InTrack Selector Columns dialog box appears:



InTrack Selector Columns Dialog Box

The **Available Columns** window lists all default column headers (field headers) available for the selector. Use the scroll bar to view all of the column header names.

The **Displayed Columns** window lists all column headers to be displayed in the selector. The order that the column headers appear vertically (from top to bottom) in the Displayed Columns window is the order that they will appear horizontally (from left to right) in the selector.

To add an available column header to the selector, click the desired column header in the Available Columns window, and click the **Add** button. This button will be disabled (grayed) if no column header is selected in the Available Columns window. The column header name will move to the Displayed Columns window and be listed in the selector.

To remove a displayed column header in the selector, click the desired column header in the Displayed Columns window, and click the **Remove** button. This button will be disabled (grayed) if no column header is selected in the Displayed Columns window. The column header name will move to the Available Columns window and be removed from the selector.

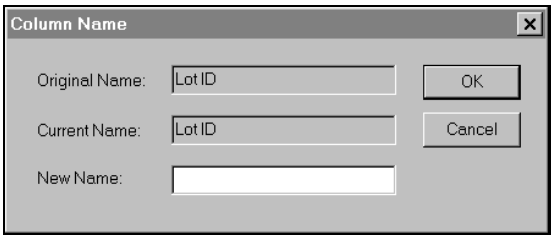
To include all available column headers in the selector, click the **Move All** >> button.

To remove all column headers from the selector, click the << **Move All** button.

Note: Removing all available column headers will default to all the columns being displayed in the selector. The color status column (field) cannot be removed from the selector.

To move a column header down a row in the Displayed Columns window, select the desired column header and click the **Move Down** button.

To change the name of a column header, select the desired column header in the Displayed Columns window and click the **Name** button. The Column Name dialog box appears:



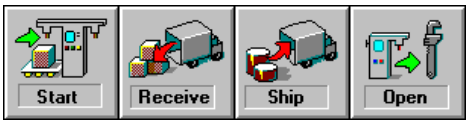
Column Name Dialog Box

The original name of the column header is displayed in the **Original Name** field.

The current name for the column header is displayed in the **Current Name** field.

Enter the new name for the column header in the **New Name** field.

InTrack Button Wizards



Button wizards are graphical control objects for InTrack database information. They are assigned OLE scripts that control operations at runtime. There is an equivalent OLE script for each button wizard. Button wizards provide ease of use since they contain all intelligence required to issue their assigned OLE script (start a WIP lot, receive inventory lots, ship bulk inventory, open a machine repair). The OLE script assigned to the button wizard operates on the currently selected record (row) in the respective selector wizard. For example, clicking on the WIP Start button wizard invokes the OLE script to start the WIP lot currently selected in the WIP Selector.

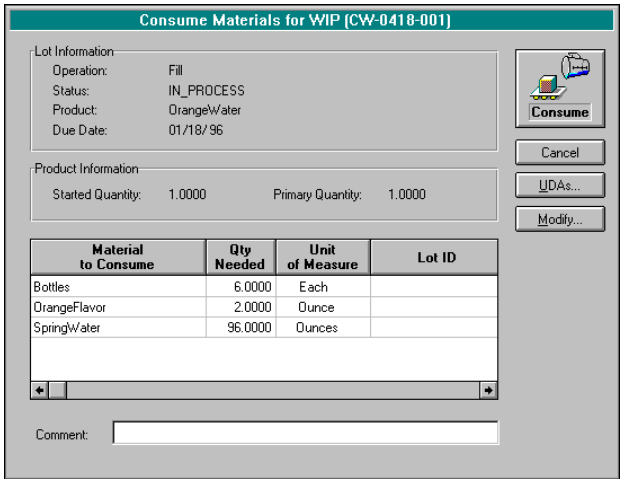
Note: InTrack commands are supported for backward compatibility with InTrack for the Dialog Mode, TagList Mode, and the Results Modifier.

Button wizards function according to the manufacturing model defined in InTrack ModelMaker. InTrack will not let a button wizard process an OLE script that was not defined for the currently selected record in the selector wizard. For example, if the Complete button wizard is clicked before the Start button wizard, and the operation was defined as Start and Complete, InTrack will not let the OLE Complete script be processed. (A message appears indicating the error.)

When invoked, all button wizard and OLE scripts provide you with a dialog box for entering required information.

Note: An OLE script also can be processed without using the button wizard (using an application script, key script, etc.). When invoked, scripts will display the same dialog box as the equivalent button wizard.

For example, if a WIP lot has been started at an operation that requires consumption, and the Consume button wizard is clicked, the Consume Materials for WIP dialog box appears:



You are responsible for entering all pertinent information in the dialog box. When the Consume button (in this example) is pressed, the entered information is written to the

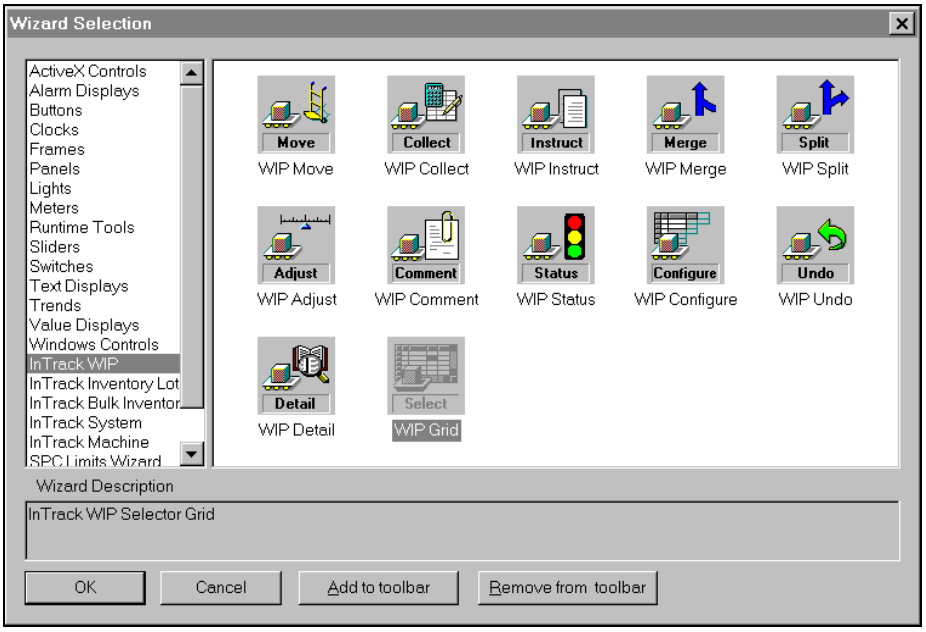
database, and all related tables are updated. In this example, the inventory for bottles, orange flavor, and spring water would be updated to show the consumption. (NewQuantity = CurrentQuantity - ConsumedQuantity). If consumption is not required or has already been completed, a message appears informing you.

Note: Dialog boxes appearing at runtime do not halt background tasks. A dialog box can remain on the screen indefinitely without affecting WindowViewer activities.

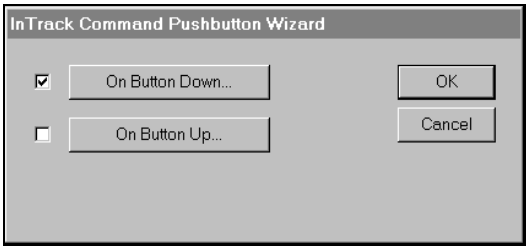
Placing a Button Wizard in a Window

➢ **To place a button wizard in a window, perform these steps:**

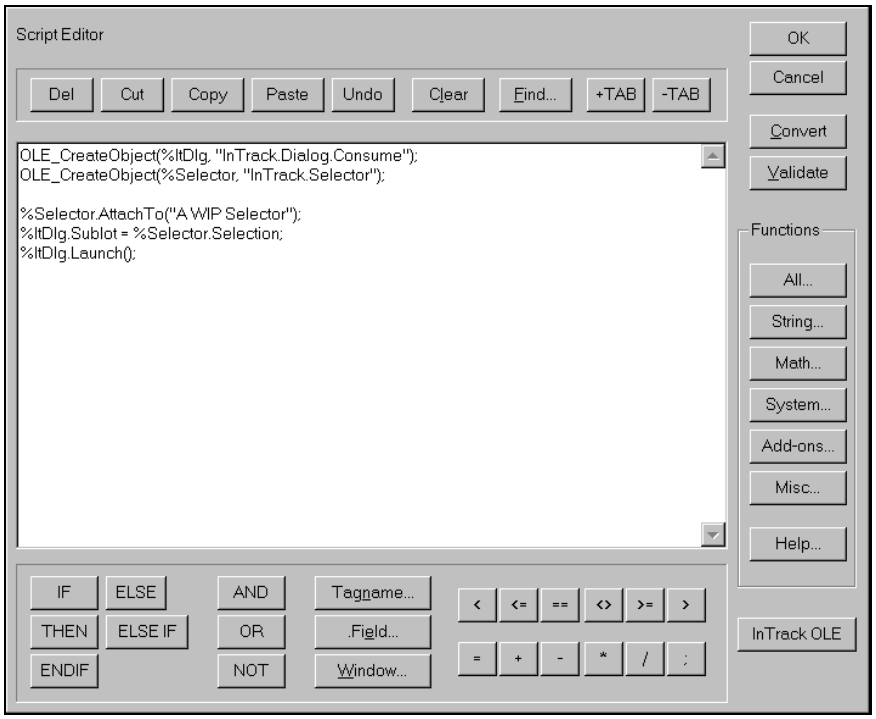
1. Start WindowMaker.
2. Choose the File/New Window... command to open a new window.
3. Click the  tool in the toolbox to access the Wizard Selection dialog box.
4. Click the InTrack category for the desired selector. For example, to place a WIP lot button wizard in the window, click the InTrackWIP category. All of the WIP wizards appear in the Wizard Selection dialog box:



5. Double-click the button wizard (outlined in example). The Wizard Selection dialog box will close. The WindowMaker window reappears with the cursor in the paste mode.
6. Click in the window to paste the button wizard.
7. To view the OLE script function assigned to a button wizard, double-click it. The InTrack Command Pushbutton Wizard dialog box appears:



8. By default, the On Button Down... button will be selectd. Click the button to view the OLE script. The Script Editor for that button wizard appears:



9. Additional script can be added to any button wizard OLE script. Type in the desired function or select a function by clicking on any of the Functions buttons.

Button Wizard OLE Scripts

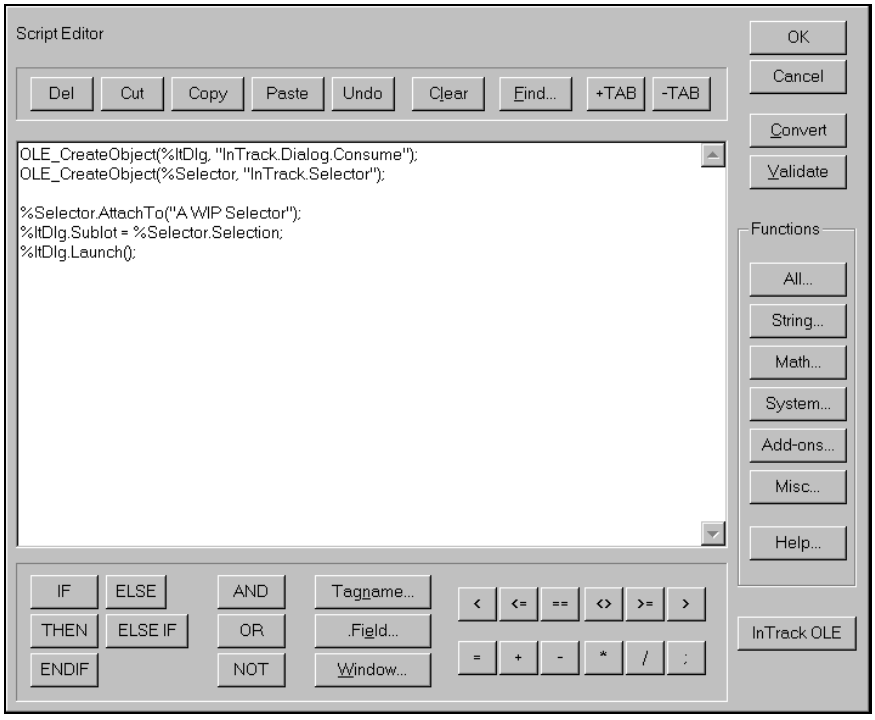
For each InTrack button wizard, there is an equivalent OLE script that performs a specific action. For example:


```
= OLE_CreateObject(%ItDlg, "InTrack.Dialog.Connect");
 %ItDlg.Launch();
```

To perform a specific action, the operator can use the corresponding button wizard or the button wizard OLE script in a script, (application script). A button wizard requires an operator to click it to perform an action, while a wizard OLE script can be issued automatically.

When a wizard OLE script is issued at runtime, a corresponding dialog box appears. The same dialog box displays for a button wizard or its corresponding OLE script. Dialog boxes appearing at runtime do not halt background tasks. A dialog box can remain on the screen indefinitely without affecting WindowViewer activities.

The procedure issue OLE scripts at runtime is dependent upon how the developer designed the application. For example, InTrack can be connected to the database manually using the Connect button wizard at runtime, or an application script can be created to automatically connect InTrack to the database at startup:



All InTrack button wizards and OLE scripts are grouped into categories based on functionality:

- System
- Work In Process (WIP)
- Inventory Lot
- Bulk Inventory
- Machine

Each category performs several actions. For example, the System category contains a button wizard (and corresponding OLE script) to connect to the database and another to disconnect from the database. The action is completed after an operator enters all required information in the dialog box. All OLE scripts are logged to the database.

InTrack System Button Wizards

Several button wizards are associated with system tasks. At runtime, these button wizards process system OLE scripts. No configuration is required for button wizards. Button wizards function according to the manufacturing model defined in InTrack ModelMaker. InTrack will not let a button wizard process an OLE script that was not defined. System button wizards are identified by various symbols, including a computer monitor, and use the OLE script:

```
OLE_CreateObject(%ItDlg, "InTrack.ObjectClass");
%ItDlg.DoneFlagName = "DoneFlag";
%ItDlg.Launch();
```

The following list describes the system button wizards:

Button Wizard	Used To
Connect	Connect to and log on the database
Disconnect	Disconnect from and log off the database
Logoff	Log off runtime
Logon	Log on runtime
Help	Access the runtime online Help file
Print	Print a database report using Crystal Reports
Purge	Purge the database of unnecessary information

System Connect

This button wizard establishes a connection to the database and logs on the specified user.

To connect to the database at runtime, click this button wizard. The InTrack Logon dialog box appears.

System Disconnect

This button wizard logs off the logged-on user and terminates the database connection.

To disconnect from the database at runtime, click this button wizard.

System Help

This button wizard accesses the online Help file for InTrack runtime. To point to a different Help file, modify the path and the name of the Help file. For example:

```
StartApp "WINHELP.EXE C:\INTOUCH.32\MYHELP.HLP";
```

will execute the Help file **myhelp.hlp**, located in the INTOUCH.32 directory.

To access online Help at runtime, click this button wizard.

System Logoff

This button wizard logs the logged-on user off the runtime application, but maintains the database connection.

To log off the runtime application at runtime, click this button wizard.

System Logon

This button wizard logs a user on to the runtime application. If a user is already logged on when a new user logs on, the previous user is automatically logged off.

To log on the runtime application at runtime, click this button wizard. The InTrack Logon dialog box appears.

System Print 

This button wizard is equivalent to the script:

```
WWReport("Print", ReportFileName, "NONE");
```

where:

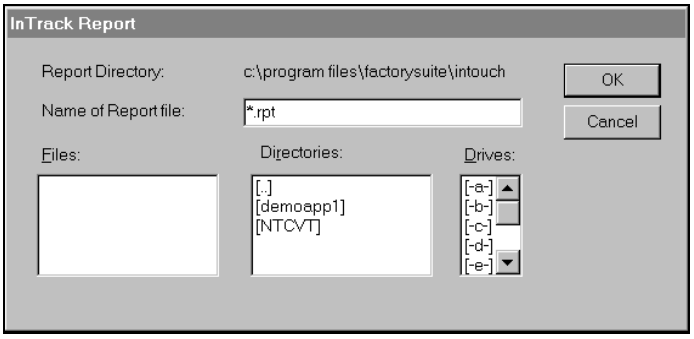
Parameter	Description
Print	Command to execute
ReportFileName	Name of the actual report file (including the path) to be printed
NONE	Placeholder for currently unused parameter

This button wizard prints a Crystal Reports report containing database information formatted using the included Crystal Reports software. Reports can be generated by modifying the sample report files provided with InTrack, or custom report files can be created. Before a database report can be printed at runtime, a report file must be associated with the Print button wizard. One or more report files can be associated with a single Print button wizard by duplicating the Print script for each report file name.

Associating a Report File

➢ **To associate a file with the Print button wizard, perform these steps:**

1. Place the Print button wizard in a WindowMaker window.
2. Double-click the Print button wizard. The InTrack Report dialog box appears:



InTrack Report Dialog Box

The Report Directory is the path to the current directory.

Enter the name of the report file to print when the Print button wizard is pressed. To browse for a report file, use the Files:, Directories:, and Drives: list boxes.

To select a report file, scroll to the desired file and double-click the file name.

To select a different directory, scroll to the desired directory and double-click the directory name. Double-clicking the [...] symbol will cause the directories in the next highest level of the directory tree to be displayed.

To select a different drive, scroll to the desired drive and double-click the drive letter.

Data Refreshes

The Crystal Reports reporting component contains a menu option, Save Data with Report, that allows only saved data to be included in the report printout. If the Save Data with Report option is selected, any new data added to the report after the report was last saved will not be included in the printout. Disabling this option will allow all data (saved and new) to be included in the printout generated by the Print command script.

➢ **To disable the Save Data with Report option for a Crystal Reports file:**

1. Open the file for which you want to disable the Save Data with Report option.
2. Choose the File/Save Data with Report command and toggle the option off (unchecked).

Note: The Save Data with Report option is not global for all Crystal Report files; you must disable the option for each file.

ReportTransCount

InTrack keeps a record of the total number of report printouts that have been generated and writes this total to the tag "ReportTransCount." Each time a database report is printed, value for ReportTransCount is incremented by one. The ReportTransCount tag is listed in the WWREPORT.INI file located in the application directory. For example:

```
[General];

MessageLevel=1;

TransCountTag=ReportTransCount;
```

To display the ReportTransCount in WindowViewer (to verify that the report printed successfully), create an animation link for the ReportTransCount tag.

➢ **To change the tag used for the ReportTransCount:**

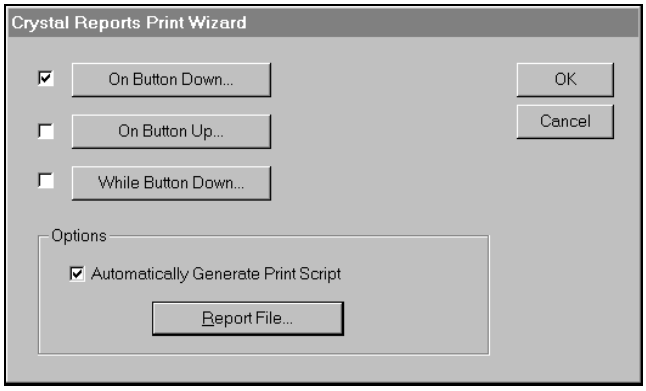
1. Shutdown WindowMaker and WindowViewer.
2. Open the WWREPORT.INI file and change the setting for the TransCountTag (TransCountTag=MyReportCount). Save the file.
3. Restart WindowMaker and WindowViewer.

Editing or Creating a Print Button Wizard Script

When a report file is associated with a Print button wizard, the command script for printing that report file is automatically generated.

➢ **To edit the default Print command script or create additional scripts for a Print button wizard, perform these steps:**

1. Start WindowMaker.
2. Choose the File/Open Window... command to open the window containing the Print button wizard.
3. Double-click the Print button wizard. The Crystal Reports Print Wizard dialog box appears:



Crystal Reports Print Wizard Dialog Box

Note: A report file must be associated with the Print button wizard for the Crystal Reports Print Wizard dialog box to appear.

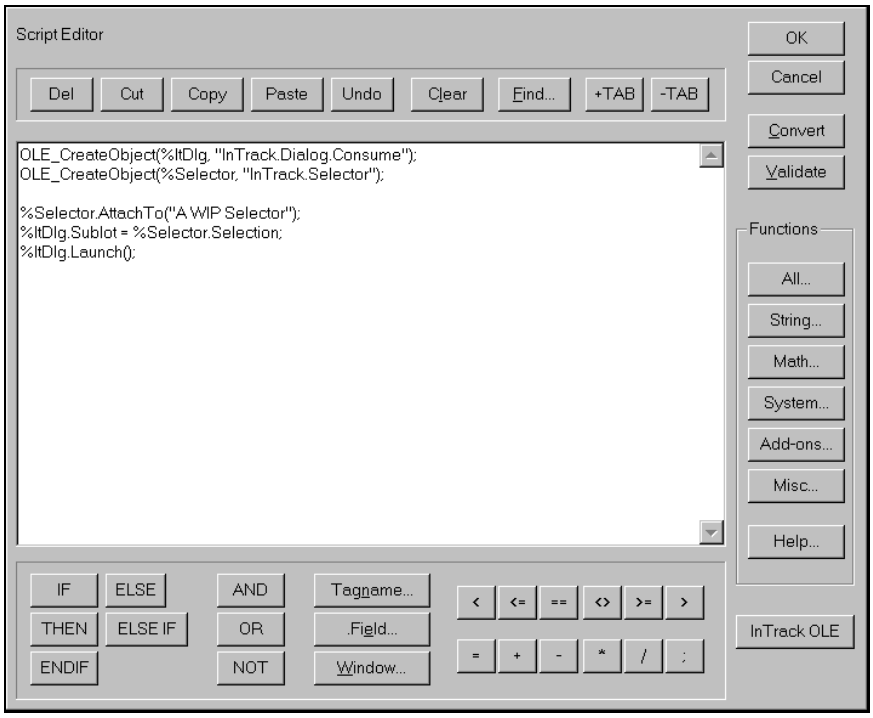
Select Automatically Generate Print Script to automatically create the WWReport command script based on a selected report.

Click the Report File button to access the InTrack Report dialog box.

Select the On Button Down option to process a command script when the Print button wizard clicks down.

Note: On Button Down, On Button Up, and While Button Down are selected only if the Automatically Generate Print Script option is not selected.

To edit the script, click the On Button Down... button. The Script Editor appears:



Note: To print more than one report file, duplicate the command script and substitute the second report file name.

Select the On Button Up option to process a command script when the Print button wizard is released. To edit or create the script, click the On Button Up... button. The Script Editor appears.

Select the While Button Down option to repeatedly process a command script while the Print button wizard button is pressed down. Enter a time interval for processing the script. To edit or create the script, click the While Button Down... button. The Script Editor appears.

Note: This option is not recommended for printing a report file.

To accept the input and close the dialog box, click OK.

To close the dialog box and cancel the input, click Cancel.

System Purge 

This button wizard deletes from the database all WIP lots, inventory lots, and bulk inventory that have both of the following characteristics:

- A quantity of zero
- A date past a specified number of days from the last activity date

To purge the database at runtime, click this button wizard. The Database Purge dialog box appears.

Work In Process (WIP) Lot Wizards

A WIP lot is a collection of the same material types processed and tracked together through the manufacturing process. The display object for viewing WIP lots on a manufacturing route is the WIP Selector wizard. Runtime control of WIP lots is accomplished using dialog boxes created with either button wizards or OLE scripts. WIP lot button wizards are identified by a lot on a conveyor belt and use the OLE script:

```
OLE_CreateObject(%ItDlg, "InTrack.ObjectClass");
OLE_CreateObject(%Selector, "InTrack.Selector");

%Selector.AttachTo("A WIP Selector");
%ItDlg.Sublot = %Selector.Selection;
%ItDlg.DoneFlagName = "DoneFlag";
%ItDlg.Launch();
```

WIP Selector Wizard

The WIP Selector wizard is a display object used to view the current status of WIP lots and to select WIP lots for processing. The WIP Selector displays the current database information associated with all WIP lots. This information is displayed in a scrollable, spreadsheet type object, where each row is a record and each column is a field. The record (row) represents an actual WIP lot, and each field (column) displays associated information for the WIP lot.

Note: When WindowViewer is first started, all selector wizards will be blank until the operator logs on to the database using the OLE System Connect script or the Connect button wizard.

After a Selector object is created by clicking the button wizard or generating an OLE script, the OLE AttachTo method associates it with a specific WIP Selector wizard. This gives you access to all the WIP lots within the WIP Selector. To select a WIP lot on which to work, click anywhere on the row containing the WIP lot. The selected row will be highlighted.

WIP Selector

		Lot ID	Operation	Operation Ver	Work State	Priority	Due Date	Material	Material Ver	Queued Qty	Started Qty
1		CWRA-0411-001	Label	NONE	QUEUED	0	12/21/96	RaspberryFlav	1	15.0000	0.0000
2		CW-0418-002	Verify	NONE	QUEUED	0	12/21/96	OrangeFlavor	2	25.0000	0.0000
3		CWNF-0411-001	Label	NONE	IN_PROCESS	0	12/21/96	WonderWate	1	0.0000	25.0000
4		CWNF-0411-002	Package	NONE	IN_PROCESS	0	12/21/96	WonderWate	1	0.0000	15.0000
5		CWNF-0411-003	Cap	NONE	QUEUED	0	12/21/96	WonderWate	1	2.0000	0.0000
6		CWOR--0418-001	Fill	NONE	IN_PROCESS	0	12/21/96	OrangeFlavor	2	0.0000	1.0000
7		CW-0418-001	Verify	NONE	QUEUED	0	12/21/96	OrangeFlavor	2	20.0000	0.0000
8		CWRA-0411-002	Fill	NONE	QUEUED	0	12/21/96	RaspberryFlar	2	45.0000	0.0000
9		CWRA-0411-003	Fill	NONE	QUEUED	0	12/21/96	RaspberryFlar	2	50.0000	0.0000
10		CWRA-0411-004	Fill	NONE	IN_PROCESS	0	12/21/96	RaspberryFlar	1	10.0000	10.0000
11		PCNF-0411	Label	NONE	QUEUED	0	12/21/96	WonderWate	1	15.0000	0.0000
12		PCNF-0411-001	Label	NONE	IN_PROCESS	0	12/21/96	WonderWate	1	0.0000	50.0000
13		PCOR-0411-002	Package	NONE	IN_PROCESS	0	12/21/96	OrangeFlavor	2	0.0000	15.0000

Unless otherwise specified with the optional Sort property of a Selector object, WIP lots in the WIP Selector are displayed in alphabetical order based on the Due Date (the seventh field in the WIP Selector). The first field is reserved for color status.

Note: After it is created, the order of the lots in a WIP Selector cannot be changed by the user at runtime unless the "Allow user runtime sorting" checkbox is selected in the InTrack WIP Grid Configuration dialog box. If this is done, the WIP Selector default settings and the optional Sort property of a Selector object are overridden and the user can alphabetically resort a WIP Selector by any column simply by double-clicking the selected column heading.

By default, the WIP Selector displays the following information for WIP lots in the fields (columns) at runtime:

Field (column) Name	Description
Lot ID	WIP lot identifier
Operation	Current WIP lot operation
Operation Ver	Current operation version
Work State	Current status for the WIP lot on the route, queued, in process, completed
Priority	Priority of the WIP lot
Due Date	Date the WIP lot is scheduled to be closed into inventory
Material	Material consumed as well as material and products created during manufacturing
Material Ver	Material version
Queued Qty	WIP lot quantity currently queued for an operation
Started Qty	WIP lot quantity actually started for an operation
Location	Current WIP lot location
Route	Current route to which the WIP lot is assigned
Route Ver	Current route version
Step	Route step in which the WIP lot is currently processing
Customer Spec	Customer specification name
Customer Ver	Customer specification version
Hold Status	Indicates whether the WIP lot is on hold or released
Quar Status	Indicates whether the WIP lot is quarantined or active

WIP Lot Button Wizards

A WIP lot button wizard processes an OLE script for the currently selected WIP lot in the WIP Selector. No configuration is required for button wizards. Button wizards function according to the manufacturing model as defined in InTrack ModelMaker. InTrack will not let a button wizard process an OLE script that was not defined for the currently selected WIP lot in the WIP Selector. For example, if the Complete button wizard is clicked before the Start button wizard, and the operation was defined as Start and Complete, InTrack will not allow the OLE Complete script to be processed. (A message appears indicating the error.)

Note: All button wizards provide the operator with a dialog box for entering required information.

The following list describes the WIP button wizards:

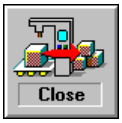
Button Wizard	Used To
Adjust	Adjust WIP lot quantities
Close	Close a portion (or all) of the WIP lot to inventory
Collect	Gather quantitative and qualitative data about the WIP lot's manufacturing process
Comment	Log a comment for a WIP lot
Complete	Complete a portion (or all) of the WIP lot at one operation and queue it at the next operation. Portions of the WIP lot may also go to inventory, another route, or to scrap.
Configure	Show WIP lots in the WIP Selector according to specified parameters (show only queued WIP lots)
Consume	Select the location and quantity of input materials to be consumed for the WIP lot. Only applicable if the WIP lot is at an operation that has input materials assigned to it in InTrack ModelMaker.
Create	Create a new WIP lot and queue it at a selected step of a selected route
Detail	Show detailed information about a selected WIP lot
Instruct	Display the work instructions for a WIP lot at a certain route step
Merge	Merge WIP lots into the selected WIP lot
Move	Move a WIP lot to another step and/or location on the same route or on another route
Split	Split a WIP lot into two separate WIP lots
Start	Start a portion or all of a WIP lot queued at a particular operation
Status	Change a WIP lot's hold or quarantine status
Undo	Undo the last transaction issued for the WIP lot

WIP Lot Adjust 

This button wizard makes corrections and adjustments to WIP lot quantities. When adjusting a WIP lot, the queued and started quantities, the WIP lot priority level, the due date, and the user-defined attribute (UDA) values can be modified.

Note: Adjust is also important if output is defined on a bill of material that decrements the target product. If the standard quantity of a target product is one and the output quantity is also one, use this button wizard to offset the decrement.

To adjust a WIP lot at runtime, select the desired WIP lot in the WIP Selector, then click this button wizard. The Adjust WIP dialog box appears.

WIP Lot Close 

This button wizard moves quantities that have completed the route into inventory. The Close button wizard is used at the last step on the route.

Note: A WIP lot on hold cannot be closed. To move a by-product or coproduct to inventory, use the Complete button wizard.

To close a WIP lot to inventory at runtime, select the desired WIP lot in the WIP Selector, then click this button wizard. The Close WIP dialog box appears.

WIP Lot Collect 

This button wizard lets the operator collect engineering data along a route to provide in process quality checks for tracking, analysis, and reporting. Information collected during runtime is entered using data set templates that were defined in InTrack ModelMaker. At runtime, the stored data is then referred to as a data set. A data set may include multiple samples, and data collection may be a required procedure at an operation. Multiple data set templates can be assigned to an operation.

To collect data at an operation for a WIP lot at runtime, select the desired WIP lot in the WIP Selector, then click this button wizard. The DataSet for WIP dialog box appears.

WIP Lot Comment 

This button wizard logs comments for a particular WIP lot. The comments are stored in the LotBaseLog in InTrack database. The comment file records the operator ID, the time the comments are entered, and the WIP lot ID.

To log comments for a WIP lot at runtime, select the desired WIP lot in the WIP Selector, then click this button wizard. The Comment for WIP dialog box appears.

WIP Lot Complete 

This button wizard completes a portion (or all) of a WIP lot at one operation and moves it to the next operation. Using the Complete button wizard changes a WIP lot's status to "queued" at the next operation (step). (If operations are defined as Complete Only, the status of a WIP lot is always "queued.") This button wizard:

- Completes a partial quantity or the entire WIP lot at the current operation, and queues it for the next operation
- Dispositions WIP lots to rework or reject operations
- Scraps WIP lot quantities
- Closes by-products or coproducts (*output*) to inventory or another route
- Closes an entire WIP lot or partial quantity to inventory or to another route. The Complete button wizard can only be used if there are disposition options other than closing the entire WIP lot to inventory (closing a partial quantity to inventory, closing a WIP lot to another route). If closing to inventory is the only option, use the Close button wizard on 2-**Error! Bookmark not defined.**.

To complete a WIP lot at runtime, select the desired WIP lot in the WIP Selector at runtime, then click this button wizard. The Complete WIP dialog box appears.

WIP Lot Configure 

This button wizard displays WIP lots in a WIP Selector according to specified parameters. For example, a WIP Selector can be configured so that only WIP lots queued or in process at the Cap operation are displayed. Multiple parameters, including the maximum number of rows to display in a specific WIP Selector, also can be specified.

When a WIP Selector is configured, all WIP lots might not be displayed in the selector. However, as long as you provide a fully qualified SublotKey, automatic OLE scripts can still be executed against the configured WIP lots. All WIP lots are still tracked by the system, and current data will be reported when the WIP lots are configured back into the selector.

Note: To let the user double-click a column heading in a WIP Selector at runtime and alphabetically reorder the lots by the selected column, select the "Allow user runtime sorting" check box in the InTrack WIP Grid Configuration dialog box.

To configure a WIP Selector at runtime, click this button wizard. The WIP Configuration dialog box appears.

WIP Lot Consume 

This button wizard controls and tracks the quantities of input material consumed at an operation. Material can be consumed only for a product that is manufactured with a bill of material. Material can be consumed from multiple quantities of an inventory lot or bulk inventory.

To consume material at an operation, the following conditions must exist:

- The operation must have input material assigned to the route step
- The material must already be received into inventory
- The consuming lot must be in process if the operation is Start and Complete or queued if the operation is Complete Only
- The consuming lot must not be on hold

To consume material at an operation at runtime, select the desired WIP lot in the WIP Selector, then click this button wizard. The Consume Materials for WIP dialog box appears.

WIP Lot Create

This button wizard creates a new WIP lot and queues it at the initial step on the manufacturing route. A WIP lot is a collection of units of the same material type processed and tracked together along one or more routes. WIP lots represent the products being manufactured.

Note: It is not necessary to select a WIP lot in the WIP Selector before using this button wizard.

To create a new WIP lot at runtime, click this button wizard. The Create WIP dialog box appears, prompting the operator to enter all required information about the WIP lot being created.

WIP Lot Detail

This button wizard displays a dialog box containing the following information about a selected WIP lot:

- General information about the WIP lot (status, due date)
- Quantity information
- Details about the WIP lot's current operation

Note: The Details for WIP dialog box is read-only.

To view details for a WIP lot at runtime, click the desired WIP lot in the WIP Selector, then click this button wizard. The Details for WIP dialog box appears.

WIP Lot Instruct

This button wizard displays a window containing work instructions that were assigned to an operation in InTrack ModelMaker or the OLE API. Operations may require specific instructions in order to be completed properly. The Work Instruction for Lot window may contain information such as text, graphics, CAD drawings, and spreadsheets.

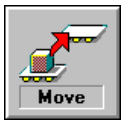
Any work instruction overrides associated with the operation also display in the Work Instruction for Lot window. The window can be sized optimally to display the instructions; it can be minimized to an icon or maximized to the size of the runtime window.

Note: For work instructions stored in the database in the internal format, it is recommended that, as part of the application system test, you verify that the information is displayed in viewable form in the runtime dialog.

To view a set of work instructions for an operation at runtime, select the desired WIP lot in the WIP Selector, then click this button wizard. The Work Instruction for Lot window appears.

Note: A message box notifying you to read the work instructions for an operation will display before the Work Instruction for Lot window as long as the current work instructions fall within the "Days to Notify After Change" value defined in InTrack ModelMaker.

WIP Lot Move



This button wizard moves a WIP lot to a new route step or location (or both).

Note: To perform a WIP move, the WIP lot must be queued at an operation.

All of the following are considered WIP moves - moving some or all of a WIP lot quantity to:

- A specific step on the same route or a different route
- A new location on the same route or a different route
- A specific step and location on the same route or a different route

---

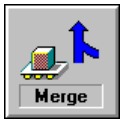
Note: A WIP lot on hold cannot be moved.

To perform a WIP move on a WIP lot at runtime, select the desired WIP lot in the WIP Selector, then click this button wizard. The Move WIP dialog box appears.

WIP Lot Split 

This button wizard divides a WIP lot into two WIP lots. Split lots are tracked independently from each other. To split a WIP lot, it must have multiple quantities and be queued at an operation or closed. Split lots inherit the same priority level and due date of the original WIP lot.

To split a WIP lot at runtime, select the desired WIP lot in the WIP Selector, then click this button wizard. The Split WIP dialog box appears.

WIP Lot Merge 

This button wizard combines two or more WIP lots into a single WIP lot.

Note: Only WIP lots with the same status, queued at the same operation, and producing the same product can be merged. A message box appears if there are no WIP lots available for merging.

To merge one or more WIP lots at runtime, select the desired WIP lot in the WIP Selector, then click this button wizard. The Merge into WIP dialog box appears.

WIP Lot Start 

This button wizard changes a specified amount of the selected WIP lot from "queued" to "in process" at an operation. A WIP lot must be queued at the route step to start the operation.

Note: A WIP lot on hold cannot be started. If an operation is defined as Complete Only, do not use the Start button wizard. A Complete Only operation only tracks lot quantities when they complete an operation, not when they are in process. A warning message will display if you attempt to use a Start button wizard for a Complete Only operation.

To start a WIP lot at runtime, select the desired WIP lot in the WIP Selector, then click this button wizard. The Start WIP dialog box appears.

WIP Lot Status

This button wizard places or releases a WIP lot on hold or in quarantine status. If a WIP lot is on hold, it cannot be processed on a route. (The Close button wizard can be used, provided that the WIP lot is in process at the final route step.) A WIP lot placed in quarantine can be processed on the route, but retains a quarantine status when closed to inventory. When a WIP lot that is on hold or in quarantine is closed to inventory, its status is retained. No other transaction can be performed until the WIP lot is released. If the WIP lot is placed on hold or in quarantine while in process at an operation defined with output (coproduct or by-product), the output also inherits the status of the parent WIP lot.

To change the status for a WIP lot at runtime, select the desired WIP lot in the WIP Selector, then click this button wizard. The Status of WIP dialog box appears.

WIP Lot Undo

This button wizard returns the selected WIP lot to the status it had prior to the most recent transaction. Several consecutive transactions can be undone by repeatedly clicking Undo button wizard. It is not necessary to undo transactions immediately after performing them; as long as the transaction log has not been purged, you can undo any transaction for any WIP lot at any time.

Note: If a Split transaction is performed, and subsequent transactions are performed on the split lots, transactions can only be undone back to the split.

To undo a previous WIP lot transaction at runtime, click the desired WIP lot in the WIP Selector, then click this button wizard. The Undo for WIP dialog box appears.

Inventory Lot Wizards

An inventory lot is a collection of material of the same type stored in inventory and tracked by a lot number. The display object for viewing all inventory lots for a manufacturing route is the Inventory Lot Selector wizard. Runtime control of inventory lots is accomplished using either button wizards or OLE scripts. Inventory lot button wizards are identified by stacked boxes and use the OLE script:

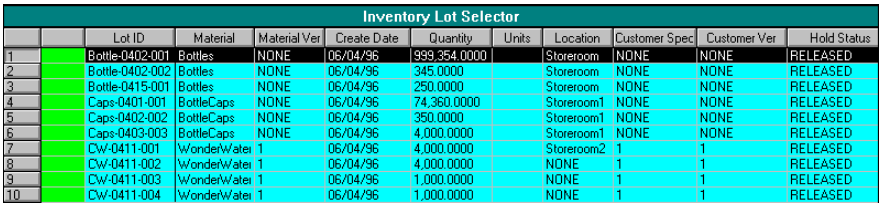
```
OLE_CreateObject(%ItDlg, "InTrack.ObjectClass");
OLE_CreateObject(%Selector, "InTrack.Selector");

%Selector.AttachTo("An Inventory Lot Selector");
%ItDlg.Sublot = %Selector.Selection;
%ItDlg.DoneFlagName = "DoneFlag";
%ItDlg.Launch();
```

Inventory Lot Selector Wizard

The Inventory Lot Selector wizard is used to view all inventory lot quantities available for consumption, and all completed products and output closed to inventory. The Inventory Lot Selector wizard displays the current database information associated with all inventory lot quantities. This information is displayed in a scrollable, spreadsheet type object where each row is a record, and each column is a field. The record (row) represents the actual inventory lots, and each field (column) displays associated information for the inventory lot.

Note: When WindowViewer is first started, all selector wizards will be blank until the operator logs on to the database using the OLE System Connect script or the Connect button wizard.

After a Selector object is created, the AttachTo method associates it with a specific Inventory Lot Selector wizard, which gives you access to all the inventory lots within the Inventory Lot Selector wizard. To select an inventory lot on which to work, click anywhere on the row containing the inventory lot. The selected row will be highlighted.



Inventory Lot Selector

		Lot ID	Material	Material Ver	Create Date	Quantity	Units	Location	Customer Spec	Customer Ver	Hold Status
1		Bottle-0402-001	Bottles	NONE	06/04/96	999,354.0000		Storeroom	NONE	NONE	RELEASED
2		Bottle-0402-002	Bottles	NONE	06/04/96	345.0000		Storeroom	NONE	NONE	RELEASED
3		Bottle-0415-001	Bottles	NONE	06/04/96	250.0000		Storeroom	NONE	NONE	RELEASED
4		Caps-0401-001	BottleCaps	NONE	06/04/96	74,360.0000		Storeroom1	NONE	NONE	RELEASED
5		Caps-0402-002	BottleCaps	NONE	06/04/96	350.0000		Storeroom1	NONE	NONE	RELEASED
6		Caps-0403-003	BottleCaps	NONE	06/04/96	4,000.0000		Storeroom1	NONE	NONE	RELEASED
7		CW-0411-001	WonderWater	1	06/04/96	4,000.0000		Storeroom2	1	1	RELEASED
8		CW-0411-002	WonderWater	1	06/04/96	4,000.0000		NONE	1	1	RELEASED
9		CW-0411-003	WonderWater	1	06/04/96	1,000.0000		NONE	1	1	RELEASED
10		CW-0411-004	WonderWater	1	06/04/96	1,000.0000		NONE	1	1	RELEASED

Unless otherwise specified with the optional Sort property of a Selector object, inventory lots in the Inventory Lot Selector are displayed in alphabetical order based on lot ID (the second field in the Inventory Lot Selector). The first field is reserved for color status.

Note: After it is created, the order of the lots in an Inventory Lot Selector cannot be changed by the user at runtime unless the "Allow user runtime sorting" check box is selected in the InTrack Inventory Grid Configuration dialog box. If this is done, the Inventory Lot Selector default settings and the optional Sort property of a Selector object are overridden and the user can alphabetically resort an Inventory Lot Selector by any column simply by double-clicking the selected column heading.

By default, the Inventory Lot Selector displays the following information for an inventory lot in the fields (columns) at runtime:

Field (column) Name	Description
Lot ID	Inventory lot identifier
Material	Consumable material or product in the inventory lot
Material Ver	Material version
Create Date	Date the inventory lot was received or closed to inventory
Quantity	Amount of material currently in stock at a specified location for the lot ID
Units	Quantity units of measure
Location	Inventory lot location
Customer Spec	Customer specification name
Customer Ver	Customer specification version
Hold Status	Indicates whether the inventory lot is on hold or released
Quar Status	Indicates whether the inventory lot is quarantined or active

Inventory Lot Button Wizards

Several button wizards are associated with inventory lots. At runtime, these button wizards process inventory lots. All button wizards either receive new inventory lots or act upon existing inventory lot quantities. No configuration is required for button wizards. Button wizards function according to the manufacturing model defined in InTrack ModelMaker. InTrack will not allow a button wizard to process an OLE script that was not defined for the currently selected lot in the Inventory Lot Selector.

Note: All button wizards provide the operator with a dialog box for entering required information.

The following list describes the inventory lot button wizards:

Button Wizard	Used To
Adjust	Adjust inventory lot quantities
Comment	Log a comment for an inventory lot
Configure	Show inventory lots in the Inventory Lot Selector according to specified parameters (show only quarantined inventory lots)
Merge	Merge an inventory lot into the current inventory lot
Move	Move inventory lot to another location
Receive	Receive materials into existing an inventory lot or create a new inventory lot
Ship	Track quantities of finished goods inventory shipped to customers and remove shipped quantities from the system
Split	Split an inventory lot into two separate inventory lots
Status	Change the hold or quarantine status of an inventory lot
Undo	Reverse the last transaction issued for an inventory lot

Inventory Lot Adjust

This button wizard modifies the primary quantity of an inventory lot.

To adjust an inventory lot's primary quantity at runtime, select the desired inventory lot in the Inventory Lot Selector, then click this button wizard. The Adjust Inventory Lot dialog box appears.

Inventory Lot Comment 

This button wizard logs comments for an inventory lot. The comments are stored in the LotBaseLog in the InTrack database. When a comment is added to an inventory lot, the operator ID, the time the comments were entered, and the inventory lot ID or location are logged along with the comment.

To log comments for an inventory lot at runtime, select the desired inventory lot in the Inventory Lot Selector, then click this button wizard. The Comment for Inventory Lot dialog box appears.

Inventory Lot Configure 

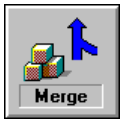
This button wizard displays inventory lots in an Inventory Lot Selector according to specified parameters. For example, an Inventory Lot Selector can be configured so that only quarantined inventory lots are displayed. Multiple parameters, including the maximum number of rows to display in a specific Inventory Lot Selector, also can be specified.

When an Inventory Lot Selector is configured, all inventory lots might not be displayed in the selector. However, as long as you provide a fully qualified SublotKey, automatic OLE scripts can still be executed against the configured inventory lots. All inventory lots are still tracked by the system, and current data will be reported when the inventory lots are configured back into the selector.

Note: To let the user double-click a column heading in an Inventory Lot Selector at runtime and alphabetically reorder the lots by the selected column, select the "Allow user runtime sorting" check box in the InTrack Inventory Grid Configuration dialog box.

To configure an Inventory Lot Selector at runtime, click this button wizard. The Inventory Lot Configuration dialog box appears.

Inventory Lot Merge



This button wizard combines two or more inventory lots into one inventory lot. Inventory lots being merged must have the same material type.

Note: A message box appears if there are no inventory lots available for merging.

To merge an inventory lot at runtime, select the desired inventory lot in the Inventory Lot Selector, then click this button wizard. The Merge with Inventory Lot dialog box appears.

Inventory Lot Move

This button wizard moves some or all of an inventory lot to another location.

Note: An inventory lot on hold or in quarantine cannot be moved.

To move an inventory lot at runtime, select the desired inventory lot in the Inventory Lot Selector, then click this button wizard. The Move from Inventory Lot dialog box appears.

Inventory Lot Receive

This button wizard receives an inventory lot. Consumable materials must first be received as inventory lots before they can be consumed at an operation. The Receive button wizard defines an inventory lot and places it in a specified optional location. Inventory lots are defined with a lot number or name, a lot quantity, and product information. The Receive button wizard also adds quantities to existing inventory lots.

Note: If the quantity of consumable material is depleted from an inventory lot, the lot ID will remain listed in the Inventory Lot Selector, even though the quantity is zero.

To receive or add material into an inventory lot at runtime, select the desired inventory lot in the Inventory Lot Selector, then click this button wizard. The Receive Inventory Lot dialog box appears.

Inventory Lot Ship

This button wizard tracks finished goods inventory being shipped to customers, and removes the shipped quantity from the database.

Note: An inventory lot on hold or in quarantine cannot be shipped.

To ship inventory at runtime, select the desired inventory lot in the Inventory Lot Selector, then click this button wizard. The Ship Inventory Lot dialog box appears.

Inventory Lot Split

This button wizard divides an inventory lot into two inventory lots. Split lots are tracked independently from each other. To split an inventory lot, it must have multiple quantities.

Note: An inventory lot on hold cannot be split.

To split an inventory lot at runtime, select the desired inventory lot in the Inventory Lot Selector, then click this button wizard. The Split Inventory Lot dialog box appears.

Inventory Lot Status

This button wizard places an inventory lot on hold or in quarantine when the manufacturing activity for the inventory lot must be stopped. The Status button wizard also releases an inventory lot from hold or quarantine. Inventory lots on hold cannot be shipped, moved, split, or consumed. Quarantined inventory lots also cannot be shipped, moved, or consumed.

To change an inventory lot's status at runtime, select the desired inventory lot in the Inventory Lot Selector, then click this button wizard. The Status of Inventory Lot dialog box appears.

Inventory Lot Undo 

This button wizard returns the selected inventory lot to the status it had prior to the most recent transaction. Several consecutive transactions can be undone by repeatedly using the Undo button wizard. It is not necessary to undo transactions immediately after performing them; as long as the transaction log has not been purged, you can undo any transaction for any inventory lot at any time.

Note: If a Split transaction is performed, and subsequent transactions performed on the split lots, transactions can only be undone back to the split.

To undo a previous inventory lot transaction at runtime, select the desired inventory lot in the Inventory Lot Selector, then click this button wizard. The Undo for Inventory Lot dialog box appears.

Bulk Inventory Wizards

Bulk inventory is a collection of material of the same type (product) stored in inventory and tracked by storage location. The display object for viewing all bulk inventory for a manufacturing route is the Bulk Inventory Selector wizard. Runtime control of bulk inventory is accomplished using either button wizards or OLE scripts. Bulk inventory button wizards are identified by two barrels and use the OLE script:

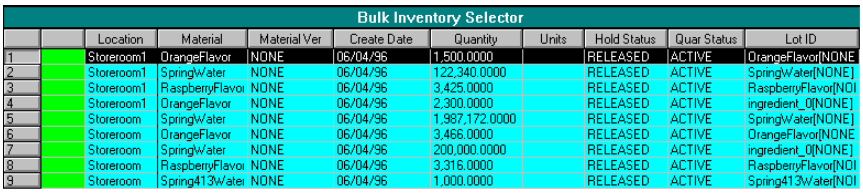
```
OLE_CreateObject(%ItDlg, "InTrack.ObjectClass");
OLE_CreateObject(%Selector, "InTrack.Selector");

%Selector.AttachTo("An Inventory Bulk Selector");
%ItDlg.Sublot = %Selector.Selection;
%ItDlg.DoneFlagName = "DoneFlag";
%ItDlg.Launch();
```

Bulk Inventory Selector Wizard

The Bulk Inventory Selector wizard is used to view all bulk inventory quantities available for consumption, and all completed products and output closed to inventory. The Bulk Inventory Selector wizard displays the current database information associated with all bulk inventory quantities. This information is displayed in a scrollable, spreadsheet type object where each row is a record, and each column is a field. The record (row) represents the actual bulk inventory and each field (column) displays associated information for the bulk inventory.

Note: When WindowViewer is first started, all selector wizards will be blank until the operator logs on to the database using the OLE System Connect script or the Connect button wizard.

After a Selector object is created, the AttachTo method associates it with a specific Bulk Inventory Selector wizard, which gives you access to all the bulk inventory within the Bulk Inventory Selector wizard. To select a bulk inventory on which to work, click anywhere on the row containing the bulk inventory. The selected row will be highlighted.



Bulk Inventory Selector

		Location	Material	Material Ver	Create Date	Quantity	Units	Hold Status	Quar Status	Lot ID
1		Storeroom1	OrangeFlavor	NONE	06/04/96	1,500.0000		RELEASED	ACTIVE	OrangeFlavor[NONE
2		Storeroom1	SpringWater	NONE	06/04/96	122,340.0000		RELEASED	ACTIVE	SpringWater[NONE]
3		Storeroom1	RaspberryFlavor	NONE	06/04/96	3,425.0000		RELEASED	ACTIVE	RaspberryFlavor[NOI
4		Storeroom1	OrangeFlavor	NONE	06/04/96	2,300.0000		RELEASED	ACTIVE	ingredient_0[NONE]
5		Storeroom	SpringWater	NONE	06/04/96	1,987,172.0000		RELEASED	ACTIVE	SpringWater[NONE]
6		Storeroom	OrangeFlavor	NONE	06/04/96	3,466.0000		RELEASED	ACTIVE	OrangeFlavor[NONE
7		Storeroom	SpringWater	NONE	06/04/96	200,000.0000		RELEASED	ACTIVE	ingredient_0[NONE]
8		Storeroom	RaspberryFlavor	NONE	06/04/96	3,316.0000		RELEASED	ACTIVE	RaspberryFlavor[NOI
9		Storeroom	Spring413Water	NONE	06/04/96	1,000.0000		RELEASED	ACTIVE	Spring413Water[NOI

Unless otherwise specified with the optional Sort property of a Selector object, bulk inventory in the Bulk Inventory Selector is displayed in alphabetical order based on location (the second field in the Bulk Inventory Selector). The first field is reserved for color status.

Note: After it is created, the order of the lots in a Bulk Inventory Selector cannot be changed by the user at runtime unless the "Allow user runtime sorting" check box is selected in the InTrack Bulk Grid Configuration dialog box. If this is done, the Bulk Inventory Selector default settings and the optional Sort property of a Selector object are overridden and the user can alphabetically resort a Bulk Inventory Selector by any column simply by double-clicking the selected column heading.

By default, the Bulk Inventory Selector displays the following information for bulk inventory in the fields (columns) at runtime:

Field (column) Name	Description
Location	Bulk inventory location
Material	Consumable material/product in bulk inventory
Material Ver	Material version
Create Date	Date the bulk material was received or closed to inventory
Quantity	Amount of bulk inventory currently in stock at the specified location
Units	Quantity units of measure
Hold Status	Indicates whether bulk inventory is on hold or released
Quar Status	Indicates whether bulk inventory is quarantined or active
Lot ID	Bulk inventory identifier

Bulk Inventory Button Wizards

The InTrack manufacturing model supports several button wizards for controlling bulk inventory. Bulk inventory button wizards include all instructions necessary for moving bulk inventory through a manufacturing environment. All supported button wizards either create new bulk inventory or act upon existing bulk inventory. No configuration is required for button wizards. Button wizards function according to the manufacturing model defined in InTrack ModelMaker. InTrack will not let a button wizard process an OLE script that was not defined.

Note: All button wizards provide the operator with a dialog box for entering required information.

The following list describes the bulk inventory button wizards:

Button Wizard	Used To
Adjust	Adjust bulk inventory quantities
Comment	Log a comment for a bulk inventory
Configure	Show bulk inventory in the Bulk Inventory Selector according to specified parameters (show only quarantined bulk inventory)
Move	Move bulk inventory to another location
Receive	Receive materials into existing bulk inventory or create new bulk inventory
Ship	Track quantities of finished goods inventory shipped to customers and remove shipped quantities from the system
Status	Change the hold or quarantine status of a bulk inventory
Undo	Reverse the last transaction issued for a bulk inventory

Bulk Inventory Adjust

This button wizard modifies the primary quantity of a bulk inventory.

To adjust a bulk inventory's primary quantity at runtime, select the desired bulk inventory in the Bulk Inventory Selector, then click this button wizard. The Adjust Bulk Inventory dialog box appears.

Bulk Inventory Comment 

This button wizard logs comments for a bulk inventory. The comments are saved in the LotBaseLog in the InTrack database. The Comment button wizard records the operator ID, the time the comments were entered, and the bulk inventory location along with the comment.

To log comments for a bulk inventory at runtime, select the desired bulk inventory in the Bulk Inventory Selector, then click this button wizard. The Comment for Bulk Inventory dialog box appears.

Bulk Inventory Configure 

This button wizard displays bulk inventory in a Bulk Inventory Selector according to specified parameters. For example, a Bulk Inventory Selector can be configured so that only quarantined bulk inventory appears in the selector. Multiple parameters, including the maximum number of rows to display in a specific Bulk Inventory Selector, also can be specified.

When a Bulk Inventory Selector is configured, all the bulk inventory might not be displayed in the selector. However, as long as you provide a fully qualified SublotKey, automatic OLE scripts can still be executed against the configured bulk inventory. All bulk inventory is still tracked by the system, and current data will be reported when the bulk inventory is configured back into the selector.

Note: To let the user double-click a column heading in a Bulk Inventory Selector at runtime and alphabetically reorder the lots by the selected column, select the "Allow user runtime sorting" check box in the InTrack Bulk Grid Configuration dialog box.

To configure a Bulk Inventory Selector at runtime, click this button wizard. The Bulk Inventory Configuration dialog box appears.

Bulk Inventory Move 

This button wizard moves some or all of a bulk inventory into another storage location.

To move bulk inventory at runtime, select the desired bulk inventory in the Bulk Inventory Selector, then click this button wizard. The Move from Bulk Inventory dialog box appears.

Bulk Inventory Receive

This button wizard receives bulk material into existing bulk inventory or creates new bulk inventory. Bulk inventory is typically specified by volume or weight and is not assigned a lot ID. The Receive Bulk Inventory dialog box defines bulk inventory and places it in a specified inventory location.

Note: If the quantity of consumable material is depleted from bulk inventory, the inventory remains listed in the Bulk Inventory Selector, even though the quantity is zero.

To receive bulk inventory at runtime, select the desired bulk inventory in the Bulk Inventory Selector, then click this button wizard. The Receive Bulk Inventory dialog box appears.

Bulk Inventory Ship

This button wizard tracks bulk finished goods inventory being shipped to customers, and removes the shipped quantity from the database. Bulk inventory on hold or in quarantine cannot be shipped.

To ship bulk inventory at runtime, select the desired bulk inventory in the Bulk Inventory Selector, then click this button wizard. The Ship Bulk Inventory dialog box appears.

Bulk Inventory Status

This button wizard places or releases bulk inventory on hold or in quarantine when the manufacturing activity for the bulk inventory must be stopped. Bulk inventory is quarantined when found to be unusable. Bulk inventory on hold cannot be shipped, moved, or consumed until it is released. Quarantined bulk inventory cannot be shipped, moved, or consumed until it is released.

To change a bulk inventory's status at runtime, select the desired bulk inventory in the Bulk Inventory Selector, then click this button wizard. The Status of Bulk Inventory dialog box appears.

Bulk Inventory Undo 

This button wizard returns the selected bulk inventory to the status it had prior to the most recent transaction. Several consecutive transactions can be undone by repeatedly clicking the Undo button wizard. As long as the transaction log has not been purged, you can undo any transaction for any bulk inventory at any time.

To undo a previous bulk inventory transaction at runtime, select the desired bulk inventory in the Bulk Inventory Selector, then click this button wizard. The Undo for Bulk Inventory dialog box appears.

Machine Wizards

Machine tracking manages machine maintenance tasks and repairs. The display object for viewing all machine functions is the Machine Selector wizard. Runtime control of machine tracking is accomplished using either button wizards or OLE scripts. Machine button wizards are identified by a machine and use the OLE script:

```
OLE_CreateObject(%ItDlg, "InTrack.ObjectClass");
OLE_CreateObject(%Selector, "InTrack.Selector");

%Selector.AttachTo("Machine Selector");
%ItDlg.Sublot = %Selector.Selection;
%ItDlg.DoneFlagName = "DoneFlag";
%ItDlg.Launch();
```

Machine Selector Wizard

The Machine Selector wizard is used to view machine status, repair, and capacity information. The Machine Selector wizard displays all current database information associated with all machines. This information is displayed in a scrollable, spreadsheet type object where each row is a record, and each column is a field. The record (row) represents the machine and each field (column) displays associated information for the machine.

Note: When WindowViewer is first started, all selector wizards will be blank until the operator logs on to the database using the OLE System Connect script or the Connect button wizard.

After a Selector object is created by clicking the button wizard or generating your own OLE script, the OLE AttachTo method associates it with a specific Machine Selector wizard. This gives you access to all the machines within the Machine Selector wizard. To select a machine, click anywhere on the row containing the machine. The selected row will be highlighted.

Machine Selector

		Machine	Operation Status	Maint Status	Open Repair Order	Capacity Qty	Capacity UOM	Free Capacity
1		Mixer	IN_USE	READY		1,000.0000	gallons	500.0000
2		Labeler	OFFLINE	AWAITING_REPAIR	Clean	1,000.0000	each	1,000.0000
3		BottleCleaner	IDLE	READY		1,000.0000	each	1,000.0000
4		BottlePacker	IDLE	READY		1,000.0000	each	1,000.0000
5		BottleFiller	IDLE	IN_PM		1,000.0000	gallons	1,000.0000

Unless otherwise specified with the optional Sort property of a Selector object, machines in the Machine Selector are displayed in alphabetical order based on the machine name (the second field in the machine selector). The first field is reserved for color status.

Note: After it is created, the order of the machines in a Machine Selector cannot be changed by the user at runtime unless the "Allow user runtime sorting" check box is selected in the InTrack Machine Grid Configuration dialog box. If this is done, the Machine Selector default settings and the optional Sort property of a Selector object are overridden and the user can alphabetically resort a Machine Selector by any column simply by double-clicking the selected column heading.

By default, the Machine Selector displays the following information for machines in the fields (columns) at runtime:

Field (column) Name	Description
Machine	Machine name
Operation Status	Machine operation status
Maint Status	Machine maintenance status
Open Repair Order	Open repair order name
Capacity	Total machine capacity
Capacity UOM	Capacity units of measure
Free Capacity	Available machine capacity

Machine Button Wizards

The InTrack manufacturing model supports several button wizards for machines. Machine button wizards include all instructions necessary to control machine repairs and procedures. No configuration is required for button wizards. Button wizards function according to the manufacturing model defined in InTrack ModelMaker. InTrack will not let a button wizard process an OLE script that was not defined.

Note: All button wizards provide the operator with a dialog box for entering required information.

The following list describes each type of machine button wizards:

Button Wizard	Used To
Close	Close a machine repair order
Complete	Complete a machine maintenance procedure
Configure	Show machines in the Machine Selector according to specified parameters (show only idle machines)
Open	Open a machine repair order
Start	Start a machine maintenance procedure
Status	Permit or prevent machine starts at runtime
Update	Update repair information for an in-repair machine

Machine Close

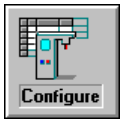
This button wizard closes a repair record for a specified machine.

To close a machine repair at runtime, choose the desired machine in the Machine Selector, and then click this button wizard. The Close Repair dialog box appears.

Machine Complete 

This button wizard completes a task on a machine. A task is a maintenance procedure that has been defined in ModelMaker. A task can be periodic (after 20 starts, recalibrate the machine) or non-periodic (recalibrate the machine at any one time).

To complete a machine task at runtime, choose the desired machine in the Machine Selector, and then click this button wizard. The Complete Task dialog box appears.

Machine Configure 

This button wizard displays machines in a Machine Selector according to specified parameters. For example, a Machine Selector can be configured so that only machines with an operational status of IDLE and a maintenance status of READY are displayed. The maximum number of rows to display in a specific Machine Selector also can be specified.

When a Machine Selector is configured, all machines might not be displayed in the selector. However, as long as you provide a fully qualified NameKey for the machine, automatic OLE scripts can still be executed against the configured machines. All machines are still tracked by the system, and current data will be reported when the machines are configured back into the selector.

Note: To let the user double-click a column heading in a Machine Selector at runtime and alphabetically reorder the lots by the selected column, select the "Allow user runtime sorting" check box in the InTrack Machine Grid Configuration dialog box.

To configure a Machine Selector at runtime, click this button wizard. The Machine Configuration dialog box appears.

Machine Open 

This button wizard opens a repair order for a specified machine. Only one repair can be opened for a machine at any one time.

To open a machine repair at runtime, choose the desired machine in the Machine Selector, and then click this button wizard. The Open Repair dialog box appears.

Machine Start 

This button wizard starts a task on a machine. A task is a maintenance procedure that has been defined in ModelMaker. A task can be periodic (after 20 starts, recalibrate the machine) or non-periodic (recalibrate the machine at any one time).

To start a machine task at runtime, choose the desired machine in the Machine Selector, and then click this button wizard. The Start Task dialog box appears.

Machine Status 

This button wizard permits or prevents machine starts on a machine at runtime.

To permit or prevent machine starts at runtime, choose the desired machine in the Machine Selector, and then click this button wizard. The Machine Status dialog box appears.

Machine Update 

This button wizard updates a repair record for a machine in-repair.

To update a machine repair at runtime, choose the desired machine in the Machine Selector, and then click this button wizard. The Update Repair dialog box appears.

CHAPTER 5

InTrack Database Schema

Table Name		Column Name	Data Type	Length	Description
AdjustLog	1	TransLogID	char	17	Internal transaction log index
	2	AdjustedAttrib	char	15	Attribute type: (DUE_DATE / ID / MATERIAL / PRI_Q_QTY / PRI_S_QTY / PRIORITY / ROUTE_STEP / SEC_QTY / UDA)
	3	Attribute	varchar	30	Attribute name: (predefined or <column name>)
	4	DataType	smallint	2	Data type identifier
	5	OldStringAttrib	varchar	80	Attribute string value before adjustment
	6	NewStringAttrib	varchar	80	Attribute string value after adjustment
	7	OldIntAttrib	int	4	Attribute integer value before adjustment
	8	NewIntAttrib	int	4	Attribute integer value after adjustment
	9	OldRealAttrib	real	4	Attribute real value before adjustment
	10	NewRealAttrib	real	4	Attribute real value after adjustment
	11	OldDoubleAttrib	float	8	Attribute double value before adjustment
	12	NewDoubleAttrib	float	8	Attribute double value after adjustment
	13	OldDateTimeAttrib	datetime	8	Attribute date-time value before adjustment
	14	NewDateTimeAttrib	datetime	8	Attribute date-time value after adjustment
	15	OldDateAttrib	datetime	8	Attribute date value before adjustment
	16	NewDateAttrib	datetime	8	Attribute date value after adjustment
	17	OldTimeInterAttrib	int	4	Attribute time interval value before adjustment
	18	NewTimeInterAttrib	int	4	Attribute time interval value after adjustment
	19	OldTimeAttrib	datetime	8	Attribute time value before adjustment
	20	NewTimeAttrib	datetime	8	Attribute time value after adjustment
	21	SublotIndex	int	4	Internal index of sublot whose attribute value was adjusted
			Table Total:	**316**	

BillOfMaterial	1	BOMMaterialName	varchar	30	Name of target material associated with bill of material
	2	BOMMaterialVersion	varchar	8	Version of target material associated with bill of material
	3	StandardQty	float	8	Default quantity produced
	4	StandardUOM	varchar	16	Default quantity unit of measure
	5	LockCount	smallint	2	Internal table usage count
			Table Total:	**64**	

Table Name		Column Name	Data Type	Length	Description
BOMItems	1	BOMMaterialName	varchar	30	Name of target material associated with bill of material
	2	BOMMaterialVersion	varchar	8	Version of target material associated with bill of material
	3	ItemIndex	int	4	Internal item index
	4	ItemID	int	4	Sort sequence number of input or output material
	5	ItemType	char	15	Type of material: (INPUT / OUTPUT)
	6	MaterialName	varchar	30	Material name
	7	MaterialVersion	varchar	8	Material version
	8	StandardQty	float	8	Default quantity consumed
	9	StandardUOM	varchar	16	Default quantity unit of measure
	10	UpperTolerance	float	8	Maximum tolerable quantity deviation: (Absolute: +20% = 1.2)
	11	LowerTolerance	float	8	Minimum tolerable quantity deviation: (Absolute: -20% = 0.8)
	12	Scaled	char	5	Enable quantity scaling: (TRUE / FALSE)
	13	AdjPrimaryQty	char	5	Decrease primary quantity by material output: (TRUE / FALSE)
			Table Total:	**149**	

BOMSubItems	1	BOMMaterialName	varchar	30	Name of target material associated with bill of material
	2	BOMMaterialVersion	varchar	8	Version of target material associated with bill of material
	3	ItemIndex	int	4	Internal input item index
	4	SubItemIndex	int	4	Internal substitute item index
	5	ItemID	int	4	Sort sequence number of input material
	6	ItemType	char	15	Type of material: (INPUT)
	7	MaterialName	varchar	30	Name of substitute material
	8	MaterialVersion	varchar	8	Version of substitute material
	9	StandardQty	float	8	Default quantity consumed
	10	StandardUOM	varchar	16	Default quantity unit of measure
	11	UpperTolerance	float	8	Maximum tolerable quantity deviation: (Absolute: +20% = 1.2)
	12	LowerTolerance	float	8	Minimum tolerable quantity deviation: (Absolute: -20% = 0.8)
	13	MaxSubstituteRatio	float	8	Maximum allowable quantity of substitute material expressed as a ratio: (substitute quantity ÷ input quantity)
			Table Total:	**151**	

Calendar	1	Name	varchar	30	Calendar name
	2	Description	varchar	80	Descriptive comments
	3	FirstDayOfWeek	smallint	2	Specified first day of calendar week
	4	FirstMonthOfYear	smallint	2	Specified first month of calendar year
	5	PeriodicHolidayMap	smallint	2	Internal mask denoting periodic holidays in week
	6	LockCount	smallint	2	Internal table usage count
			Table Total:	**118**	

ClassDefinition	1	InstClassId	smallint	2	Internal instance class identifier
	2	BaseClassId	smallint	2	Internal base class identifier
	3	InstClassName	varchar	32	Internal instance class name
	4	InstMethod	int	4	Internal instance method
	5	KeyMethod	int	4	Internal key method
	6	LockCount	smallint	2	Internal table usage count
	7	Description	varchar	32	Internal class description
	8	TableSpace	varchar	32	Internal tablespace
	9	ClassAttribute	image	16	Internal class attribute
			Table Total:	**126**	

Table Name		Column Name	Data Type	Length	Description
ClassDictionary	1	Name	varchar	32	Object name
	2	Version	smallint	2	Internal version identifier
	3	CreateTime	datetime	8	Creation time stamp
	4	NextClassId	smallint	2	Next class identifier
	5	LockCount	smallint	2	Internal table usage count
			Table Total:	**46**	

CloseLog	1	TransLogID	char	17	Internal transaction log index
	2	FromRouteName	varchar	30	From route name
	3	FromRouteVersion	varchar	8	From route version
	4	FromRouteStepName	varchar	30	From route step name
	5	FromLocation	varchar	30	From location
	6	FromOperationName	varchar	30	From operation name
	7	FromOperationVersion	varchar	8	From operation version
	8	QueueTime	int	4	Cumulative time lot queued: (seconds)
	9	OperationTime	int	4	Cumulative lot processing time: (seconds)
	10	DispositionCode	varchar	30	Lot condition code
	11	Machine	varchar	30	Machine name
	12	ForcedQty	float	8	Lot primary quantity forced to close
	13	ForcedUOM	varchar	16	Forced quantity's unit of measure
	14	InventoryLotID	varchar	40	Destination inventory lot identifier
	15	InventoryLocation	varchar	30	Destination inventory location name
	16	QuarantineStatus	char	15	Lot quarantine status: (ACTIVE / QUARANTINED)
	17	InvCustSpecName	varchar	30	Name of customer specification for inventory lot
	18	InvCustSpecVersion	varchar	8	Version of customer specification for inventory lot
	19	InvLotCreated	char	5	Inventory lot created: (TRUE / FALSE)
			Table Total:	**373**	

CloseLogCollection	1	TransLogID	char	17	Internal transaction log index
	2	ItemIndex	int	4	Internal index of data sets collected for this lot
	3	RouteName	varchar	30	Route name
	4	RouteVersion	varchar	8	Route version
	5	RouteStepName	varchar	30	Route step name
	6	Location	varchar	30	Location name
	7	DSTemplateName	varchar	30	Data set template name
	8	DSTemplateVersion	varchar	8	Data set template version
	9	RTDataTableName	varchar	28	Runtime data set table name
	10	RequiredSampleCount	int	4	Number of required samples collected to date
	11	SampleName	varchar	30	Name of data sample collected
			Table Total:	**219**	

Table Name		Column Name	Data Type	Length	Description
CloseLogConsumption		TransLogID	char	17	Internal transaction log index
	2	ItemIndex	int	4	Internal index of material consumed for this lot
	3	RouteName	varchar	30	Route name
	4	RouteVersion	varchar	8	Route version
	5	RouteStepName	varchar	30	Route step name
	6	Location	varchar	30	Location name
	7	ConsumedMaterialName	varchar	30	Name of material consumed
	8	ConsumedMaterialVersion	varchar	8	Version of material consumed
	9	SubstituteMaterialName	varchar	30	Name of substitute material consumed
	10	SubstituteMaterialVersion	varchar	8	Version of substitute material consumed
	11	ConsumedQty	float	8	Quantity of material consumed
	12	ConsumedUOM	varchar	16	Consumed material's unit of measure
	13	SubstituteConsumedQty	float	8	Quantity of substitute material consumed
	14	SubstituteConsumedUOM	varchar	16	Consumed substitute material's unit of measure
	15	ConsumerSerialNumber	varchar	30	Serial number of lot unit that consumed the material
	16	ConsumerMachine	varchar	30	Name of machine that consumed material
			Table Total:	**303**	

CommentLog		TransLogID	char	17	Internal transaction log index
			Table Total:	**17**	

CompleteLog	1	TransLogID	char	17	Internal transaction log index
	2	FromRouteName	varchar	30	From route name
	3	FromRouteVersion	varchar	8	From route version
	4	FromRouteStepName	varchar	30	From route step name
	5	FromLocation	varchar	30	From location
	6	FromOperationName	varchar	30	From operation name
	7	FromOperationVersion	varchar	8	From operation version
	8	QueueTime	int	4	Cumulative time lot queued: (seconds)
	9	OperationTime	int	4	Cumulative lot processing time: (seconds)
	10	DispositionCode	varchar	30	Lot condition code
	11	Machine	varchar	30	Machine name
	12	ForcedQty	float	8	Lot primary quantity forced to complete
	13	ForcedUOM	varchar	16	Forced quantity's unit of measure
	14	GradedLotID	varchar	40	Graded lot identifier
	15	GradedMaterialName	varchar	30	Graded material name
	16	GradedMaterialVersion	varchar	8	Graded material version
	17	CompletionType	char	15	Type of COMPLETE transaction: (ON_ROUTE / OUTPUT / SCRAP)
			Table Total:	**338**	

CompleteLogCollection	TransLogID	char	17	Internal transaction log index
2	ItemIndex	int	4	Internal index of data sets collected for this lot
3	RouteName	varchar	30	Name of route on which data was collected
4	RouteVersion	varchar	8	Version of route on which data was collected
5	RouteStepName	varchar	30	Name of route step at which data was collected
6	Location	varchar	30	Name of location in which data was collected
7	DSTemplateName	varchar	30	Name of data set template used to collect data
8	DSTemplateVersion	varchar	8	Version of data set template used to collect data
9	RTDataTableName	varchar	28	Runtime data set table name
10	RequiredSampleCount	int	4	Number of required samples collected to date
11	SampleName	varchar	30	Name of data sample collected
		Table Total:	**219**	

Table Name		Column Name	Data Type	Length	Description
CompleteLogConsumption		TransLogID	char	17	Internal transaction log index
		ItemIndex	int	4	Internal index of material consumed for this lot
	3	RouteName	varchar	30	Route name
	4	RouteVersion	varchar	8	Route version
	5	RouteStepName	varchar	30	Route step name
	6	Location	varchar	30	Location name
	7	ConsumedMaterialName	varchar	30	Name of material consumed
	8	ConsumedMaterialVersion	varchar	8	Version of material consumed
	9	SubstituteMaterialName	varchar	30	Name of substitute material consumed
	10	SubstituteMaterialVersion	varchar	8	Version of substitute material consumed
	11	ConsumedQty	float	8	Quantity of material consumed
	12	ConsumedUOM	varchar	16	Consumed material's unit of measure
	13	SubstituteConsumedQty	float	8	Quantity of substitute material consumed
	14	SubstituteConsumedUOM	varchar	16	Consumed substitute material's unit of measure
	15	ConsumerSerialNumber	varchar	30	Serial number of lot unit that consumed the material
	16	ConsumerMachine	varchar	30	Name of machine that consumed material
			Table Total:	**303**	

Configuration	1	OptionName	varchar	30	Configuration option name
	2	Description	varchar	80	Configuration option description
	3	StringValue	varchar	80	Configuration option string value
	4	IntegerValue	int	4	Configuration option integer value
	5	DoubleValue	float	8	Configuration option real value
	6	LockCount	smallint	2	Internal table usage count
			Table Total:	**204**	

ConsumeLog	1	TransLogID	char	17	Internal transaction log index
	2	ConsumedMaterialName	varchar	30	Name of material consumed
	3	ConsumedMaterialVersion	varchar	8	Version of material consumed
	4	ConsumedLotID	varchar	40	Lot identifier
	5	ConsumptionRouteName	varchar	30	Route name
	6	ConsumptionRouteVersion	varchar	8	Route version
	7	ConsumptionRouteStep	varchar	30	Route step name
	8	ConsumptionLocation	varchar	30	Location name
	9	ForcedConsumption	char	5	Material consumption forced at this route step: (TRUE / FALSE)
	10	ConsumerSerialNumber	varchar	30	Serial number of lot unit that consumed the material
	11	ConsumerMachine	varchar	30	Name of machine that consumed material
	12	ConsumptionRatio	float	8	Ratio of quantity consumed while in the process status – if process status is QUEUED: (quantity consumed ÷ quantity queued) – if process status is IN_PROCESS: (quantity consumed ÷ quantity in process)
			Table Total:	**266**	

CreateLog	1	TransLogID	char	17	Internal transaction log index
	2	Priority	smallint	2	Lot priority value
	3	DueDate	datetime	8	Lot due date
	4	ParentLotID	varchar	40	Source lot identifier
	5	VendorID	varchar	30	Inventory vendor name
	6	VendorLotNum	varchar	30	Vendor's inventory lot number
			Table Total:	**127**	

CustomerSpecification	CustSpecName	varchar	30	Customer specification name
2	CustSpecVersion	varchar	8	Customer specification version
3	Description	varchar	80	Descriptive comments
4	LockCount	smallint	2	Internal table usage count
		Table Total:	**120**	

Table Name		Column Name	Data Type	Length	Description
DataCollectLog	1	TransLogID	char	17	Internal transaction log index
	2	DSTemplateName	varchar	30	Data set template name
	3	DSTemplateVersion	varchar	8	Data set template version
	4	RTDataTableName	varchar	28	Runtime data set table name
	5	CollectTime	datetime	8	Time stamp when data sample collected
	6	SampleName	varchar	30	Name of data sample collected
	7	Machine	varchar	30	Machine name
	8	RouteName	varchar	30	Route name
	9	RouteVersion	varchar	8	Route version
	10	RouteStepName	varchar	30	Route step name
			Table Total:	**219**	

DaysOfMonth	1	MonthID	char	17	Internal month identifier
	2	DayIndex	int	4	Internal day of month index (relative to specified first day of month)
	3	DayOfMonth	smallint	2	Day of month
	4	UseDefaultShifts	char	5	Use default shifts: (TRUE / FALSE)
			Table Total:	**28**	

DefaultShifts	1	Name	varchar	30	Calendar name
	2	DefaultShiftIndex	int	4	Internal default shift index
	3	DefaultShiftID	char	17	Internal default shift identifier
			Table Total:	**51**	

DisassembleLog	1	TransLogID	char	17	Internal transaction log index
	2	ToLotID	varchar	40	Lot identifier of destination lot
	3	ToMaterialName	varchar	30	Material name of destination lot
	4	ToMaterialVersion	varchar	8	Material version of destination lot
	5	ToRouteName	varchar	30	Route name of destination lot
	6	ToRouteVersion	varchar	8	Route version of destination lot
	7	ToRouteStep	varchar	30	Route step name of destination lot
	8	ToLocation	varchar	30	Location name of destination lot
	9	CreatedSublot	char	5	New destination sublot created: (TRUE / FALSE)
	10	ConsumerSerialNumber	varchar	30	Serial number of lot unit that consumed the material before disassembly and from which components are being disassembled
			Table Total:	**228**	

DispositionCode	1	DispositionCode	varchar	30	Lot condition code
	2	Description	varchar	80	Descriptive comments
	3	DispositionType	char	15	Code condition type: (GOOD / REJECT / REWORK / SCRAP)
	4	LockCount	smallint	2	Internal table usage count
			Table Total:	**127**	

DSBaseRTValues	1	SampleName	varchar	30	Name of sample data collected
	2	CollectTime	datetime	8	Time stamp when data sample collected
	3	LotID	varchar	40	Lot identifier
	4	Machine	varchar	30	Machine name
	5	Location	varchar	30	Location name
	6	OperationName	varchar	30	Operation name
	7	OperationVersion	varchar	8	Operation version
	8	UserName	varchar	30	Name of user who is logged on at time of collection
	9	Superseded	char	5	Data superseded: (TRUE / FALSE)
	10	TimeSuperseded	datetime	8	Time stamp when data superseded
	11	SampleIndex	int	4	Internal index for sample name
			Table Total:	**223**	

Table Name		Column Name	Data Type	Length	Description
DSDataItemHeader	1	DSTemplateName	varchar	30	Data set template name
	2	DSTemplateVersion	varchar	8	Data set template version
	3	DataItemIndex	int	4	Internal item index
	4	DataItemName	varchar	30	Runtime SQL table column name
	5	DataItemDescription	varchar	80	Runtime SQL table column description
	6	DataItemLabel	varchar	16	Data item's runtime interface label
	7	DataItemDataReqFlag	char	5	Data collection required: (TRUE / FALSE)
	8	DataItemDataType	char	29	Item's data type: (ALPHANUMERIC / ANALOG / BOOLEAN / DATE / DATETIME / INTEGER / SELECTION / TIME / TIMEINTERVAL)
	9	DataItemFormat	varchar	254	Comma-separated set of string selections
	10	DataItemLimChkFlag	char	5	Perform limit checking on this item: (TRUE / FALSE)
	11	DataItemFloat	real	4	Default float value
	12	DataItemIntegerData	int	4	Default integer value
	13	DataItemString	varchar	80	Default string value
	14	DataItemBooleanData	smallint	2	Default Boolean value
	15	DataItemDateData	datetime	8	Default date value
	16	DataItemDateTimeData	datetime	8	Default date-time value
	17	DataItemTimeIntervalData	int	4	Default time interval value
	18	DataItemTimeData	datetime	8	Default time value
	19	DataItemRealInstrHigh	real	4	Maximum real value for instrument
	20	DataItemRealHHighLimit	real	4	High-high real value range limit
	21	DataItemRealHighLimit	real	4	High real value range limit
	22	DataItemRealLowLimit	real	4	Low real value range limit
	23	DataItemRealLLowLimit	real	4	Low-low real value range limit
	24	DataItemRealInstrLow	real	4	Minimum real value for instrument
	25	DataItemIntInstrHigh	int	4	Maximum integer value for instrument
	26	DataItemIntHHighLimit	int	4	High-high integer value range limit
	27	DataItemIntHighLimit	int	4	High integer value range limit
	28	DataItemIntLowLimit	int	4	Low integer value range limit
	29	DataItemIntLLowLimit	int	4	Low-low integer value range limit
	30	DataItemIntInstrLow	int	4	Minimum integer value for instrument
	31	DataItemHighHighFlag	char	5	Enable lot hold for high-high value: (TRUE / FALSE)
	32	DataItemHighFlag	char	5	Enable lot hold for high value:
	33	DataItemLowFlag	char	5	Enable lot hold for low value:
	34	DataItemLowLowFlag	char	5	Enable lot hold for low-low value:
	35	DataItemClassID	smallint	2	Internal UDA object type identifier
	36	DataItemStringWidth	int	4	Width of the string column
	37	DataItemDouble	float	8	Data item value
	38	DataItemDblInstrHigh	float	8	Data item instrument high limit value
	39	DataItemDblHHighLimit	float	8	Data item high high limit value
	40	DataItemDblHighLimit	float	8	Data item high limit value
	41	DataItemLowLimit	float	8	Data item low limit value
	42	DataItemLLowLimit	float	8	Data item low low limit value
	43	DataItemDblInstrLow	float	8	Data item instrument low limit value
			Table Total:	**688**	

DSSampleHeader	1	DSTemplateName	varchar	30	Data set template name
	2	DSTemplateVersion	varchar	8	Data set template version
	3	SampleIndex	int	4	Internal sample index
	4	SampleName	varchar	30	Data collection sample name
	5	SampleDescription	varchar	80	Sample descriptive comments
			Table Total:	**152**	

Table Name		Column Name	Data Type	Length	Description
DSTemplate	1	DSTemplateName	varchar	30	Data set template name
	2	DSTemplateVersion	varchar	8	Data set template version
	3	Description	varchar	80	Descriptive comments
	4	Type	char	15	Data set type: (UDA / DS)
	5	SamplingPlan	char	15	Sampling plan: (VARIABLE / FIXED)
	6	RTDataTableName	varchar	28	Runtime data set table name
	7	RTDataTableClassID	smallint	2	Internal runtime data set table class
	8	RequiredSampleCount	int	4	Number of required samples
	9	RequiredSampleRatio	real	4	Number of required samples, expressed as ratio to material quantity: ($0.0 \leq n \leq 1.0$)
	10	SampleNameSeed	varchar	30	Prefix for sample name generator
	11	LockCount	smallint	2	Internal table usage count
	12	ParentDSTemplateName	varchar	30	Parent data set template name
	13	ParentDSTemplateVersion	varchar	8	Parent data set template version
			Table Total:	**256**	

HoldLog	1	TransLogID	char	17	Internal transaction log index
	2	OldHoldStatus	varchar	16	Most recent lot hold status: (RELEASED / ON_HOLD)
	3	NewHoldStatus	varchar	16	Current lot hold status: (RELEASED / ON_HOLD)
			Table Total:	**49**	

IssuedUserCert	1	Name	varchar	30	User name
	2	ElementIndex	int	4	Internal user certification index
	3	UserCertification	varchar	30	User certification name
	4	IssueTime	datetime	8	User certification issuance time stamp
	5	LockCount	smallint	2	Internal table usage count
			Table Total:	**74**	

Location	1	Location	varchar	30	Location name
	2	Description	varchar	80	Descriptive comments
	3	LockCount	smallint	2	Internal table usage count
			Table Total:	**112**	

Lot	1	LotID	varchar	40	Lot identifier
	2	MaterialName	varchar	30	Material name
	3	MaterialVersion	varchar	8	Material version
	4	HoldStatus	char	15	Lot hold status: (RELEASED / ON_HOLD)
	5	QuarantineStatus	char	15	Lot quarantine status: (ACTIVE / QUARANTINED)
	6	CreateDate	datetime	8	Lot creation time stamp
	7	ExpirationDate	datetime	8	Lot expiration time stamp: (creation date-time + lifetime of material)
	8	ZeroDate	datetime	8	Time stamp when the lot's primary and secondary quantities reached zero
	9	PrimaryQty	float	8	Lot primary quantity
	10	PrimaryUOM	varchar	16	Unit of measure for lot primary quantity
	11	SecondaryQty	float	8	Lot secondary quantity
	12	SecondaryUOM	varchar	16	Unit of measure for lot secondary quantity
	13	CustSpecName	varchar	30	Customer specification name
	14	CustSpecVersion	varchar	8	Customer specification version
	15	Priority	smallint	2	Lot priority value
	16	DueDate	datetime	8	Lot due date
	17	LotSuffix	smallint	2	Name suffix for new lot this lot generates
	18	NextSublotIndex	int	4	Internal index for identifying next sublot to create
	19	LastSublotIndex	int	4	Internal index for identifying last sublot deleted from database
	20	LockCount	smallint	2	Internal table usage count
			Table Total:	**240**	

Table Name		Column Name	Data Type	Length	Description
LotBaseLog	1	BaseLogID	char	17	Internal lot log index
	2	TimeLogged	datetime	8	Transaction time stamp (might be backdated)
	3	ActualTime	datetime	8	Actual time stamp for transaction
	4	UserName	varchar	30	Name of logged on user
	5	TransactionName	char	26	Transaction name: (ADJUST / CLOSE / COLLECT_DATA / COMMENT / COMPLETE / CONSUME / CREATE/RECEIVE / HOLD / MERGE / MOVE / QUARANTINE / RELEASE_HOLD / RELEASE_QUAR / SHIP / SPLIT / START / TRANSFER / UNDO)
	6	CalendarName	varchar	30	Calendar name in use when transaction occurred
	7	CalendarDay	smallint	2	Sequential day in calendar month
	8	CalendarMonth	smallint	2	Sequential month in calendar year
	9	CalendarYear	smallint	2	Sequential year in calendar
	10	CalendarWorkWeek	smallint	2	Sequential work week in calendar
	11	CalendarDayOfWeek	smallint	2	Sequential day in calendar week (value is **zero**-based)
	12	CalendarShift	varchar	30	Calendar shift name
	13	CommentText	varchar	255	Transaction comments
	14	Undone	char	5	Transaction has been undone: (TRUE / FALSE)
	15	CompoundTrans	char	5	Transaction is part of a compound transaction: (TRUE / FALSE)
	16	TransLogIDType	smallint	2	Internal transaction log type
	17	TransLogID	char	17	Internal transaction log index
	18	LotID	varchar	40	Lot identifier
	19	MaterialName	varchar	30	Material name
	20	MaterialVersion	varchar	8	Material version
	21	PrimaryQty	float	8	Lot primary quantity
	22	PrimaryUOM	varchar	16	Unit of measure for lot primary quantity
	23	SecondaryQty	float	8	Lot secondary quantity
	24	SecondaryUOM	varchar	16	Unit of measure for lot secondary quantity
	25	PrimaryTransQty	float	8	Lot primary quantity for transaction
	26	PrimaryTransUOM	varchar	16	Unit of measure for lot primary transaction quantity
	27	SecondaryTransQty	float	8	Lot secondary quantity for transaction
	28	SecondaryTransUOM	varchar	16	Unit of measure for lot secondary transaction quantity
	29	RouteName	varchar	30	Route name
	30	RouteVersion	varchar	8	Route version
	31	RouteStepName	varchar	30	Route step name
	32	OperationName	varchar	30	Operation name
	33	OperationVersion	varchar	8	Operation version
	34	Location	varchar	30	Location name
			Table Total:	**753**	

Lot Collection	1	LotID	varchar	40	Lot identifier
	2	SublotIndex	int	4	Internal sublot index
	3	RouteName	varchar	30	Route name
	4	RouteVersion	varchar	8	Route version
	5	RTDataTableName	varchar	30	Route data table name
	6	SampleName	varchar	30	Sample name
	7	SampleCount	int	4	Sample count
	8	LockCount	smallint	2	Internal table usage count
			Table Total:	**118**	

Table Name		Column Name	Data Type	Length	Description
LotConsumption	1	LotID	varchar	40	Lot identifier
	2	SublotIndex	int	4	Internal sublot index
	3	ConsumedMaterialName	varchar	30	Consumed material name
	4	ConsumedMaterialVersion	varchar	8	Consumed material version
	5	ConsumerSerialNumber	varchar	30	Machine serial number
	6	ConsumerMachine	varchar	30	Machine name
	7	SubstituteMaterialName	varchar	30	Substitute material name
	8	SubstituteMaterialVersion	varchar	8	Substitute material version
	9	ConsumedQty	float	8	Consumed quantity
	10	ConsumedUOM	varchar	16	Consumed unit of measure
	11	SubsituteConsumedQty	float	8	Substitute consumed quantity
	12	SubstituteConsumedUOM	varchar	16	Substitute consumed unit of measure
	13	LockCount	smallint	2	Internal table usage count
			Table Total:	**266**	

LotUndoLog	1	TransLogID	char	17	Internal transaction log index
	2	UndoTransID	char	17	Internal index of undone transaction
			Table Total:	**34**	

Machine	1	Machine	varchar	30	Machine name
	2	Description	varchar	80	Descriptive comments
	3	Vendor	varchar	30	Vendor name
	4	Model	varchar	30	Vendor model number of machine
	5	SerialNumber	varchar	30	Vendor serial number of machine
	6	CapacityQty	float	8	Maximum sustainable capacity for machine
	7	CapacityUOM	varchar	16	Unit of measure for capacity
	8	ContactInfo	varchar	30	Information on how to contact repair person
	9	Location	varchar	30	Name of location where machine is situated
	10	MachineType	varchar	30	Machine type associated with machine
	11	LockCount	smallint	2	Internal table usage count
			Table Total:	**316**	

MachineDataCollectLog		TransLogID	char	17	Internal transaction log index
	2	DSTemplateName	varchar	30	Data set template name
	3	DSTemplateVersion	varchar	8	Data set template version
	4	RTDataTableName	varchar	28	Runtime data set table name
	5	CollectTime	datetime	8	Time stamp when data sample collected
	6	SampleName	varchar	30	Name of data sample collected
	7	RouteName	varchar	30	Route name
	8	RouteVersion	varchar	8	Route version
	9	RouteStepName	varchar	30	Route step name
	10	OperationName	varchar	30	Operation name
	11	OperationVersion	varchar	8	Operation version
	12	Location	varchar	30	Location name
			Table Total:	**257**	

MachineDataSets	1	Machine	varchar	30	Machine name
	2	ElementIndex	int	4	Internal data set index
	3	DSTemplateName	varchar	30	Data set template name
	4	DSTemplateVersion	varchar	8	Data set template version
			Table Total:	**72**	

MachineMaintActivity-		TransLogID	char	17	Internal transaction log index
Log		MaintenanceID	varchar	30	Repair ID or machine task name

3	MaintenanceType	char	26	Activity (e.g., MACHINE_OPEN_REPAIR)
4	Description	varchar	80	Additional repair information
		Table Total:	**153**	

Table Name		Column Name	Data Type	Length	Description
MachineQuantity	1	LotID	varchar	40	Lot identifier
	2	MachQtyIndex	int	4	Internal index for machine quantity
	3	RouteName	varchar	30	Route name
	4	RouteVersion	varchar	8	Route version
	5	RouteStepName	varchar	30	Route step name
	6	Location	varchar	30	Name of location where lot is situated
	7	Machine	varchar	30	Machine name
	8	StartedQty	float	8	Lot primary quantity started
	9	StartedUOM	varchar	16	Unit of measure for primary quantity started
	10	LockCount	smallint	2	Internal table usage count
			Table Total:	**198**	

MachineStatus	1	Machine	varchar	30	Machine name
	2	MaintenanceStatus	char	21	Machine maintenance status: (READY / IN_PM / AWAITING_MANDATORY_PM / AWAITING_REPAIR / IN_REPAIR)
	3	PreviousMaintStatus	char	21	Most recent maintenance status of machine
	4	OperationalStatus	char	21	Machine operational status: (OFFLINE / IDLE / IN_USE)
	5	PreviousOperStatus	char	21	Most recent operational status of machine
	6	RepairID	varchar	30	Identifier for machine repair/maintenance event
	7	StartedQty	float	8	Lot quantity started
	8	StartedUOM	varchar	16	Unit of measure for quantity started
	9	CompletedLots	int	4	Number of lots completed on machine since last repair/maintenance
	10	CompletedQty	float	8	Quantity of material completed on machine since last repair/maintenance
	11	WhenStartedRunning	datetime	8	Time stamp when machine started running after last repair/maintenance
	12	TimeInUse	int	4	Amount of time machine in use since last repair/maintenance: (seconds)
	13	TimeStartedInMaintStatus	datetime	8	Time stamp when machine started in maintenance status
	14	TimeInPrevMaintStatus	int	4	Amount of time machine in previous maintenance status: (seconds)
	15	TimeStartedInOperStatus	datetime	8	Time stamp when machine started in operational status
	16	TimeInPrevOperStatus	int	4	Amount of time machine in previous operational status: (seconds)
	17	RunContinuous	char	5	Run machine in “continuous mode”–i.e., don’t stop machine when lot completes: (TRUE / FALSE)
	18	LockCount	smallint	2	Internal table usage count
			Table Total:	**223**	

MachineStatusLog	1	TransLogID	char	17	Internal transaction log index
	2	StatusType	char	15	Machine status type: (MAINTENANCE / OPERATIONAL)
	3	CurrentStatus	char	21	Current machine status
	4	PreviousStatus	char	21	Previous machine status
	5	TimeInPreviousStatus	int	4	Time in previous status: (seconds)
			Table Total:	**78**	

Table Name		Column Name	Data Type	Length	Description
MachineTaskStatus	1	Machine	varchar	30	Machine name
	2	TaskIndex	int	4	Internal machine task index
	3	MachineTask	varchar	30	Machine task name
	4	ExecutionState	char	15	Machine execution state: (RUNNING / NOT_RUNNING / IN_WARNING / EXPIRED)
	5	LastCompletedTime	datetime	8	Time stamp for last lot completion
	6	LastCompletedTimeInterval	int	4	Amount of time since last lot completion
	7	LastCompletedQtyUnits	float	8	Quantity of material in last completion
	8	LastCompletedLotUnits	int	4	Number of lots in last completion
			Table Total:	**103**	

MachineTaskWorkInst	1	MachineType	varchar	30	Machine type name
	2	ElementIndex	int	4	Internal machine task index
	3	ElementIndex2	int	4	Internal work instructions index
	4	Name	varchar	30	Work instructions name
	5	Version	varchar	8	Work instructions version
			Table Total:	**76**	

MachineType	1	MachineType	varchar	30	Machine type name
	2	Description	varchar	80	Descriptive comments
	3	Type	char	15	Machine type: (BATCH / SERIAL)
	4	CapacityUOM	varchar	16	Capacity units
	5	LockCount	smallint	2	Internal table usage count
			Table Total:	**143**	

MachineTypeFailure-Reason		MachineType	varchar	30	Machine type name
		ElementIndex	int	4	Internal machine failure reason index
	3	FailureReasonName	varchar	30	Failure reason name
	4	Description	varchar	80	Descriptive comments
	5	LockCount	smallint	2	Internal table usage count
			Table Total:	**146**	

MachineTypeFailure-Symptom		MachineType	varchar	30	Machine type name
		ElementIndex	int	4	Internal failure symptom index
	3	FailureSymptomName	varchar	30	Failure symptom name
	4	Description	varchar	80	Descriptive comments
	5	LockCount	smallint	2	Internal table usage count
			Table Total:	**146**	

Table Name		Column Name	Data Type	Length	Description
MachineTypeMachine-Task		MachineType	varchar	30	Machine type name
		ElementIndex	int	4	Internal machine task index
	3	MachineTask	varchar	30	Machine task name
	4	Description	varchar	80	Descriptive comments
	5	Type	char	15	Machine task type: (ON_DEMAND / PERIODIC)
	6	TakeMachineOffline	char	5	Take machine offline when task is invoked: (TRUE / FALSE)
	7	IntervalType	char	18	Wear measure type: (CLOCK_TIME / LOTS_COMPLETED / QUANTITY_COMPLETED / RUNNING_TIME)
	8	IntervalTime	int	4	Wear measure interval—elapsed time (clock or running): (seconds)
	9	QtyIntervalUnits	float	8	Wear measure interval—lot quantity completed
	10	LotIntervalUnits	int	4	Wear measure interval—number of lots completed
	11	WarningTimeInterval	int	4	Wear measure warning interval—elapsed time (clock or running): (seconds)
	12	QtyWarnUnitsInterval	float	8	Wear measure warning interval—lot quantity completed
	13	LotWarnUnitsInterval	int	4	Wear measure warning interval—number of lots completed
	14	LockCount	smallint	2	Internal table usage count
			Table Total:	**216**	

Material	1	MaterialName	varchar	30	Material name
	2	MaterialVersion	varchar	8	Material version
	3	Description	varchar	80	Descriptive comments
	4	Life	int	4	Material life span: (seconds)
	5	MaterialControl	char	15	Material control type: (LOT / BULK)
	6	IssueUOM	varchar	16	Unit of measure for issued material
	7	ReceiveUOM	varchar	16	Unit of measure for received material
	8	BOMMaterialName	varchar	30	Name of target material associated with bill of material
	9	BOMMaterialVersion	varchar	8	Version of target material associated with bill of material
	10	LockCount	smallint	2	Internal table usage count
			Table Total:	**209**	

Table Name		Column Name	Data Type	Length	Description
MaterialOverride	1	MaterialName	varchar	30	Material name
	2	MaterialVersion	varchar	8	Material version
	3	ElementIndex	int	4	Index for multiple overrides for this target material
	4	CustSpecName	varchar	30	Customer specification name
	5	CustSpecVersion	varchar	8	Customer specification version
	6	OperationName	varchar	30	Operation name
	7	OperationVersion	varchar	8	Operation version
	8	CycleTime	int	4	Operation duration: (seconds)
	9	Yield	real	4	Operation yield: (Absolute: +20% = 1.2)
	10	QueueTime	int	4	Maximum queue time for this operation: (seconds)
	11	CompleteOnly	char	5	Explicit lot start: (TRUE / FALSE)
	12	Overlapped	char	5	Allow overlapped lot processing: (TRUE / FALSE)
	13	UseCycleTime	char	5	Use this override's cycle time value: (TRUE / FALSE)
	14	UseYield	char	5	Use this override's yield value: (TRUE / FALSE)
	15	UseOverlapped	char	5	Use this override's overlapped value: (TRUE / FALSE)
	16	UseTrackingData	char	5	Use this override's queue time and complete-only values: (TRUE / FALSE)
	17	UserCertification	varchar	30	User certification name
			Table Total:	**190**	

MaterialStructure	1	MaterialName	varchar	30	Material name
	2	MaterialVersion	varchar	8	Material version
	3	RouteName	varchar	30	Route name
	4	RouteVersion	varchar	8	Route version
	5	LockCount	smallint	2	Internal table usage count
			Table Total:	**78**	

MergeLog	1	TransLogID	char	17	Internal transaction log index
	2	SecondLotID	varchar	40	Identifier of merged lot
	3	FromRouteName	varchar	30	Route name of merged lot
	4	FromRouteVersion	varchar	8	Route version of merged lot
	5	FromRouteStepName	varchar	30	Route step name of merged lot
	6	FromLocation	varchar	30	Location name of merged lot
			Table Total:	**155**	

Month	1	MonthID	char	17	Internal month identifier
	2	MonthOfYear	smallint	2	Sequential month in calendar year
	3	Name	varchar	30	Month name
	4	StartDate	datetime	8	First calendar day in month
	5	EndDate	datetime	8	Last calendar day in month
	6	LockCount	smallint	2	Internal table usage count
			Table Total:	**67**	

MonthsOfYear	1	YearID	char	17	Internal year identifier
	2	MonthIndex	int	4	Internal month index
	3	MonthID	char	17	Internal month identifier
			Table Total:	**38**	

Table Name		Column Name	Data Type	Length	Description
MoveLog	1	TransLogID	char	17	Internal transaction log index
	2	FromRouteName	varchar	30	Source route name
	3	FromRouteVersion	varchar	8	Source route version
	4	FromRouteStepName	varchar	30	Source route step name
	5	FromLocation	varchar	30	Source location
	6	FromOperationName	varchar	30	Source operation name
	7	FromOperationVersion	varchar	8	Source operation version
	8	QueueTime	int	4	Cumulative time lot queued: (seconds)
	9	OperationTime	int	4	Cumulative lot processing time: (seconds)
	10	DispositionCode	varchar	30	Lot condition code
	11	Machine	varchar	30	Machine name
	12	ForcedQty	float	8	Lot primary quantity forced to move
	13	ForcedUOM	varchar	16	Forced quantity's unit of measure
			Table Total:	**245**	

Operation	1	OperationName	varchar	30	Operation name
	2	OperationVersion	varchar	8	Operation version
	3	Description	varchar	80	Descriptive comments
	4	CycleTime	int	4	Operation duration: (seconds)
	5	Yield	real	4	Operation yield: (Absolute: +20% = 1.2)
	6	QueueTime	int	4	Maximum queue time for this operation: (seconds)
	7	CompleteOnly	char	5	Explicit lot start: (TRUE / FALSE)
	8	Overlapped	char	5	Allow overlapped lot processing: (TRUE / FALSE)
	9	BitmapFile	varchar	80	Operation symbol filename
	10	UserCertification	varchar	30	User certification name
	11	LockCount	smallint	2	Internal table usage count
			Table Total:	**252**	

OperationDataSets	1	OperationName	varchar	30	Operation name
	2	OperationVersion	varchar	8	Operation version
	3	ElementIndex	int	4	Internal data set index
	4	DSTemplateName	varchar	30	Data set template name
	5	DSTemplateVersion	varchar	8	Data set template version
			Table Total:	**80**	

OperationMachine	1	OperationName	varchar	30	Operation name
	2	OperationVersion	varchar	8	Operation version
	3	ElementIndex	int	4	Internal machine index
	4	Machine	varchar	30	Machine name
	5	LockCount	smallint	2	Internal table usage count
			Table Total:	**74**	

OperationWorkInst	1	OperationName	varchar	30	Operation name
	2	OperationVersion	varchar	8	Operation version
	3	ElementIndex	int	4	Internal work instructions index
	4	WorkInstName	varchar	30	Work instructions name
	5	WorkInstVersion	varchar	8	Work instructions version
			Table Total:	**80**	

OperSetpointTemplates	OperationName	varchar	30	Operation name
2	OperationVersion	varchar	8	Operation version
3	ElementIndex	int	4	Internal work instructions index
4	SPTemplateName	varchar	30	Setpoint template name
5	SPTemplateVersion	varchar	8	Setpoint template version
		Table Total:	**80**	

Table Name		Column Name	Data Type	Length	Description
OvrDataSets	1	MaterialName	varchar	30	Material name
	2	MaterialVersion	varchar	8	Material version
	3	ElementIndex	int	4	Internal data set index
	4	DataSetNum	int	4	Data set override number
	5	DSTemplateName	varchar	30	Name of data set template being overridden
	6	DSTemplateVersion	varchar	8	Version of data set template being overridden
			Table Total:	**84**	

OvrOperationMachine	1	MaterialName	varchar	30	Material name
	2	MaterialVersion	varchar	8	Material version
	3	ElementIndex	int	4	Internal machine index
	4	MachineNum	int	4	Machine override number
	5	Machine	varchar	30	Name of machine being overridden
	6	LockCount	smallint	2	Internal table usage count
			Table Total:	**78**	

OvrSetpointTemplates	1	MaterialName	varchar	30	Material name
	2	MaterialVersion	varchar	8	Material version
	3	ElementIndex	int	4	Internal work instructions index
	4	SetpointTemplateNum	int	4	Setpoint template override number
	5	SPTemplateName	varchar	30	Name of setpoint template being overridden
	6	SPTemplateVersion	varchar	8	Setpoint template version
			Table Total:	**84**	

OvrWorkInstructions	1	MaterialName	varchar	30	Material name
	2	MaterialVersion	varchar	8	Material version
	3	ElementIndex	int	4	Internal work instructions index
	4	WorkInstNum	int	4	Work instructions override number
	5	WorkInstName	varchar	30	Name of work instructions being overridden
	6	WorkInstVersion	varchar	8	Work instructions version
			Table Total:	**84**	

QuarantineLog	1	TransLogID	char	17	Internal transaction log index
	2	OldQuarantineStatus	varchar	16	Most recent lot quarantine status: (ACTIVE / QUARANTINED)
	3	NewQuarantineStatus	varchar	16	New lot quarantine status: (ACTIVE / QUARANTINED)
			Table Total:	**49**	

Table Name		Column Name	Data Type	Length	Description
ResourceBaseLog	1	BaseLogID	char	17	Internal resource log index
	2	TimeLogged	datetime	8	Transaction time stamp (might be backdated)
	3	ActualTime	datetime	8	Actual time stamp for transaction
	4	UserName	varchar	30	Name of logged on user
	5	TransactionName	char	26	Transaction name: (MACHINE_BATCH_LOAD / MACHINE_BATCH_UNLOAD / MACHINE_BRING_ONLINE / MACHINE_CLOSE_REPAIR / MACHINE_COLLECT_DATA / MACHINE_COMPLETE / MACHINE_OPEN_REPAIR / MACHINE_REPAIR_DIAGNOSIS / MACHINE_REPAIR_PERSON / MACHINE_REPAIR_REASON / MACHINE_REPAIR_STATUS / MACHINE_REPAIR_SYMPTOM / MACHINE_START / MACHINE_START_REPAIR / MACHINE_STOP_REPAIR / MACHINE_TAKE_OFFLINE / MACHINE_TASK_COMPLETE / MACHINE_TASK_START / USER_CLOCK_IN / USER_CLOCK_OUT)
	6	CalendarName	varchar	30	Calendar name in use when transaction occurred
	7	CalendarDay	smallint	2	Sequential day in calendar month
	8	CalendarMonth	smallint	2	Sequential month in calendar year
	9	CalendarYear	smallint	2	Sequential year in calendar
	10	CalendarWorkWeek	smallint	2	Sequential work week in calendar
	11	CalendarDayOfWeek	smallint	2	Sequential day in calendar week (value is **zero**-based)
	12	CalendarShift	varchar	30	Calendar shift name
	13	CommentText	varchar	255	Transaction comments
	14	Undone	char	5	Transaction has been undone:
	15	CompoundTrans	char	5	Transaction is part of a compound transaction: (TRUE / FALSE)
	16	TransLogIDType	smallint	2	Internal transaction log type
	17	TransLogID	char	17	Internal transaction log index
	18	ResourceName	varchar	30	Resource name: (<machine name> or <user name>)
			Table Total:	**473**	

ResourceUndoLog	1	TransLogID	char	17	Internal transaction log index
	2	UndoTransID	char	17	Internal index of undone transaction
			Table Total:	**34**	

Route	1	RouteName	varchar	30	Route name
	2	RouteVersion	varchar	8	Route version
	3	Description	varchar	80	Descriptive comments
	4	FirstStep	varchar	30	Name of first step on route
	5	NextStructStep	int	4	Internal index for identifying next structure step to create
	6	DefaultDispCode	varchar	30	Default disposition code for route
	7	GraphicalData	int	4	Internal route drawing information
	8	NumberSteps	int	4	Number of steps on route's main path
	9	LockCount	smallint	2	Internal table usage count
			Table Total:	**192**	

Table Name		Column Name	Data Type	Length	Description
RoutePath	1	RouteName	varchar	30	Route name
	2	RouteVersion	varchar	8	Route version
	3	PathIndex	int	4	Internal path index
	4	MainExit	char	5	This path for this disposition code is along main path: (TRUE / FALSE)
	5	DispositionCode	varchar	30	Lot condition code
	6	FromStepIndex	int	4	Internal source route step index
	7	ToStepIndex	int	4	Internal destination route step index
			Table Total:	**85**	

RouteStep	1	RouteName	varchar	30	Route name
	2	RouteVersion	varchar	8	Route version
	3	StepIndex	int	4	Internal route step index
	4	NumberPaths	int	4	Number of paths from this step
	5	StepName	varchar	30	Route step name
	6	StepInternalIndex	int	4	Internal route step ID
	7	StepStructStep	int	4	Internal material structure step index
	8	StepOperationName	varchar	30	Name of operation at route step
	9	StepOperationVersion	varchar	8	Version of operation at route step
	10	StepUserData	int	4	Internal format information
	11	StepSeqNumber	int	4	Internal step sequence index
			Table Total:	**130**	

SecGroupToPriv	1	Name	varchar	30	Security group name
	2	Privilege	varchar	30	Internal security privilege name
			Table Total:	**60**	

SecurityGroup	1	Name	varchar	30	Security group name
	2	Description	varchar	80	Descriptive comments
	3	LockCount	smallint	2	Internal table usage count
	4	SecurityMask	binary	18	Internal security bit mask
			Table Total:	**130**	

SecurityMasterTag	1	Name	varchar	30	Internal security master tag name
	2	Tag	int	4	Internal update counter
	3	LockCount	smallint	2	Internal table usage count
			Table Total:	**36**	

SecurityPrivilege	1	Name	varchar	30	Security group name
	2	Description	varchar	80	Internal security group description
	3	Code	smallint	2	Security privilege code
	4	LockCount	smallint	2	Internal table usage count
			Table Total:	**114**	

SecUserAddedPriv	1	Name	varchar	30	User name
	2	AddPrivilege	varchar	30	Add privilege code
			Table Total:	**60**	

SecUserDeletPriv	1	Name	varchar	30	User name
	2	DeletePrivilege	varchar	30	Remove privilege code
			Table Total:	**60**	

SecUserToGroup	1	Name	varchar	30	User name
	2	SecurityGroup	varchar	30	Security group name
			Table Total:	**60**	

Table Name		Column Name	Data Type	Length	Description
SerialNumber	1	SerialNumber	varchar	30	Serial number
	2	LotID	varchar	40	Lot identifier
	3	SublotIndex	int	4	Internal sublot index
	4	WorkState	char	15	Process status of serialized unit: (QUEUED / IN_PROCESS / CLOSED)
	5	Machine	varchar	30	Machine name
			Table Total:	**119**	

SerialNumberLog	1	BaseLogID	char	17	Internal serial number log index
	2	SerialIndex	int	4	Internal serial index
	3	SerialNumber	varchar	30	Serial number of lot unit involved in transaction
	4	Machine	varchar	30	Name of machine that processed the serialized unit
			Table Total:	**81**	

SetpointItem	1	SPTemplateName	varchar	30	Setpoint template name
	2	SetpointTemplateVersion	varchar	8	Setpoint template version
	3	ElementIndex	int	4	Internal item index
	4	Name	varchar	30	Setpoint item name
	5	Description	varchar	80	Setpoint item description
	6	DataType	char	30	Item's data type: (ALPHANUMERIC / ANALOG / BOOLEAN / DATE / DATETIME / INTEGER / SELECTION / TIME / TIMEINTERVAL)
	7	RealValue	real	4	Setpoint item real value
	8	DoubleValue	float	8	Setpoint item double value
	9	IntegerValue	int	4	Setpoint item integer value
	10	StringValue	varchar	255	Comma-delineated string
	11	BooleanValue	smallint	4	Setpoint item Boolean value
	12	RealHighLimit	real	4	Setpoint item high limit
	13	RealLowLimit	real	4	Setpoint item low limit
	14	DoubleHighLimit	float	8	Setpoint item high high limit
	15	DoubleLowLimit	float	8	Setpoint item low low limit
	16	IntegerHighLimit	int	4	Setpoint item integer high limit
	17	IntegerLowLimit	int	4	Setpoint item integer low limit
	18	StringWidth	int	4	String width
	19	StringSelections	varchar	255	Comma-delineated string
			Table Total:	**748**	

SetpointTemplate	1	SPTemplateName	varchar	30	Setpoint template name
	2	SetpointTemplateVersion	varchar	8	Setpoint template version
	3	Description	varchar	80	Description
	4	ParentSPTemplateName	varchar	30	Parent setpoint template name
	5	ParentSPTemplate Version	varchar	8	Parent setpoint template version
			Table Total:	**152**	

Shift	1	ShiftID	char	17	Internal shift identifier
	2	Name	varchar	30	Shift name
	3	StartTime	datetime	8	Start time: (time of day)
	4	Length	int	4	Shift length: (seconds)
	5	DefaultFlag	char	5	This is the default shift: (TRUE / FALSE)
	6	Range	smallint	2	Shift range: (0 = starts yesterday 1 = starts and ends today 2 = ends tomorrow)
	7	LockCount	smallint	2	Internal table usage count
			Table Total:	**68**	

Table Name		Column Name	Data Type	Length	Description
ShiftsOfCalendar	1	Name	varchar	30	Calendar name
	2	ShiftIndex	int	4	Internal shift index
	3	ShiftID	char	17	Internal shift identifier
			Table Total:	**51**	

ShiftsOfDay	1	MonthID	char	17	Internal month identifier
	2	DayIndex	int	4	Internal day index
	3	ShiftIndex	int	4	Internal shift index
	4	ShiftID	char	17	Internal shift identifier
			Table Total:	**42**	

ShipLog	1	TransLogID	char	17	Internal transaction log index
	2	Customer	varchar	30	Customer name
			Table Total:	**47**	

SplitLog	1	TransLogID	char	17	Internal transaction log index
	2	SecondLotID	varchar	40	Identifier of lot split off
			Table Total:	**57**	

StartLog	1	TransLogID	char	17	Internal transaction log index
	2	QueueTime	int	4	Cumulative time lot queued: (seconds)
	3	Machine	varchar	30	Machine name
			Table Total:	**51**	

StepDispCodes	1	RouteName	varchar	30	Route name
	2	RouteVersion	varchar	8	Route version
	3	StepIndex	int	4	Internal route step index
	4	CodeIndex	int	4	Internal disposition code index
	5	DispositionCode	varchar	30	Lot condition code
			Table Total:	**76**	

StepInput	1	MaterialName	varchar	30	Target material name
	2	MaterialVersion	varchar	8	Material version
	3	RouteName	varchar	30	Route name
	4	RouteVersion	varchar	8	Route version
	5	ElementIndex	int	4	Internal step input index
	6	StepInNum	int	4	Internal material structure step number
	7	StepMaterialName	varchar	30	Input material name
	8	StepMaterialVersion	varchar	8	Input material version
	9	StandardQty	float	8	Default quantity consumed
	10	StandardUOM	varchar	16	Default quantity unit of measure
	11	UpperTolerance	float	8	Maximum tolerable quantity deviation: (Absolute: +20% = 1.2)
	12	LowerTolerance	float	8	Minimum tolerable quantity deviation: (Absolute: -20% = 0.8)
	13	AdjPrimaryQty	char	5	Reserved for future use
	14	Scaled	char	5	Scale quantity of input material to target material quantity: (TRUE / FALSE)
	15	Location	varchar	30	Source location name
			Table Total:	**202**	

Table Name		Column Name	Data Type	Length	Description
StepOutput	1	MaterialName	varchar	30	Material name
	2	MaterialVersion	varchar	8	Material version
	3	RouteName	varchar	30	Route name
	4	RouteVersion	varchar	8	Route version
	5	ElementIndex	int	4	Internal step output index
	6	StepOutNum	int	4	Internal material structure step number
	7	DispositionCode	varchar	30	Lot condition code
	8	StepMaterialName	varchar	30	Output material name
	9	StepMaterialVersion	varchar	8	Output material version
	10	StandardQty	float	8	Default lot quantity produced at step
	11	StandardUOM	varchar	16	Default quantity unit of measure
	12	UpperTolerance	float	8	Maximum tolerable quantity deviation: (Absolute: +20% = 1.2)
	13	LowerTolerance	float	8	Minimum tolerable quantity deviation: (Absolute: -20% = 0.8)
	14	AdjPrimaryQty	char	5	Decrease primary quantity by material output: (TRUE / FALSE)
	15	Scaled	char	5	Scale quantity of output material to target material quantity: (TRUE / FALSE)
	16	ToRouteName	varchar	30	Destination route name
	17	ToRouteVersion	varchar	8	Destination route version
	18	ToRouteStep	varchar	30	Destination route step name
	19	ToLocation	varchar	30	Destination location name
			Table Total:	**300**	

StepSetpoint	1	MaterialName	varchar	30	Material name
	2	MaterialVersion	varchar	8	Material version
	3	RouteName	varchar	30	Route name
	4	RouteVersion	varchar	8	Route version
	5	ElementIndex	int	4	Internal step output index
	6	SetpointNumber	int	4	Internal step number
	7	SPTemplateName	varchar	30	Setpoint template name
	8	SPTemplateVersion	varchar	8	Setpoint template version
	9	Name	varchar	30	Setpoint item name
	10	RealValue	real	4	Setpoint item real value
	11	DoubleValue	float	8	Setpoint item double value
	12	IntegerValue	int	4	Setpoint item integer value
	13	StringValue	varchar	255	Comma-delineated string
	14	BooleanValue	smallint	4	Setpoint item Boolean value
			Table Total:	**437**	

StructureStep	1	MaterialName	varchar	30	Material name
	2	MaterialVersion	varchar	8	Material version
	3	RouteName	varchar	30	Route name
	4	RouteVersion	varchar	8	Route version
	5	ElementIndex	int	4	Internal structure step index
	6	StepNumber	int	4	Internal material structure step number
			Table Total:	**84**	

Table Name		Column Name	Data Type	Length	Description
Sublot	1	LotID	varchar	40	Lot identifier
	2	SublotIndex	int	4	Internal sublot index
	3	RouteName	varchar	30	Route name
	4	RouteVersion	varchar	8	Route version
	5	RouteStepName	varchar	30	Route step name
	6	OperationName	varchar	30	Operation name
	7	OperationVersion	varchar	8	Operation version
	8	WorkState	char	15	Sublot process status: (QUEUED / IN_PROCESS / CLOSED)
	9	QueuedQty	float	8	Lot quantity queued or closed
	10	QueuedUOM	varchar	16	Unit of measure for quantity queued or closed
	11	StartedQty	float	8	Lot quantity started
	12	StartedUOM	varchar	16	Unit of measure for quantity started
	13	TimeQueued	datetime	8	Lot queued time stamp
	14	TimeStarted	datetime	8	Lot started time stamp
	15	DataCollectDone	char	5	Data collection has been done: (TRUE / FALSE)
	16	ConsumptionDone	char	5	Consumption has been done: (TRUE / FALSE)
	17	SerializedState	char	15	Sublot serialized state: (UNSERIALIZED / PARTIAL / SERIALIZED)
	18	Location	varchar	30	Location name
	19	CriticalRatio	real	4	Sublot's critical ratio: (time remaining until due date ÷ processing time left)
	20	EstimatedCompletion	datetime	8	Date and time sublot is estimated to complete, based on main path cycle times
	21	MachineQty	float	8	Total quantity started on all machines for sublot
	22	LockCount	smallint	2	Internal table usage count
			Table Total:	**306**	

SublotCollection 1		LotID	varchar	40	Lot identifier
	2	SublotIndex	int	4	Internal sublot index
	3	ItemIndex	int	4	Internal index of data sets collected for this sublot
	4	RouteName	varchar	30	Route name
	5	RouteVersion	varchar	8	Route version
	6	RouteStepName	varchar	30	Route step name
	7	Location	varchar	30	Location name
	8	DSTemplateName	varchar	30	Data set template name
	9	DSTemplateVersion	varchar	8	Data set template version
	10	RTDataTableName	varchar	28	Runtime data table name
	11	RequiredSampleCount	int	4	Number of required samples collected to date
	12	SampleName	varchar	30	Name of data sample collected
			Table Total:	**246**	

Table Name		Column Name	Data Type	Length	Description
SublotConsumption	1	LotID	varchar	40	Lot identifier
	2	SublotIndex	int	4	Internal sublot index
	3	ItemIndex	int	4	Index of material consumed by this sublot
	4	RouteName	varchar	30	Route name
	5	RouteVersion	varchar	8	Route version
	6	RouteStepName	varchar	30	Route step name
	7	Location	varchar	30	Location name
	8	ConsumedMaterialName	varchar	30	Name of material consumed
	9	ConsumedMaterialVersion	varchar	8	Version of material consumed
	10	SubstituteMaterialName	varchar	30	Name of substitute material consumed
	11	SubstituteMaterialVersion	varchar	8	Version of substitute material consumed
	12	ConsumedQty	float	8	Quantity of material consumed
	13	ConsumedUOM	varchar	16	Consumed material's unit of measure
	14	SubstituteConsumedQty	float	8	Quantity of substitute material consumed
	15	SubstituteConsumedUOM	varchar	16	Consumed substitute material's unit of measure
	16	ConsumerSerialNumber	varchar	30	Serial number of lot unit that consumed the material
	17	ConsumerMachine	varchar	30	Name of machine that consumed material
			Table Total:	**330**	

UniqueId	1	Name	varchar	32	Object name
	2	NextId	varchar	17	Next identifier to generate in sequence
	3	DBVersion	smallint	4	Internal database version
			Table Total:	**53**	

UserCertification	1	UserCertification	varchar	30	User certification name
	2	Description	varchar	80	Descriptive comments
	3	ExpirationInterval	int	4	Interval certification is valid: (seconds)
	4	LockCount	smallint	2	Internal table usage count
			Table Total:	**116**	

UserLog		TransLogID	char	17	Internal transaction log index
			Table Total:	**17**	

Users	1	Name	varchar	30	User name
	2	CommonName	varchar	30	Reserved for future use
	3	Password	binary	38	Encrypted password
	4	SecurityMask	binary	18	Internal security bit mask
	5	SecurityValue	int	4	Internal security tag
	6	LockCount	smallint	2	Internal table usage count
			Table Total:	**122**	

Table Name		Column Name	Data Type	Length	Description
WorkInstructions	1	WorkInstName	varchar	30	Work instructions name
	2	WorkInstVersion	varchar	8	Work instructions version
	3	Description	varchar	80	Descriptive comments
	4	Label	varchar	30	Runtime label for work instructions
	5	LastChanged	datetime	8	Time stamp of latest change
	6	DaysToNotify	int	4	Days to show work instructions have changed
	7	IsTextInternal	char	5	Text is stored internally: (TRUE / FALSE)
	8	ExternFile	varchar	80	Name of file containing work instructions
	9	LockCount	smallint	2	Internal table usage count
			Table Total:	**247**	

WorkInstText	1	WorkInstName	varchar	30	Work instructions name
	2	WorkInstVersion	varchar	8	Work instructions version
	3	ElementIndex	int	4	Internal index for work instructions
	4	TextInfo	image	16	Binary data of internal work instructions text
			Table Total:	**58**	

Year	1	YearID	char	17	Internal year identifier
	2	YearOfCalendar	smallint	2	Year in calendar (e.g., 1996)
	3	StartDate	datetime	8	Start date in calendar (e.g., Jan 1, 1996 12:00:00 AM)
	4	EndDate	datetime	8	End date in calendar (e.g., Dec 31, 1998 12:00:00 AM)
	5	LockCount	smallint	2	Internal table usage count
			Table Total:	**37**	

YearsOfCalendar	1	Name	varchar	30	Calendar name
	2	YearIndex	int	4	Internal year index
	3	YearID	char	17	Internal year identifier
			Table Total:	**51**	

Table Name	Column Name	Data Type	Length	Description
<Database table name> (*user-defined runtime data collection table*)	SampleName	varchar	30	Name of data sample collected
	CollectTime	datetime	8	Time stamp when data sample collected
	LotID	varchar	40	Lot identifier
	Machine	varchar	30	Machine name
	Location	varchar	30	Location name
	OperationName	varchar	30	Operation name
	OperationVersion	varchar	8	Operation version
	UserName	varchar	30	Name of logged-in user at time of collection
	Superseded	char	5	Data superseded: (TRUE / FALSE)
	TimeSuperseded	datetime	8	Time stamp when data superseded
	SampleIndex	int	4	Internal index for sample name
	<Data item 1 name>	*user defined*		Value of data collected for first user-defined data item in template
	<Data item 2 name>	*user defined*		Value of data collected for second user-defined data item in template
	◈			
	<Data item *n* name>	*user defined*		Value of data collected for last user-defined data item in template

CHAPTER 6

InTrack Data Dictionary

Column Name	Table Name	Data Type	Length	Description
ActualTime	LotBaseLog	datetime	8	Actual time stamp for transaction
	ResourceBaseLog	datetime	8	Actual time stamp for transaction
AddPrivilege	SecUserAddedPriv	varchar	30	Add privilege code
AdjPrimaryQty	BOMItems	char	5	Decrease primary quantity by material output: (TRUE / FALSE)
	StepInput	char	5	Reserved for future use
	StepOutput	char	5	Decrease primary quantity by material output: (TRUE / FALSE)
AdjustedAttrib	AdjustLog	char	15	Attribute type: (DUE_DATE / ID / MATERIAL / PRI_Q_QTY / PRI_S_QTY / PRIORITY / ROUTE_STEP / SEC_QTY / UDA)
Attribute	AdjustLog	varchar	30	Attribute name: (predefined or <column name>)
BaseClassId	ClassDefinition	smallint	2	Internal base class identifier
BaseLogID	LotBaseLog	char	17	Internal lot log index
	ResourceBaseLog	char	17	Internal resource log index
	SerialNumberLog	char	17	Internal serial number log index
BitmapFile	Operation	varchar	80	Operation symbol filename
BooleanValue	SetpointItem	smallint	4	Boolean value of setpoint limit
	StepSetpoint	smallint	4	Boolean value of setpoint limit
BOMMaterialName	BillOfMaterial	varchar	30	Name of target material associated with bill of material
	BOMItems	varchar	30	Name of target material associated with bill of material
	BOMSubItems	varchar	30	Name of target material associated with bill of material
	Material	varchar	30	Name of target material associated with bill of material
BOMMaterialVersion	BillOfMaterial	varchar	8	Version of target material associated with bill of material
	BOMItems	varchar	8	Version of target material associated with bill of material
	BOMSubItems	varchar	8	Version of target material associated with bill of material
	Material	varchar	8	Version of target material associated with bill of material
CalendarDay	LotBaseLog	smallint	2	Sequential day in calendar month
	ResourceBaseLog	smallint	2	Sequential day in calendar month
CalendarDayOfWeek	LotBaseLog	smallint	2	Sequential day in calendar week
	ResourceBaseLog	smallint	2	Sequential day in calendar week
CalendarMonth	LotBaseLog	smallint	2	Sequential month in calendar year
	ResourceBaseLog	smallint	2	Sequential month in calendar year
CalendarName	LotBaseLog	varchar	30	Calendar name in use when transaction occurred
	ResourceBaseLog	varchar	30	Calendar name in use when transaction occurred

Column Name	Table Name	Data Type	Length	Description
CalendarShift	LotBaseLog	varchar	30	Calendar shift name
	ResourceBaseLog	varchar	30	Calendar shift name
CalendarWorkWeek	LotBaseLog	smallint	2	Sequential work week in calendar
	ResourceBaseLog	smallint	2	Sequential work week in calendar
CalendarYear	LotBaseLog	smallint	2	Sequential year in calendar
	ResourceBaseLog	smallint	2	Sequential year in calendar
CapacityQty	Machine	float	8	Maximum sustainable capacity for machine
CapacityUOM	Machine	varchar	16	Unit of measure for capacity
	MachineType	varchar	16	Capacity units
ClassAttribute	ClassDefinition	image	16	Internal class attribute
Code	SecurityPrivilege	smallint	2	Security privilege code
CodeIndex	StepDispCodes	int	4	Internal disposition code index
CollectTime	DataCollectLog	datetime	8	Time stamp when sample collected
	DSBaseRTValues	datetime	8	Time stamp when sample collected
	MachineDataCollectLog	datetime	8	Time stamp when sample collected
	user-defined runtime data table	datetime	8	Time stamp when sample collected
CommentText	LotBaseLog	varchar	255	Transaction comments
	ResourceBaseLog	varchar	255	Transaction comments
CommonName	Users	varchar	30	Reserved for future use
CompletedLots	MachineStatus	int	4	Number of lots completed on machine since last repair/maintenance
CompletedQty	MachineStatus	float	8	Quantity of material completed on machine since last repair/maintenance
CompleteOnly	MaterialOverride	char	5	Explicit lot start: (TRUE / FALSE)
	Operation	char	5	Explicit lot start: (TRUE / FALSE)
CompletionType	CompleteLog	char	15	Type of COMPLETE transaction: (ON_ROUTE / OUTPUT / SCRAP)
CompoundTrans	LotBaseLog	char	5	Transaction is part of a compound transaction: (TRUE / FALSE)
	ResourceBaseLog	char	5	Transaction is part of a compound transaction: (TRUE / FALSE)
ConsumedLotID	ConsumeLog	varchar	40	Lot identifier
ConsumedMaterialName	CloseLogConsumption	varchar	30	Name of material consumed
	CompleteLogConsumption	varchar	30	Name of material consumed
	ConsumeLog	varchar	30	Name of material consumed
	LotConsumption	varchar	30	Name of material consumed
	SublotConsumption	varchar	30	Name of material consumed
ConsumedMaterialVersion	CloseLogConsumption	varchar	8	Version of material consumed
	CompleteLogConsumption	varchar	8	Version of material consumed
	ConsumeLog	varchar	8	Version of material consumed
	LotConsumption	varchar	8	Version of material consumed
	SublotConsumption	varchar	8	Version of material consumed
ConsumedQty	CloseLogConsumption	float	8	Quantity of material consumed
	CompleteLogConsumption	float	8	Quantity of material consumed
	LotConsumption	float	8	Quantity of material consumed
	SublotConsumption	float	8	Quantity of material consumed
ConsumedUOM	CloseLogConsumption	varchar	16	Consumed material's unit of measure
	LotConsumption	varchar	16	Consumed material's unit of measure
	CompleteLogConsumption	varchar	16	Consumed material's unit of measure
	SublotConsumption	varchar	16	Consumed material's unit of measure
ConsumeMachine	CloseLogConsumption	varchar	30	Machine that consumed material
	CompleteLogConsumption	varchar	30	Machine that consumed material
	ConsumeLog	varchar	30	Machine that consumed material
	LotConsumption	varchar	30	Machine that consumed material
	SublotConsumption	varchar	30	Machine that consumed material

Column Name	Table Name	Data Type	Length	Description
ConsumerSerialNumber	CloseLogConsumption	varchar	30	Serial number of lot unit that consumed the material
	CompleteLogConsumption	varchar	30	Serial number of lot unit that consumed the material
	ConsumeLog	varchar	30	Serial number of lot unit that consumed the material
	DisassembleLog	varchar	30	Serial number of lot unit that consumed the material before disassembly and from which components are being disassembled
	SublotConsumption	varchar	30	Serial number of lot unit that consumed the material
ConsumptionDone	Sublot	char	5	Consumption has been done: (TRUE / FALSE)
ConsumptionLocation	ConsumeLog	varchar	30	Location name
ConsumptionRatio	ConsumeLog	float	8	Ratio of quantity consumed while in the process status – if process status is QUEUED: (quantity consumed ÷ quantity queued) – if process status is IN_PROCESS: (quantity consumed ÷ quantity in process)
ConsumptionRouteName	ConsumeLog	varchar	30	Route name
ConsumptionRouteStep	ConsumeLog	varchar	30	Route step name
ConsumptionRouteVersion	ConsumeLog	varchar	8	Route version
ContactInfo	Machine	varchar	30	Information on how to contact repair person
CreateDate	Lot	datetime	8	Lot creation time stamp
CreatedSublot	DisassembleLog	char	5	New destination sublot created: (TRUE / FALSE)
CreateTime	ClassDictionary	datetime	8	Lot creation time stamp
CriticalRatio	Sublot	real	4	Sublot's critical ratio: (time remaining until due date ÷ processing time left)
CurrentStatus	MachineStatusLog	char	21	Current machine status
Customer	ShipLog	varchar	30	Customer name
CustSpecName	CustomerSpecification	varchar	30	Customer specification name
	Lot	varchar	30	Customer specification name
	MaterialOverride	varchar	30	Customer specification name
CustSpecVersion	CustomerSpecification	varchar	8	Customer specification version
	Lot	varchar	8	Customer specification version
	MaterialOverride	varchar	8	Customer specification version
CycleTime	MaterialOverride	int	4	Operation duration: (seconds)
	Operation	int	4	Operation duration: (seconds)
DataCollectDone	Sublot	char	5	Data collection has been done: (TRUE / FALSE)
DataItemBooleanData	DSDataItemHeader	smallint	2	Default Boolean value
DataItemClassID	DSDataItemHeader	smallint	2	Internal UDA object type identifier
DataItemDataReqFlag	DSDataItemHeader	char	5	Data collection required: (TRUE / FALSE)
DataItemDataType	DSDataItemHeader	char	29	Item's data type: (ALPHANUMERIC / ANALOG / BOOLEAN / DATE / DATETIME / INTEGER / SELECTION / TIME / TIMEINTERVAL)
DataItemDateData	DSDataItemHeader	datetime	8	Default date value
DataItemDateTimeData	DSDataItemHeader	datetime	8	Default date-time value

Column Name	Table Name	Data Type	Length	Description
DataItemDescription	DSDataItemHeader	varchar	80	Runtime SQL table column description
DataItemDoubl	DSDataItemHeader	float	4	Double value
DataItemFloat	DSDataItemHeader	real	4	Default float value
DataItemFormat	DSDataItemHeader	varchar	254	Comma-separated set of string selections
DataItemHighFlag	DSDataItemHeader	char	5	Enable lot hold for high value: (TRUE / FALSE)
DataItemHighHighFlag	DSDataItemHeader	char	5	Enable lot hold for high-high value: (TRUE / FALSE)
DataItemIndex	DSDataItemHeader	int	4	Internal item index
DataItemIntegerData	DSDataItemHeader	int	4	Default integer value
DataItemIntHHighLimit	DSDataItemHeader	int	4	High-high integer value range limit
DataItemIntHighLimit	DSDataItemHeader	int	4	High integer value range limit
DataItemIntInstrHigh	DSDataItemHeader	int	4	Maximum integer value for instrument
DataItemIntInstrLow	DSDataItemHeader	int	4	Minimum integer value for instrument
DataItemIntLLowLimit	DSDataItemHeader	int	4	Low-low integer value range limit
DataItemIntLowLimit	DSDataItemHeader	int	4	Low integer value range limit
DataItemLabel	DSDataItemHeader	varchar	16	Data item's runtime interface label
DataItemLimChkFlag	DSDataItemHeader	char	5	Perform limit checking on this item: (TRUE / FALSE)
DataItemLowFlag	DSDataItemHeader	char	5	Enable lot hold for low value: (TRUE / FALSE)
DataItemLowLowFlag	DSDataItemHeader	char	5	Enable lot hold for low-low value: (TRUE / FALSE)
DataItemName	DSDataItemHeader	varchar	30	Runtime SQL table column name
DataItemRealHHighLimit	DSDataItemHeader	real	4	High-high real value range limit
DataItemRealHighLimit	DSDataItemHeader	real	4	High real value range limit
DataItemRealInstrHigh	DSDataItemHeader	real	4	Maximum real value for instrument
DataItemRealInstrLow	DSDataItemHeader	real	4	Minimum real value for instrument
DataItemRealLLowLimit	DSDataItemHeader	real	4	Low-low real value range limit
DataItemRealLowLimit	DSDataItemHeader	real	4	Low real value range limit
DataItemString	DSDataItemHeader	varchar	80	Default string value
DataItemStringWidth	DSDataItemHeader	int	4	Width of the string column
DataItemTimeData	DSDataItemHeader	datetime	8	Default time value
DataItemTimeIntervalData	DSDataItemHeader	int	4	Default time interval value
DataSetNum	OvrDataSets	int	4	Data set override number
DataType	AdjustLog	smallint	2	Data type identifier
	SetpointItem	char	30	Data type identifier
DayIndex	DaysOfMonth	int	4	Internal day of month index (relative to specified first day of month)
	ShiftsOfDay	int	4	Internal day index
DayOfMonth	DaysOfMonth	smallint	2	Day of month
DaysToNotify	WorkInstructions	int	4	Days to show work instructions have changed
DBVersion	UniqueId	int	4	Internal database version
DefaultDispCode	Route	varchar	30	Default disposition code for route
DefaultFlag	Shift	char	5	This is the default shift: (TRUE / FALSE)
DefaultShiftID	DefaultShifts	char	17	Internal default shift identifier
DefaultShiftIndex	DefaultShifts	int	4	Internal default shift index
DeletePrivilege	SecUserDeletPriv	varchar	30	Remove privilege code
Description	Calendar	varchar	80	Descriptive comments
	ClassDefinition	varchar	32	Internal class description
	Configuration	varchar	80	Configuration option description
	CustomerSpecification	varchar	80	Descriptive comments
	DispositionCode	varchar	80	Descriptive comments
	DSTemplate	varchar	80	Descriptive comments

Column Name	Table Name	Data Type	Length	Description
Description (*continued*)	Location	varchar	80	Descriptive comments
	Machine	varchar	80	Descriptive comments
	MachineMaintActivityLog	varchar	80	Additional repair information
	MachineType	varchar	80	Descriptive comments
	MachineTypeFailureReason	varchar	80	Descriptive comments
	MachineTypeFailureSymptom	varchar	80	Descriptive comments
	MachineTypeMachineTask	varchar	80	Descriptive comments
	Material	varchar	80	Descriptive comments
	Operation	varchar	80	Descriptive comments
	Route	varchar	80	Descriptive comments
	SecurityGroup	varchar	80	Descriptive comments
	SecurityPrivilege	varchar	80	Internal security group description
	SetpointItem	varchar	80	Descriptive comments
	SetpointTemplate	varchar	80	Descriptive comments
	UserCertification	varchar	80	Descriptive comments
	WorkInstructions	varchar	80	Descriptive comments
DispositionCode	CloseLog	varchar	30	Lot condition code
	CompleteLog	varchar	30	Lot condition code
	DispositionCode	varchar	30	Lot condition code
	MoveLog	varchar	30	Lot condition code
	RoutePath	varchar	30	Lot condition code
	StepDispCodes	varchar	30	Lot condition code
	StepOutput	varchar	30	Lot condition code
DispositionType	DispositionCode	char	15	Code condition type: (GOOD / REJECT / REWORK / SCRAP)
DoubleHighLimit	SetpointItem	float	8	Setpoint double high limit
DoubleLowLimit	SetpointItem	float	8	Setpoint double low limit
DoubleValue	Configuration	float	8	Configuration option real value
	SetpointItem	float	8	Setpoint double value
	StepSetpoint	float	4	Setpoint double value
DSTemplateName	CloseLogCollection	varchar	30	Data set template name
	CompleteLogCollection	varchar	30	Name of data set template used to collect data
	DataCollectLog	varchar	30	Data set template name
	DSDataItemHeader	varchar	30	Data set template name
	DSSampleHeader	varchar	30	Data set template name
	DSTemplate	varchar	30	Data set template name
	MachineDataCollectLog	varchar	30	Data set template name
	MachineDataSets	varchar	30	Data set template name
	OperationDataSets	varchar	30	Data set template name
	OvrDataSets	varchar	30	Name of data set template being overridden
	SublotCollection	varchar	30	Data set template name
DSTemplateVersion	CloseLogCollection	varchar	8	Data set template version
	CompleteLogCollection	varchar	8	Version of data set template used to collect data
	DataCollectLog	varchar	8	Data set template version
	DSDataItemHeader	varchar	8	Data set template version
	DSSampleHeader	varchar	8	Data set template version
	DSTemplate	varchar	8	Data set template version
	MachineDataCollectLog	varchar	8	Data set template version
	MachineDataSets	varchar	8	Data set template version
	OperationDataSets	varchar	8	Data set template version
	OvrDataSets	varchar	8	Version of data set template being overridden
	SublotCollection	varchar	8	Data set template version
DueDate	CreateLog	datetime	8	Lot due date
	Lot	datetime	8	Lot due date

Column Name	Table Name	Data Type	Length	Description
ElementIndex	IssuedUserCert	int	4	Internal user certification index
	MachineDataSets	int	4	Internal data set index
	MachineTaskWorkInst	int	4	Internal machine task index
	MachineTypeFailureReason	int	4	Internal machine failure reason index
	MachineTypeFailureSymptom	int	4	Internal failure symptom index
	MachineTypeMachineTask	int	4	Internal machine task index
	MaterialOverride	int	4	Index for multiple overrides for this target material
	OperationDataSets	int	4	Internal data set index
	OperationMachine	int	4	Internal machine index
	OperationWorkInst	int	4	Internal work instructions index
	OperSetpointTemplates	int	4	
	OvrDataSets	int	4	Internal data set index
	OvrOperationMachine	int	4	Internal machine index
	OvrSetpointTemplates	int	4	Internal setpoint item index
	OvrWorkInstructions	int	4	Internal work instructions index
	SetpointItem	int	4	Internal setpoint item index
	StepInput	int	4	Internal step input index
	StepOutput	int	4	Internal step output index
	StepSetpoint	int	4	Internal setpoint item index
	StructureStep	int	4	Internal structure step index
	WorkInstText	int	4	Internal index for work instructions
ElementIndex2	MachineTaskWorkInst	int	4	Internal work instructions index
EndDate	Month	datetime	8	Last calendar day in month
	Year	datetime	8	End date in calendar (e.g., Dec 31, 1998 12:00:00 AM)
EstimatedCompletion	Sublot	datetime	8	Date and time sublot is estimated to complete, based on main path cycle times
ExecutionState	MachineTaskStatus	char	15	Machine execution state: (RUNNING / NOT_RUNNING / IN_WARNING / EXPIRED)
ExpirationDate	Lot	datetime	8	Lot expiration time stamp: (creation date-time + lifetime of material)
ExpirationInterval	UserCertification	int	4	Interval certification is valid: (seconds)
ExternFile	WorkInstructions	varchar	80	Name of file containing work instructions
FailureReasonName	MachineTypeFailureReason	varchar	30	Failure reason name
FailureSymptomName	MachineTypeFailureSymptom	varchar	30	Failure symptom name
FirstDayOfWeek	Calendar	smallint	2	Specified first day of calendar week
FirstMonthOfYear	Calendar	smallint	2	Specified first month of calendar year
FirstStep	Route	varchar	30	Name of first step on route
ForcedConsumption	ConsumeLog	char	5	Material consumption forced at this route step: (TRUE / FALSE)
ForcedQty	CloseLog	float	8	Lot primary quantity forced to close
	CompleteLog	float	8	Lot primary quantity forced to complete
	MoveLog	float	8	Lot primary quantity forced to move
ForcedUOM	CloseLog	varchar	16	Forced quantity's unit of measure
	CompleteLog	varchar	16	Forced quantity's unit of measure
	MoveLog	varchar	16	Forced quantity's unit of measure
FromLocation	CloseLog	varchar	30	From location
	CompleteLog	varchar	30	From location
	MergeLog	varchar	30	Location name of merged lot
	MoveLog	varchar	30	Source location
FromOperationName	CloseLog	varchar	30	From operation name
	CompleteLog	varchar	30	From operation name
	MoveLog	varchar	30	Source operation name

Column Name	Table Name	Data Type	Length	Description
FromOperationVersion	CloseLog	varchar	8	From operation version
	CompleteLog	varchar	8	From operation version
	MoveLog	varchar	8	Source operation version
FromRouteName	CloseLog	varchar	30	From route name
	CompleteLog	varchar	30	From route name
	MergeLog	varchar	30	Route name of merged lot
	MoveLog	varchar	30	Source route name
FromRouteStepName	CloseLog	varchar	30	From route step name
	CompleteLog	varchar	30	From route step name
	MergeLog	varchar	30	Route step name of merged lot
	MoveLog	varchar	30	Source route step name
FromRouteVersion	CloseLog	varchar	8	From route version
	CompleteLog	varchar	8	From route version
	MergeLog	varchar	8	Route version of merged lot
	MoveLog	varchar	8	Source route version
FromStepIndex	RoutePath	int	4	Internal source route step index
GradedLotID	CompleteLog	varchar	40	Graded lot identifier
GradedMaterialName	CompleteLog	varchar	30	Graded material name
GradedMaterialVersion	CompleteLog	varchar	8	Graded material version
GraphicalData	Route	int	4	Internal route drawing information
HoldStatus	Lot	char	15	Lot hold status: (RELEASED / ON_HOLD)
InstClassId	ClassDefinition	smallint	2	Internal instance class identifier
InstClassName	ClassDefinition	varchar	32	Internal instance class name
InstMethod	ClassDefinition	int	4	Internal instance method
IntegerHighLimit	SetpointItem	int	4	Setpoint integer high limit
IntegerLowLimit	SetpointItem	int	4	Setpoint integer low limit
IntegerValue	Configuration	float	8	Configuration option integer value
	SetpointItem	int	4	Setpoint integer value
	StepSetpoint	int	4	Setpoint integer value
IntervalTime	MachineTypeMachineTask	int	4	Wear measure interval— elapsed time (clock or running): (seconds)
IntervalType	MachineTypeMachineTask	char	18	Wear measure type: (CLOCK_TIME / LOTS_COMPLETED / QUANTITY_COMPLETED / RUNNING_TIME)
InvCustSpecName	CloseLog	varchar	30	Name of customer specification for inventory lot
InvCustSpecVersion	CloseLog	varchar	8	Version of customer specification for inventory lot
InventoryLocation	CloseLog	varchar	30	Destination inventory location name
InventoryLotID	CloseLog	varchar	40	Destination inventory lot identifier
InvLotCreated	CloseLog	char	5	Inventory lot created: (TRUE / FALSE)
IssueTime	IssuedUserCert	datetime	8	User certification issuance time stamp
IssueUOM	Material	varchar	16	Unit of measure for issued material
IsTextInternal	WorkInstructions	char	5	Text is stored internally: (TRUE / FALSE)
ItemID	BOMItems	int	4	Sort sequence number of input or output material
	BOMSubItems	int	4	Sort sequence number of input material
ItemIndex	BOMItems	int	4	Internal item index
	BOMSubItems	int	4	Internal input item index
	CloseLogCollection	int	4	Internal index of data sets collected for this lot
	CloseLogConsumption	int	4	Internal index of material consumed for this lot
	CompleteLogCollection	int	4	Internal index of data sets collected for this lot

Column Name	Table Name	Data Type	Length	Description
ItemIndex (*continued*)	CompleteLogConsumption	int	4	Internal index of material consumed for this lot
	SublotCollection	int	4	Internal index of data sets collected for this sublot
	SublotConsumption	int	4	Index of material consumed by this sublot
ItemType	BOMItems	char	15	Type of material: (INPUT / OUTPUT)
	BOMSubItems	char	15	Type of material: (INPUT)
KeyMethod	ClassDefinition	int	4	Internal key method
Label	WorkInstructions	varchar	30	Work instructions runtime label
LastChanged	WorkInstructions	datetime	8	Time stamp of latest change
LastCompletedLotUnits	MachineTaskStatus	int	4	Number of lots in last completion
LastCompletedQtyUnits	MachineTaskStatus	float	8	Quantity of material in last completion
LastCompletedTime	MachineTaskStatus	datetime	8	Time stamp for last lot completion
LastCompletedTimeInterval	MachineTaskStatus	int	4	Amount of time since last lot completion
LastSublotIndex	Lot	int	4	Internal index for identifying last sublot deleted from database
Length	Shift	int	4	Shift length: (seconds)
Life	Material	int	4	Material life span: (seconds)
Location	CloseLogCollection	varchar	30	Location name
	CloseLogConsumption	varchar	30	Location name
	CompleteLogCollection	varchar	30	Name of location in which data was collected
	CompleteLogConsumption	varchar	30	Location name
	DSBaseRTValues	varchar	30	Location name
	Location	varchar	30	Location name
	LotBaseLog	varchar	30	Location name
	Machine	varchar	30	Name of location where machine is situated
	MachineDataCollectLog	varchar	30	Location name
	MachineQuantity	varchar	30	Name of location where lot is situated
	StepInput	varchar	30	Source location name
	Sublot	varchar	30	Location name
	SublotCollection	varchar	30	Location name
	SublotConsumption	varchar	30	Location name
	user-defined runtime data table	varchar	30	Location name
LockCount	BillOfMaterial	smallint	2	Internal table usage count
	Calendar	smallint	2	Internal table usage count
	ClassDefinition	smallint	2	Internal table usage count
	ClassDictionary	smallint	2	Internal table usage count
	Configuration	smallint	2	Internal table usage count
	CustomerSpecification	smallint	2	Internal table usage count
	DispositionCode	smallint	2	Internal table usage count
	DSTemplate	smallint	2	Internal table usage count
	IssuedUserCert	smallint	2	Internal table usage count
	Location	smallint	2	Internal table usage count
	Lot	smallint	2	Internal table usage count
	Machine	smallint	2	Internal table usage count
	MachineQuantity	smallint	2	Internal table usage count
	MachineStatus	smallint	2	Internal table usage count
	MachineType	smallint	2	Internal table usage count
	MachineTypeFailureReason	smallint	2	Internal table usage count
	MachineTypeFailureSymptom	smallint	2	Internal table usage count

Column Name	Table Name	Data Type	Length	Description
LockCount (*continued*)	MachineTypeMachineTask	smallint	2	Internal table usage count
	Material	smallint	2	Internal table usage count
	MaterialStructure	smallint	2	Internal table usage count
	Month	smallint	2	Internal table usage count
	Operation	smallint	2	Internal table usage count
	OperationMachine	smallint	2	Internal table usage count
	OvrOperationMachine	smallint	2	Internal table usage count
	Route	smallint	2	Internal table usage count
	SecurityGroup	smallint	2	Internal table usage count
	SecurityMasterTag	smallint	2	Internal table usage count
	SecurityPrivilege	smallint	2	Internal table usage count
	Shift	smallint	2	Internal table usage count
	Sublot	smallint	2	Internal table usage count
	UserCertification	smallint	2	Internal table usage count
	Users	smallint	2	Internal table usage count
	WorkInstructions	smallint	2	Internal table usage count
	Year	smallint	2	Internal table usage count
LotID	DSBaseRTValues	varchar	40	Lot identifier
	Lot	varchar	40	Lot identifier
	LotBaseLog	varchar	40	Lot identifier
	MachineQuantity	varchar	40	Lot identifier
	SerialNumber	varchar	40	Lot identifier
	Sublot	varchar	40	Lot identifier
	SublotCollection	varchar	40	Lot identifier
	SublotConsumption	varchar	40	Lot identifier
	user-defined runtime data table	varchar	40	Lot identifier
LotIntervalUnits	MachineTypeMachineTask	int	4	Wear measure interval— number of lots completed
LotSuffix	Lot	smallint	2	Name suffix for new lot this lot generates
LotWarnUnitsInterval	MachineTypeMachineTask	int	4	Wear measure warning interval— number of lots completed
LowerTolerance	BOMItems	float	8	Minimum tolerable quantity deviation: (Absolute: -20% = 0.8)
	BOMSubItems	float	8	Minimum tolerable quantity deviation: (Absolute: -20% = 0.8)
	StepInput	float	8	Minimum tolerable quantity deviation: (Absolute: -20% = 0.8)
	StepOutput	float	8	Minimum tolerable quantity deviation: (Absolute: -20% = 0.8)
Machine	CloseLog	varchar	30	Machine name
	CompleteLog	varchar	30	Machine name
	DataCollectLog	varchar	30	Machine name
	DSBaseRTValues	varchar	30	Machine name
	Machine	varchar	30	Machine name
	MachineDataSets	varchar	30	Machine name
	MachineQuantity	varchar	30	Machine name
	MachineStatus	varchar	30	Machine name
	MachineTaskStatus	varchar	30	Machine name
	MoveLog	varchar	30	Machine name
	OperationMachine	varchar	30	Machine name
	OvrOperationMachine	varchar	30	Name of machine being overridden
	SerialNumber	varchar	30	Machine name

Column Name	Table Name	Data Type	Length	Description
Machine (*continued*)	SerialNumberLog	varchar	30	Name of machine that processed the serialized unit
	StartLog	varchar	30	Machine name
	user-defined runtime data table	varchar	30	Machine name
MachineNum	OvrOperationMachine	int	4	Machine override number
MachineQty	Sublot	float	8	Total quantity started on all machines for sublot
MachineTask	MachineTaskStatus	varchar	30	Machine task name
	MachineTypeMachineTask	varchar	30	Machine task name
MachineType	Machine	varchar	30	Machine type associated with machine
	MachineTaskWorkInst	varchar	30	Machine type name
	MachineType	varchar	30	Machine type name
	MachineTypeFailureReason	varchar	30	Machine type name
	MachineTypeFailureSymptom	varchar	30	Machine type name
	MachineTypeMachineTask	varchar	30	Machine type name
MachQtyIndex	MachineQuantity	int	4	Internal index for machine quantity
MainExit	RoutePath	char	5	This path for this disposition code is along main path: (TRUE / FALSE)
MaintenanceID	MachineMaintActivityLog	varchar	30	Repair ID or machine task name
MaintenanceStatus	MachineStatus	char	21	Machine maintenance status: (READY / IN_PM / AWAITING_MANDATORY_PM / AWAITING_REPAIR / IN_REPAIR)
MaintenanceType	MachineMaintActivityLog	char	26	Activity (e.g., MACHINE_OPEN_REPAIR)
MaterialControl	Material	char	15	Material control type: (LOT / BULK)
MaterialName	BOMItems	varchar	30	Material name
	BOMSubItems	varchar	30	Name of substitute material
	Lot	varchar	30	Material name
	LotBaseLog	varchar	30	Material name
	Material	varchar	30	Material name
	MaterialOverride	varchar	30	Material name
	MaterialStructure	varchar	30	Material name
	OvrDataSets	varchar	30	Material name
	OvrOperationMachine	varchar	30	Material name
	OvrSetpointTemplates	varchar	30	Material name
	OvrWorkInstructions	varchar	30	Material name
	StepInput	varchar	30	Target material name
	StepOutput	varchar	30	Material name
	StepSetpoint	varchar	30	Material name
	StructureStep	varchar	30	Material name
MaterialVersion	BOMItems	varchar	8	Material version
	BOMSubItems	varchar	8	Version of substitute material
	Lot	varchar	8	Material version
	LotBaseLog	varchar	8	Material version
	Material	varchar	8	Material version
	MaterialOverride	varchar	8	Material version
	MaterialStructure	varchar	8	Material version
	OvrDataSets	varchar	8	Material version
	OvrOperationMachine	varchar	8	Material version
	OvrSetpointTemplates	varchar	8	Material version
	OvrWorkInstructions	varchar	8	Material version
	StepInput	varchar	8	Material version
	StepOutput	varchar	8	Material version
	StepSetpoint	varchar	8	Material version
	StructureStep	varchar	8	Material version

Column Name	Table Name	Data Type	Length	Description
MaxSubstituteRatio	BOMSubItems	float	8	Maximum allowable quantity of substitute material expressed as a ratio: (substitute quantity ÷ input quantity)
Model	Machine	varchar	30	Vendor model number of machine
MonthID	DaysOfMonth	char	17	Internal month identifier
	Month	char	17	Internal month identifier
	MonthsOfYear	char	17	Internal month identifier
	ShiftsOfDay	char	17	Internal month identifier
MonthIndex	MonthsOfYear	int	4	Internal month index
MonthOfYear	Month	smallint	2	Sequential month in calendar year
Name	Calendar	varchar	30	Calendar name
	ClassDictionary	varchar	32	Object name
	DefaultShifts	varchar	30	Calendar name
	IssuedUserCert	varchar	30	User name
	MachineTaskWorkInst	varchar	30	Work instructions name
	Month	varchar	30	Month name
	SecGroupToPriv	varchar	30	Security group name
	SecurityGroup	varchar	30	Security group name
	SecurityMasterTag	varchar	30	Internal security master tag name
	SecurityPrivilege	varchar	30	Security group name
	SecUserAddedPriv	varchar	30	User name
	SecUserDeletPriv	varchar	30	User name
	SecUserToGroup	varchar	30	User name
	SetpointItem	varchar	30	Setpoint item name
	Shift	varchar	30	Shift name
	ShiftsOfCalendar	varchar	30	Calendar name
	StepSetpoint	varchar	30	Setpoint name
	UniqueId	varchar	32	Object name
	Users	varchar	30	User name
	YearsOfCalendar	varchar	30	Calendar name
NewDateAttrib	AdjustLog	datetime	8	Attribute date value after adjustment
NewDateTimeAttrib	AdjustLog	datetime	8	Attribute date-time value after adjustment
NewDoubleAttrib	AdjustLog	float	8	Attribute double value after adjustment
NewHoldStatus	HoldLog	varchar	16	Current lot hold status: (RELEASED / ON_HOLD)
NewIntAttrib	AdjustLog	int	4	Attribute integer value after adjustment
NewQuarantineStatus	QuarantineLog	varchar	16	New lot quarantine status: (ACTIVE / QUARANTINED)
NewRealAttrib	AdjustLog	real	4	Attribute real value after adjustment
NewStringAttrib	AdjustLog	varchar	80	Attribute string value after adjustment
NewTimeAttrib	AdjustLog	datetime	8	Attribute time value after adjustment
NewTimeInterAttrib	AdjustLog	int	4	Attribute time interval value after adjustment
NextClassId	ClassDictionary	smallint	2	Next class index
NextId	UniqueId	char	17	Next unique index
NextStructStep	Route	int	4	Internal index for identifying next structure step to create
NextSublotIndex	Lot	int	4	Internal index for identifying next sublot to create
NumberPaths	RouteStep	int	4	Number of paths from this step
NumberSteps	Route	int	4	Number of steps on route's main path
OldDateAttrib	AdjustLog	datetime	8	Attribute date value before adjustment
OldDateTimeAttrib	AdjustLog	datetime	8	Attribute date-time value before adjustment
OldDoubleAttrib	AdjustLog	float	8	Attribute double value before adjustment

Column Name	Table Name	Data Type	Length	Description
OldHoldStatus	HoldLog	varchar	16	Most recent lot hold status: (RELEASED / ON_HOLD)
OldIntAttrib	AdjustLog	int	4	Attribute integer value before adjustment
OldQuarantineStatus	QuarantineLog	varchar	16	Most recent lot quarantine status: (ACTIVE / QUARANTINED)
OldRealAttrib	AdjustLog	real	4	Attribute real value before adjustment
OldStringAttrib	AdjustLog	varchar	80	Attribute string value before adjustment
OldTimeAttrib	AdjustLog	datetime	8	Attribute time value before adjustment
OldTimeInterAttrib	AdjustLog	int	4	Attribute time interval value before adjustment
OperationalStatus	MachineStatus	char	21	Machine operational status: (OFFLINE / IDLE / IN_USE)
OperationName	DSBaseRTValues	varchar	30	Operation name
	LotBaseLog	varchar	30	Operation name
	MachineDataCollectLog	varchar	30	Operation name
	MaterialOverride	varchar	30	Operation name
	Operation	varchar	30	Operation name
	OperationDataSets	varchar	30	Operation name
	OperationMachine	varchar	30	Operation name
	OperationWorkInst	varchar	30	Operation name
	OperSetpointTemplates	varchar	30	Operation name
	Sublot	varchar	30	Operation name
	user-defined runtime data table	varchar	30	Operation name
OperationTime	CloseLog	int	4	Cumulative lot processing time: (seconds)
	CompleteLog	int	4	Cumulative lot processing time: (seconds)
	MoveLog	int	4	Cumulative lot processing time: (seconds)
OperationVersion	DSBaseRTValues	varchar	8	Operation version
	LotBaseLog	varchar	8	Operation version
	MachineDataCollectLog	varchar	8	Operation version
	MaterialOverride	varchar	8	Operation version
	Operation	varchar	8	Operation version
	OperationDataSets	varchar	8	Operation version
	OperationMachine	varchar	8	Operation version
	OperationWorkInst	varchar	8	Operation version
	OperSetpointTemplate	varchar	8	Operation version
	Sublot	varchar	8	Operation version
	user-defined runtime data table	varchar	8	Operation version
OptionName	Configuration	varchar	30	Configuration option name
Overlapped	MaterialOverride	char	5	Allow overlapped lot processing: (TRUE / FALSE)
	Operation	char	5	Allow overlapped lot processing: (TRUE / FALSE)
ParentDSTemplateName	DSTemplate	varchar	30	Parent data set template name
ParentDSTemplateVersion	DSTemplate	varchar	8	Parent data set template version
ParentSPTemplateName	SetpointTemplate	varchar	30	Parent setpoint template name
ParentSPTemplateVersion	SetpointTemplate	varchar	8	Parent setpoint template version
ParentLotID	CreateLog	varchar	40	Source lot identifier
Password	Users	binary	38	Encrypted password
PathIndex	RoutePath	int	4	Internal path index
PeriodicHolidayMap	Calendar	smallint	2	Internal mask denoting periodic holidays in week
PreviousMaintStatus	MachineStatus	char	21	Most recent maintenance status of machine
PreviousOperStatus	MachineStatus	char	21	Most recent operational status of machine
PreviousStatus	MachineStatus	char	21	Previous machine status
PrimaryQty	Lot	float	8	Lot primary quantity
	LotBaseLog	float	8	Lot primary quantity
PrimaryTransQty	LotBaseLog	float	8	Lot primary quantity for transaction

Column Name	Table Name	Data Type	Length	Description
PrimaryTransUOM	LotBaseLog	varchar	16	Unit of measure for lot primary transaction quantity
PrimaryUOM	Lot	varchar	16	Unit of measure for lot primary quantity
	LotBaseLog	varchar	16	Unit of measure for lot primary quantity
Priority	CreateLog	smallint	2	Lot priority value
	Lot	smallint	2	Lot priority value
Privilege	SecGroupToPriv	varchar	30	Internal security privilege name
QtyIntervalUnits	MachineTypeMachineTask	float	8	Wear measure interval— lot quantity completed
QtyWarnUnitsInterval	MachineTypeMachineTask	float	8	Wear measure warning interval— lot quantity completed
QuarantineStatus	CloseLog	char	15	Lot quarantine status: (ACTIVE / QUARANTINED)
	Lot	char	15	Lot quarantine status: (ACTIVE / QUARANTINED)
QueuedQty	Sublot	float	8	Lot quantity queued
QueuedUOM	Sublot	varchar	16	Unit of measure for quantity queued
QueueTime	CloseLog	int	4	Cumulative time lot queued: (seconds)
	CompleteLog	int	4	Cumulative time lot queued: (seconds)
	MaterialOverride	int	4	Maximum queue time for this operation: (seconds)
	MoveLog	int	4	Cumulative time lot queued: (seconds)
	Operation	int	4	Maximum queue time for this operation: (seconds)
	StartLog	int	4	Cumulative time lot queued: (seconds)
Range	Shift	smallint	2	Shift range: (0 = starts yesterday 1 = starts and ends today 2 = ends tomorrow)
RealHighLimit	SetpointItem	real	4	Setpoint real high limit
RealLowLimit	SetpointItem	real	4	Setpoint real low limit
RealValue	SetpointItem	real	4	Real setpoint value
	StepSetpoint	real	4	Real setpoint value
ReceiveUOM	Material	varchar	16	Unit of measure for received material
RepairID	MachineStatus	varchar	30	Identifier for machine repair/maintenance event
RequiredSampleCount	CloseLogCollection	int	4	Number of required samples collected to date
	CompleteLogCollection	int	4	Number of required samples collected to date
	DSTemplate	int	4	Number of required samples
	SublotCollection	int	4	Number of required samples collected to date
RequiredSampleRatio	DSTemplate	real	4	Number of required samples, expressed as ratio to material quantity: (0.0 = *n* = 1.0)
ResourceName	ResourceBaseLog	varchar	30	Resource name: (<machine name> or <user name>)
RouteName	CloseLogCollection	varchar	30	Route name
	CloseLogConsumption	varchar	30	Route name
	CompleteLogCollection	varchar	30	Name of route on which data was collected
	CompleteLogConsumption	varchar	30	Route name
	LotBaseLog	varchar	30	Route name
	MachineDataCollectLog	varchar	30	Route name
	MachineQuantity	varchar	30	Route name
	MaterialStructure	varchar	30	Route name
	Route	varchar	30	Route name
	RoutePath	varchar	30	Route name

Column Name	Table Name	Data Type	Length	Description
RouteName (*continued*)	RouteStep	varchar	30	Route name
	StepDispCodes	varchar	30	Route name
	StepInput	varchar	30	Route name
	StepOutput	varchar	30	Route name
	StepSetpoint	varchar	30	Route name
	StructureStep	varchar	30	Route name
	Sublot	varchar	30	Route name
	SublotCollection	varchar	30	Route name
	SublotConsumption	varchar	30	Route name
RouteStepName	CloseLogCollection	varchar	30	Route step name
	CloseLogConsumption	varchar	30	Route step name
	CompleteLogCollection	varchar	30	Name of route step at which data was collected
	CompleteLogConsumption	varchar	30	Route step name
	LotBaseLog	varchar	30	Route step name
	MachineDataCollectLog	varchar	30	Route step name
	MachineQuantity	varchar	30	Route step name
	Sublot	varchar	30	Route step name
	SublotCollection	varchar	30	Route step name
	SublotConsumption	varchar	30	Route step name
RouteVersion	CloseLogCollection	varchar	8	Route version
	CloseLogConsumption	varchar	8	Route version
	CompleteLogCollection	varchar	8	Version of route on which data was collected
	CompleteLogConsumption	varchar	8	Route version
	LotBaseLog	varchar	8	Route version
	MachineDataCollectLog	varchar	8	Route version
	MachineQuantity	varchar	8	Route version
	MaterialStructure	varchar	8	Route version
	Route	varchar	8	Route version
	RoutePath	varchar	8	Route version
	RouteStep	varchar	8	Route version
	StepDispCodes	varchar	8	Route version
	StepInput	varchar	8	Route version
	StepOutput	varchar	8	Route version
	StepSetpoint	varchar	8	Route version
	StructureStep	varchar	8	Route version
	Sublot	varchar	8	Route version
	SublotCollection	varchar	8	Route version
	SublotConsumption	varchar	8	Route version
RTDataTableClassID	DSTemplate	smallint	2	Internal runtime data set table class
RTDataTableName	CloseLogCollection	varchar	28	Runtime data set table name
	CompleteLogCollection	varchar	28	Runtime data set table name
	DataCollectLog	varchar	28	Runtime data set table name
	DSTemplate	varchar	28	Runtime data set table name
	MachineDataCollectLog	varchar	28	Runtime data set table name
	SublotCollection	varchar	28	Runtime data table name
RunContinuous	MachineStatus	char	5	Run machine in "continuous mode"—i.e., don't stop machine when lot completes: (TRUE / FALSE)
SampleDescription	DSSampleHeader	varchar	80	Sample descriptive comments
SampleIndex	DSBaseRTValues	int	4	Internal index for sample name
	DSSampleHeader	int	4	Internal index for sample name
	user-defined runtime data table	int	4	Internal index for sample name

Column Name	Table Name	Data Type	Length	Description
SampleName	DSBaseRTValues	varchar	30	Name of sample collected
	CloseLogCollection	varchar	30	Name of data sample collected
	CompleteLogCollection	varchar	30	Name of data sample collected
	DataCollectLog	varchar	30	Name of data sample collected
	DSSampleHeader	varchar	30	Data collection sample name
	MachineDataCollectLog	varchar	30	Name of data sample collected
	SublotCollection	varchar	30	Name of data sample collected
	user-defined runtime data table	varchar	30	Name of data sample collected
SampleNameSeed	DSTemplate	varchar	30	Prefix for sample name generator
SamplingPlan	DSTemplate	char	15	Sampling plan: (VARIABLE / FIXED)
Scaled	BOMItems	char	5	Enable quantity scaling: (TRUE / FALSE)
	StepInput	char	5	Scale quantity of input material to target material quantity: (TRUE / FALSE)
	StepOutput	char	5	Scale quantity of output material to target material quantity: (TRUE / FALSE)
SecondaryQty	Lot	float	8	Lot secondary quantity
	LotBaseLog	float	8	Lot secondary quantity
SecondaryTransQty	LotBaseLog	float	8	Lot secondary quantity for transaction
SecondaryTransUOM	LotBaseLog	varchar	16	Unit of measure for lot secondary transaction quantity
SecondaryUOM	Lot	varchar	16	Unit of measure for lot secondary quantity
	LotBaseLog	varchar	16	Unit of measure for lot secondary quantity
SecondLotID	MergeLog	varchar	40	Identifier of merged lot
	SplitLog	varchar	40	Identifier of lot split off
SecurityGroup	SecUserToGroup	varchar	30	Security group name
SecurityMask	SecurityGroup	binary	18	Internal security bit mask
	Users	binary	18	Internal security bit mask
SecurityValue	Users	int	4	Internal security tag
SerialIndex	SerialNumberLog	int	4	Internal serial index
SerializedState	Sublot	char	15	Sublot serialized state: (UNSERIALIZED / PARTIAL / SERIALIZED)
SerialNumber	Machine	varchar	30	Vendor serial number of machine
	SerialNumber	varchar	30	Serial number
	SerialNumberLog	varchar	30	Serial number of lot unit involved in transaction
SetpointNumber	StepSetpoint	int	4	Setpoint identifier
SetpointTemplateNumber	OvrSetpointTemplates	int	4	Setpoint template number for override
ShiftID	Shift	char	17	Internal shift identifier
	ShiftsOfCalendar	char	17	Internal shift identifier
	ShiftsOfDay	char	17	Internal shift identifier
ShiftIndex	ShiftsOfCalendar	int	4	Internal shift index
	ShiftsOfDay	int	4	Internal shift index
SPTemplateName	OperSetpointTemplates	varchar	30	Setpoint template name
	OvrSetpointTemplates	varchar	30	Setpoint template name
	SetpointItem	varchar	30	Setpoint template name
	SetpointTemplate	varchar	30	Setpoint template name
	StepSetpoint	varchar	30	Setpoint template name
SPTemplateVersion	OperSetpointTemplates	varchar	8	Setpoint template version
	OvrSetpointTemplates	varchar	8	Setpoint template version
	SetpointItem	varchar	8	Setpoint template version
	SetpointTemplate	varchar	8	Setpoint template version
	StepSetpoint	varchar	8	Setpoint template version
StandardQty	BillOfMaterial	float	8	Default quantity produced
	BOMItems	float	8	Default quantity consumed
	BOMSubItems	float	8	Default quantity consumed
	StepInput	float	8	Default quantity consumed
	StepOutput	float	8	Default lot quantity produced at step
StandardUOM	BillOfMaterial	varchar	16	Default quantity unit of measure
	BOMItems	varchar	16	Default quantity unit of measure
	BOMSubItems	varchar	16	Default quantity unit of measure
	StepInput	varchar	16	Default quantity unit of measure
	StepOutput	varchar	16	Default quantity unit of measure

Column Name	Table Name	Data Type	Length	Description
StartDate	Month	datetime	8	First calendar day in month
	Year	datetime	8	Start date in calendar (e.g., Jan 1, 1996 12:00:00 AM)
StartedQty	MachineQuantity	float	8	Lot primary quantity started
	MachineStatus	float	8	Lot quantity started
	Sublot	float	8	Lot quantity started
StartedUOM	MachineQuantity	varchar	16	Unit of measure for primary quantity started
	MachineStatus	varchar	16	Unit of measure for quantity started
	Sublot	varchar	16	Unit of measure for quantity started
StartTime	Shift	datetime	8	Start time: (time of day)
StatusType	MachineStatusLog	char	15	Machine status type: (MAINTENANCE / OPERATIONAL)
StepIndex	RouteStep	int	4	Internal route step index
	StepDispCodes	int	4	Internal route step index
StepInNum	StepInput	int	4	Internal material structure step number
StepInternalIndex	RouteStep	int	4	Internal route step ID
StepMaterialName	StepInput	varchar	30	Input material name
	StepOutput	varchar	30	Output material name
StepMaterialVersion	StepInput	varchar	8	Input material version
	StepOutput	varchar	8	Output material version
StepName	RouteStep	varchar	30	Route step name
StepNumber	StructureStep	int	4	Internal material structure step number
StepOperationName	RouteStep	varchar	30	Name of operation at route step
StepOperationVersion	RouteStep	varchar	8	Version of operation at route step
StepOutNum	StepOutput	int	4	Internal material structure step number
StepSeqNumber	RouteStep	int	4	Internal step sequence index
StepStructStep	RouteStep	int	4	Internal material structure step index
StepUserData	RouteStep	int	4	Internal format information
StringSelections	SetpointItem	varchar	255	Setpoint string
StringValue	Configuration	float	8	Configuration option string value
	SetpointItem	varchar	255	Setpoint string value
	StepSetpoint	varchar	255	Setpoint string value
StringWidth	SetpointItem	int	4	Setpoint string width
SubItemIndex	BOMSubItems	int	4	Internal substitute item index
SublotIndex	AdjustLog	int	4	Internal index of sublot adjusted
	SerialNumber	int	4	Internal sublot index
	Sublot	int	4	Internal sublot index
	SublotCollection	int	4	Internal sublot index
	SublotConsumption	int	4	Internal sublot index
SubstituteConsumedQty	CloseLogConsumption	float	8	Quantity of substitute material consumed
	CompleteLogConsumption	float	8	Quantity of substitute material consumed
	LotConsumption	float	8	Quantity of substitute material consumed
	SublotConsumption	float	8	Quantity of substitute material consumed
SubstituteConsumedUOM	CloseLogConsumption	varchar	16	Substitute material's unit of measure
	CompleteLogConsumption	varchar	16	Substitute material's unit of measure
	LotConsumption	varchar	16	Substitute material's unit of measure
	SublotConsumption	varchar	16	Substitute material's unit of measure
SubstituteMaterialName	CloseLogConsumption	varchar	30	Name of substitute material consumed
	CompleteLogConsumption	varchar	30	Name of substitute material consumed
	LotConsumption	varchar	30	Name of substitute material consumed
	SublotConsumption	varchar	30	Name of substitute material consumed
SubstituteMaterialVersion	CloseLogConsumption	varchar	8	Version of substitute material consumed
	CompleteLogConsumption	varchar	8	Version of substitute material consumed
	LotConsumption	varchar	8	Version of substitute material consumed
	SublotConsumption	varchar	8	Version of substitute material consumed
Superseded	DSBaseRTValues	char	5	Data superseded
	user-defined runtime data table	char	5	Data superseded: (TRUE / FALSE)
TableSpace	ClassDefinition	varchar	32	Internal tablespace
Tag	SecurityMasterTag	int	4	Internal update counter

Column Name	Table Name	Data Type	Length	Description
TakeMachineOffline	MachineTypeMachineTask	char	5	Take machine offline when task is invoked: (TRUE / FALSE)
TaskIndex	MachineTaskStatus	int	4	Internal machine task index
TextInfo	WorkInstText	image	16	Binary data of internal work instructions text
TimeInPreviousStatus	MachineStatusLog	int	4	Time in previous status: (seconds)
TimeInPrevMaintStatus	MachineStatus	int	4	Amount of time in previous maintenance status since last repair/maintenance: (seconds)
TimeInPrevOperStatus	MachineStatus	int	4	Amount of time in previous operational status since last repair/maintenance: (seconds)
TimeInUse	MachineStatus	int	4	Amount of time machine in use since last repair/maintenance
TimeLogged	LotBaseLog	datetime	8	Transaction time stamp (might be backdated)
	ResourceBaseLog	datetime	8	Transaction time stamp (might be backdated)
TimeQueued	Sublot	datetime	8	Lot queued time stamp
TimeStarted	Sublot	datetime	8	Lot started time stamp
TimeStartedInMaintStatus	MachineStatus	datetime	8	Time stamp when machine started in maintenance status
TimeStartedInOperStatus	MachineStatus	datetime	8	Time stamp when machine started in operational status
TimeSuperseded	DSBaseRTValues	datetime	8	Time stamp when data superseded
	user-defined runtime data table	datetime	8	Time stamp when data superseded
ToLocation	DisassembleLog	varchar	30	Location name of destination lot
	StepOutput	varchar	30	Destination location name
ToLotID	DisassembleLog	varchar	40	Lot identifier of destination lot
ToMaterialName	DisassembleLog	varchar	30	Material name of destination lot
ToMaterialVersion	DisassembleLog	varchar	8	Material version of destination lot
ToRouteName	DisassembleLog	varchar	30	Route name of destination lot
	StepOutput	varchar	30	Destination route name
ToRouteStep	DisassembleLog	varchar	30	Route step name of destination lot
	StepOutput	varchar	30	Destination route step name
ToRouteVersion	DisassembleLog	varchar	8	Route version of destination lot
	StepOutput	varchar	8	Destination route version
ToStepIndex	RoutePath	int	4	Internal destination route step index
TransactionName	LotBaseLog	char	26	Transaction name: (ADJUST / CLOSE / COLLECT_DATA / COMMENT / COMPLETE / CONSUME / CREATE/RECEIVE / HOLD / MERGE / MOVE / QUARANTINE / RELEASE_HOLD / RELEASE_QUAR / SHIP / SPLIT / START / TRANSFER / UNDO)

Column Name	Table Name	Data Type	Length	Description
TransactionName (*cont'd.*)	ResourceBaseLog	char	26	Transaction name: (MACHINE_BATCH_LOAD / MACHINE_BATCH_UNLOAD / MACHINE_BRING_ONLINE / MACHINE_CLOSE_REPAIR / MACHINE_COLLECT_DATA / MACHINE_COMPLETE / MACHINE_OPEN_REPAIR / MACHINE_REPAIR_DIAGNOSIS / MACHINE_REPAIR_PERSON / MACHINE_REPAIR_REASON / MACHINE_REPAIR_STATUS / MACHINE_REPAIR_SYMPTOM / MACHINE_START / MACHINE_START_REPAIR / MACHINE_STOP_REPAIR / MACHINE_TAKE_OFFLINE / MACHINE_TASK_COMPLETE / MACHINE_TASK_START / USER_CLOCK_IN / USER_CLOCK_OUT)
TransLogID	AdjustLog	char	17	Internal transaction log index
	CloseLog	char	17	Internal transaction log index
	CloseLogCollection	char	17	Internal transaction log index
	CloseLogConsumption	char	17	Internal transaction log index
	CommentLog	char	17	Internal transaction log index
	CompleteLog	char	17	Internal transaction log index
	CompleteLogCollection	char	17	Internal transaction log index
	CompleteLogConsumption	char	17	Internal transaction log index
	ConsumeLog	char	17	Internal transaction log index
	CreateLog	char	17	Internal transaction log index
	DataCollectLog	char	17	Internal transaction log index
	DisassembleLog	char	17	Internal transaction log index
	HoldLog	char	17	Internal transaction log index
	LotBaseLog	char	17	Internal transaction log index
	LotUndoLog	char	17	Internal transaction log index
	MachineDataCollectLog	char	17	Internal transaction log index
	MachineMaintActivityLog	char	17	Internal transaction log index
	MachineStatusLog	char	17	Internal transaction log index
	MergeLog	char	17	Internal transaction log index
	MoveLog	char	17	Internal transaction log index
	QuarantineLog	char	17	Internal transaction log index
	ResourceBaseLog	char	17	Internal transaction log index
	ResourceUndoLog	char	17	Internal transaction log index
	ShipLog	char	17	Internal transaction log index
	SplitLog	char	17	Internal transaction log index
	StartLog	char	17	Internal transaction log index
	UserLog	char	17	Internal transaction log index
TransLogIDType	LotBaseLog	smallint	2	Internal transaction log type
	ResourceBaseLog	smallint	2	Internal transaction log type
Type	DSTemplate	char	15	Data set type: (UDA / DS)
	MachineType	char	15	Machine type: (BATCH / SERIAL)
	MachineTypeMachineTask	char	15	Machine task type: (ON_DEMAND / PERIODIC)

Column Name	Table Name	Data Type	Length	Description
Undone	LotBaseLog	char	5	Transaction has been undone: (TRUE / FALSE)
	ResourceBaseLog	char	5	Transaction has been undone: (TRUE / FALSE)
UndoTransID	LotUndoLog	char	17	Internal index of undone transaction
	ResourceUndoLog	char	17	Internal index of undone transaction
UpperTolerance	BOMItems	float	8	Maximum tolerable quantity deviation: (Absolute: +20% = 1.2)
	BOMSubItems	float	8	Maximum tolerable quantity deviation: (Absolute: +20% = 1.2)
	StepInput	float	8	Maximum tolerable quantity deviation: (Absolute: +20% = 1.2)
	StepOutput	float	8	Maximum tolerable quantity deviation: (Absolute: +20% = 1.2)
UseCycleTime	MaterialOverride	char	5	Use this override's cycle time value: (TRUE / FALSE)
UseDefaultShifts	DaysOfMonth	char	5	Use default shifts: (TRUE / FALSE)
UseOverlapped	MaterialOverride	char	5	Use this override's overlapped value: (TRUE / FALSE)
UserCertification	IssuedUserCert	varchar	30	User certification name
	MaterialOverride	varchar	30	User certification name
	Operation	varchar	30	User certification name
	UserCertification	varchar	30	User certification name
UserName	DSBaseRTValues	varchar	30	Name of user who is logged on at time of collection
	LotBaseLog	varchar	30	Name of logged-on user
	ResourceBaseLog	varchar	30	Name of logged-on user
	user-defined runtime data table	varchar	30	Name of user who is logged on at time of collection
UseTrackingData	MaterialOverride	char	5	Use this override's queue time and complete-only values: (TRUE / FALSE)
UseYield	MaterialOverride	char	5	Use this override's yield value: (TRUE / FALSE)
Vendor	Machine	varchar	30	Vendor name
VendorID	CreateLog	varchar	30	Inventory vendor name
VendorLotNum	CreateLog	varchar	30	Vendor's inventory lot number
Version	ClassDictionary	smallint	2	Internal version index
	MachineTaskWorkInst	varchar	8	Work instructions version
WarningTimeInterval	MachineTypeMachineTask	int	4	Wear measure warning interval— elapsed time (clock or running): (seconds)
WhenStartedRunning	MachineStatus	datetime	8	Time stamp when machine started running after last repair/maintenance
WorkInstName	OperationWorkInst	varchar	30	Work instructions name
	OvrWorkInstructions	varchar	30	Name of work instructions being overridden
	WorkInstructions	varchar	30	Work instructions name
	WorkInstText	varchar	30	Work instructions name
WorkInstNum	OvrWorkInstructions	int	4	Work instructions override number
WorkInstVersion	OperationWorkInst	varchar	8	Work instructions version
	OvrWorkInstructions	varchar	8	Work instructions version
	WorkInstructions	varchar	8	Work instructions version
	WorkInstText	varchar	8	Work instructions version

Column Name	Table Name	Data Type	Length	Description
WorkState	SerialNumber	char	15	Sublot process status: (QUEUED / IN_PROCESS / CLOSED)
	Sublot	char	15	Sublot process status: (QUEUED / IN_PROCESS / CLOSED)
YearID	MonthsOfYear	char	17	Internal year identifier
	Year	char	17	Internal year identifier
	YearsOfCalendar	char	17	Internal year identifier
YearIndex	YearsOfCalendar	int	4	Internal year index
YearOfCalendar	Year	smallint	2	Year in calendar (e.g., 1996)
Yield	MaterialOverride	real	4	Operation yield: (Absolute: +20% = 1.2)
	Operation	real	4	Operation yield: (Absolute: +20% = 1.2)
ZeroDate	Lot	datetime	8	Time stamp when the lot's primary and secondary quantities reached zero

CHAPTER 7

Archive Utility

The Archive Utility is a separate program supplied with InTrack. It can be run as needed to archive, purge and restore structural and dynamic data stored in the InTrack database.

Use the utility to store system files securely.

Archiving functions also can be performed using OLE Automation objects. Refer to the OLE Reference Guide for details.

CAUTION: The Archive Utility should be accessed only by authorized system managers. Before archiving, make sure that no InTrack Runtime activities are performed. As an added precaution, all InTrack users should logoff the system prior to an archive operation.

Contents

Ensuring Proper Archive Functions

The Archive Utility can be used to purge data from the InTrack database. To ensure that archived data can be successfully restored, system administrators are advised to become familiar with the utility's formats and operations.

Archive Data Format

The formats used by the Archive Utility are exclusive to InTrack. The proprietary data stored in an archive cannot be retrieved or used by other applications.

During the archive process, binary data is written as a series of hexadecimal characters (numerals 0-9 and letters A-F).

You cannot specify the format of the archive file. InTrack supports only variable-length comma-delimited text.

Windows Regional Settings

The regional settings established for your Windows system must be consistent when archiving and restoring information.

Formats for date and time used during the archive process must be identical when restoring archived data.

All regional settings are managed by the Regional Options operations found in the Windows Control Panel.

InTrack Version

Restoration of data should be performed only on the InTrack version used to archive the data.

Archive Storage Locations

During the archiving process, you must specify the location of the target directory and a file name.

The default archive directory is \INTOUCH.32\INTRACK\ARCHIVE. You can assign one or more directories as needed for storage of archived files. The path and directory must exist before you start the Archive function.

InTrack archive file names are assigned suffix of .ARC.

You should give each archived file a unique name to ensure that the correct data can be restored if needed.

To ensure accurate identification of archive files, use a file name that identifies the date of the archive (090902.arc). If you archive data for objects individually, use subdirectories to separate the object archives, or identify the object in the file name (lot090902.arc, etc.).

Archiving and Purging

You can specify if the data to be archived also should be purged (deleted) from the database. The options are:

1. Archive and purge (data is written to the CSV file and deleted from the database).
2. Archive, no purge (data is written to the CSV file only)
3. Purge, but do not archive

If you restore archived data that was not purged, the existing data will be updated. If the data in the RDBMS is the same as the archived data, no values will be changed.

NOTE: Referential integrity is active for purging. Be sure no data to be deleted has outstanding references to it.

Archive Rate

The archiving utility can be paced to avoid interference with the production system and reduce potential burden on the RDBMS server.

The idle time option allows you to specify an idle time, as a percentage, so that the archive process will execute for a specified amount of time.

For example, an idle time of 25% will limit the archive to 75% of its time. The remaining 25% of the time will be idle.

The archive rate, total elapsed time and estimated time to complete are displayed in the Status dialog box and written to the log file.

Running the Archive Utility

For security purposes, the Archive Utility is not listed with other FactorySuite programs accessed from the Start menu.

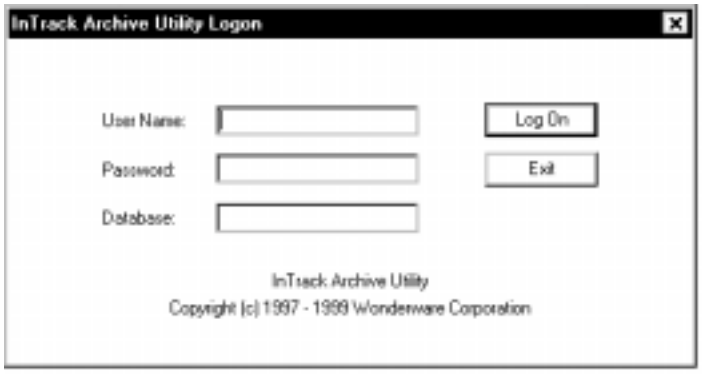
Database management privileges must be assigned for the user. Archiving should be performed only by InTrack system administrators.

Archiving should be performed on the server or client on which the archived files will be stored.

To open the Archive Utility

Use the Run option of the Start menu or the Windows Explorer to locate the file ITRARCHIVE.EXE. It will be located in the Bin for FactorySuite InTrack program files.

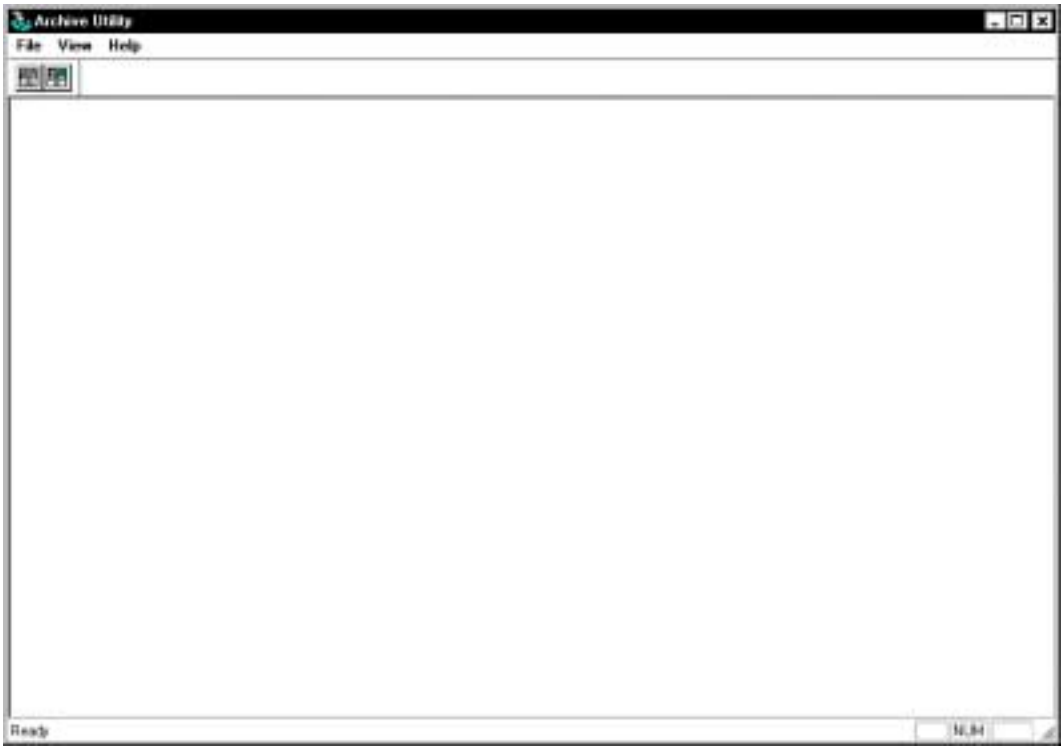
Run the file or double-click the Explorer entry. The log-in screen will appear:



1. Type a valid administrator ID in the User Name field. Click the next field or press TAB to move to the next field.
2. Type the password in the field. Click the next field or press TAB to move to the next field.
3. The name of the current InTrack Database should appear as a default. If this is not the correct database to archive, type the name of the desired.
4. Click Log On. The Archive Utility will appear. Instructions are provided on the following page.

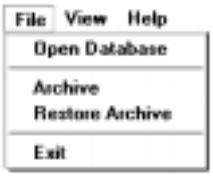
Archive Utility Screen

The Archive Utility screen provides access to all archiving functions:



The initial screen displays three menus and two shortcut keys described below.

File Menu



The File Menu provides access to the Archive and Restore Archive operations. Click the option or type ALT+F to access the menu.

Open Database

Use this option to log on to a new database. Click the option to access the original log on window described on the previous page. Complete the window with the user ID and database name.

Archive 

The Archive option provides access to the Archive screen described on the following pages. Click the option the begin the archive process.

You also can click the Archive shortcut button.

Restore Archive 

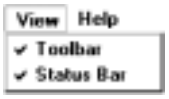
Click the option to begin the restore process.

You also can click the Restore shortcut button.

Exit

Click this option to exit the Archive Utility.

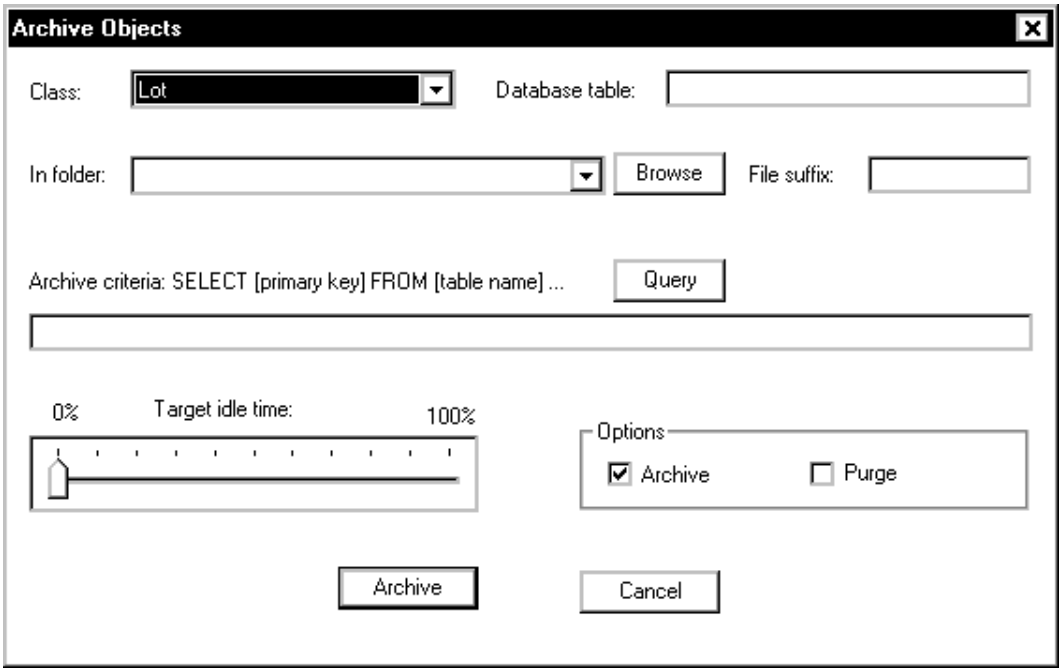
View Menu



The View menu can be used to display or hide the toolbar and status bar. Click the desired object to add or remove the checkmark indicating the view is active.

Archive Objects Entry Screen

Selecting the Archive option from the File menu or shortcut will access the Archive Objects entry screen:



Complete the fields to begin the archive process:

1. **Class**
   Click the down arrow to display a drop-down list of InTrack object classes. Click the class to be archived from the list.

2. **Database Table**
   Type the name of a database table in the field to archive a specific table in the class. This option is used primarily with dataset template tables or user defined tables. For other classes, leave the field blank to archive the complete class.

3. **In Folder**
   Use this field to type the desired path and folder of the location where the archive files will reside. The default folder is \INTRACK\ARCHIVE in the path where your InTrack program files are stored. To access the Windows File browser to locate files, click Browse.

4. **File suffix (optional)**
   You can append a character string as a suffix for the archive file. This can aid in identification of archived files. Type the desired suffix.

5. **Archive criteria**
   Use this field to enter a WHERE clause to restrict the archiving of data. This is especially useful when archiving past records. (WHERE CreateDate < "Mar 31, 2002"). The clause will be applied to the primary key records of the specified table. Click the Query button to display an object count resulting from the query. Click Yes to display the Object List dialog box and view a list of primary keys.

6. **Target idle time**
   Click and hold the slider on the target idle time display. Move it to the left or right to change the percentage of idle time that will be required for the archive operation.

7. **Options**
   Click the Archive and/or Purge button to specify whether to Archive the records, Purge the records, or both.

8. Click **Archive** to begin the process.

Archive Status Display

When the archive process begin, the Archive Status Display will appear, if you have activated it from the View menu:

The display will show the status of the archive:

- **Estimated Time**
  The time remaining to complete archiving, in hours, minutes and seconds will display.
- **Number Left**
  This displays the number of primary objects still to be archived.
- **Current idle time**
  This display shows that actual idle time, in percent.
- **Messages**
  Status messages will show warning and error messages generated by the utility, timestamped for identification. Use the scroll bars to display messages as needed.
- **Progress bar**
  The progress bar at the top of the screen shows the completion of the archive, from left to right.

To stop the archive process, click Stop. The archive utility will complete archiving the current object, then stop the process.

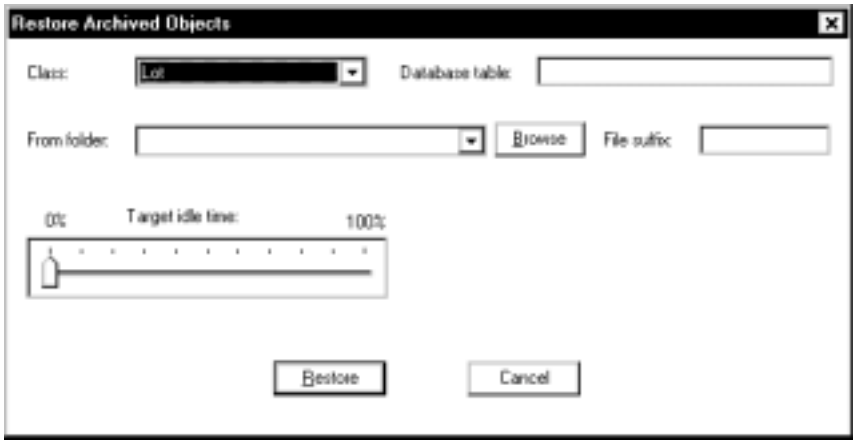
After archiving is complete, click OK to close the window.

Restore Archive Objects Entry Screen

Selecting the Restore option from the File menu or shortcut will access the Restore Archive Objects entry screen:

Archive Objects Entry Screen

Selecting the Archive option from the File menu or shortcut will access the Archive Objects entry screen:



Complete the fields to begin the restoration process:

1. **Class**
   Click the down arrow to display a drop-down list of InTrack object classes. Click the class to be restored from the list.

2. **Database Table**
   Type the name of a database table in the field to restore a specific table in the class. This option is used primarily with dataset template tables or user defined tables. For other classes, leave the field blank to restore the complete class.

3. **In Folder**
   Use this field to type the desired path and folder of the location where the archive files are located. To access the Windows File browser to locate files, click Browse.

4. **File suffix**
   Type the suffix applied to the archived objects.

5. **Target idle time**
   Click and hold the slider on the target idle time display. Move it to the left or right to change the percentage of idle time that will be required for the restore operation.

8. Click **Restore** to begin the process.

A P P E N D I X A

File Formats for WorkInstructions

Compound Documents

Ami Professional 1.x, 2.0, 3.0, 3.1

Claris Works 3.0

FrameMaker (MIF) 3.0-4.0

HTML

Interleaf Publisher 5.2, 6.0

MacWrite 5.0

MacWrite Pro 1.0, 1.1

MacWrite II, Version 1.0, 1.1

Microsoft RTF (Rich Text Format)

Microsoft PowerPoint 2.0, 3.0, 4.0

Microsoft Windows Write 3.x

Microsoft Word DOS 5.5, 6.0

Microsoft Word Mac 3.0, 4.0, 5.x, 6.0

Microsoft Word for Windows 1.x, 2.0, 6.0, 7.0, 8.0

Microsoft Works for Windows 3.0

Microsoft Works for Macintosh 4.0

Professional Write Plus

WordPerfect DOS 5.0, 5.1, 5.2, 6.0

WordPerfect Mac 3.0, 3.1

WordPerfect for Windows 5.1, 5.2, 6.0, 6.1

Write Now 3.0

Text

Ami Professional 1.x, 2.0, 3.0, 3.1

CEOWRITE 3.0

Claris Works 3.0

COM.FORMAT

CTOS DEF

Digital DX (DEC DX)

Digital WPS-Plus

DisplayWrite 2, 3, 4, 5

Enable (wp) 1.0, 2.0, 2.15

Excel (PC) 2.1, 3.0, 4.0, 5.0

Excel (Mac) 3.0, 4.0

FrameMaker (MIF) 3.0-4.0

Framework III & IV

HP AdvanceWrite Plus

HTML

IBM Writing Assistant 1.0

Interleaf Publisher 1.1

Interleaf 5.2, 6.0

Legacy 1.x, 2.0

Lotus 1-2-3, Ver. 1A, 2.x, 3.0, 4.0, 5.0

Lotus 1-2-3 for Windows 1.0

Lotus Manuscript 2.0, 2.1

MacWrite 4.5, 4.6, 5.0

MacWrite Pro 1.0, 1.1

MacWrite II, Version 1.0, 1.1

Mass-11 PC, Version 8.0 - 8.3

Microsoft Windows Write 3.x

Microsoft Word DOS 3.x - 6.0

Microsoft Word for Windows 1.x, 2.0, 6.0, 7.0, 8.0

Microsoft Word Mac 3.0, 4.0, 5.x, 6.0

Microsoft Works for Macintosh 4.0

Microsoft Works for Windows 3.0

MultiMate 3.3, Adv I, Adv II, 4

OfficeWriter 4.0, 5.0, 6.0, 6.1, 6.2

OfficePower 6-7

PeachText 5000, 2.12

PFS: First Choice (wp) 1-3

PFS: Write Version C

Professional Write 1.0, 2.x

Professional Write Plus

Q & A Write 1.x, Q & A (wp) 3.0, 4.0

Quattro Pro DOS & Windows

RapidFile (Memo Writer) 1.0, 1.2

Samna Word IV & IV Plus 1.0, 2.0

Signature 1.x

Total Word 1.2, 1.3

Uniplex onGO Write

Uniplex WP (V7–V8)

Volkswriter 2, 3, 4

Wang PC Version 3

WANG WITA

WiziWord

WordPerfect 4.1, 4.2, 5.x, 6.0

WordPerfect for Windows 5.1, 5.2, 6.0, 6.1

WordPerfect (Mac) 1.0, 2.x, 3.x

WordStar 3.x, 4.0, 5.0, 5.5, 6.0, 7.0

WordStar 2000 Release 3.0, 3.5

WordStar for Windows 1.x, 2.0

WriteNow 3.0

Xerox Interchange Format (XIF)

XyWrite III, III Plus, IV

XyWrite for Windows

Graphics

Ami Pro Graphics (SDW)

AutoCad 12 (DXF)

CompuServe (GIF)

Computer Graphics Metafile (CGM)

CorelDraw Preview

Encapsulated PostScript Bitmap Preview (EPS)

HPGL

HTML

IGES

Interleaf Graphics

JPEG

Lotus Freelance

Lotus PIC

Macintosh QuickDraw (PICT1 & PICT2)

Micrografx Designer

Microsoft Paint, Paint 2.x

Microsoft PowerPoint 2.0, 3.0, 4.0

Microsoft Word Graphic Objects

PC Paintbrush (PCX)

Portable Network Graphics (PNG)

Uniplex RGIP

Sun Raster (RAS)

Tagged Image File Format (TIFF)

Windows Bitmap (BMP)

Windows Clipboard (CLP)

Windows Icon Resource (ICO)

Windows Metafile (WMF)

WiziDraw/SysDraw

WordPerfect Graphics (WPG1, WPG2)

XIF Graphics

XWindows PixiMap (XPM)

DataBase and Mail Merge

ASCII (Comma Separated Variable, Quote Delimited)

dBase III, III+, IV

PFS: First Choice (db) 3.0

Microsoft Access 2.0

Microsoft Word 5.x, 6.0 (mail merge)

Paradox 3.5, 4.0

WordPerfect 5.1, 5.2 (mail merge)

Data

ASCII (PC, Mac, Unix, Windows)

DCA/FFT (Final Form Text)

DCA/RFT (Revisable Form Text)

EBCDIC

HP PCL

Microsoft Access 2.0

Microsoft RTF (Rich Text Format)

Navy DIF

PostScript

Glossary of Terms

Active — The status of a WIP lot, inventory lot, or bulk inventory that releases it from quarantine, enabling it to be shipped or consumed. *See also* quarantined.

Activity object — Objects created and manipulated at runtime via transactions, or OLE scripts. Transactions can be issued either manually by an operator or automatically by the InTrack system. Activity objects are dynamic components of the manufacturing model in that they can move between the different locations or operations of a manufacturing route. *See also* structural objects.

ActiveX — Microsoft's brand name for the technologies that enable interoperability using the Component Object Model (COM). ActiveX technology includes, but is not limited to, OLE.

Adjust — A WIP lot, inventory lot, or bulk inventory transaction that changes any attribute. Adjust is typically used to change the quantity, priority, or due date.

Alternate path — A path that does not follow along the main exits of a route toward the ensuing destination steps, but along side exits. A route can have more than one alternate path.

Archive — The process of extracting data from the InTrack RDBMS and saving it in operating system files for long term storage.

Backup — A set of copied data, reserved in case of a system failure.

Bill of material (BOM) — A list of all parts, components, and quantities of material needed to produce the target product (input), and any coproducts or by-products produced during the manufacture of the target product (output).

Bitmap — A graphic image, such as an icon, used to represent a route operation in the Graphical view. Bitmaps for a particular operation can be changed by selecting a different bitmap or by modifying the existing one.

BOM — *See* bill of material.

Boolean — A data type with logical (TRUE, FALSE) values.

Bulk inventory — A quantity of material that is not assigned or tracked by a lot number. Bulk inventory is tracked by storage location.

Bulk Inventory Selector A runtime display object into the InTrack database tables used by the operator for selecting bulk inventories to work on. *See also* WIP Selector, Inventory Lot Selector.

Button wizard A unique, preprogrammed combination of actions that can create or manipulate InTrack objects. Button wizards issue OLE script for the currently selected item in the appropriate selector.

By-product An output material of little or no value that is an inevitable result in manufacturing; i.e., waste. A by-product is not included in the primary quantity of the final product produced on a route. Since production of by-products reduces the WIP lot quantity, they are typically minimized in manufacturing and are tracked separately. *See also* coproduct.

Cached objects A list of objects stored locally in each workstation (node) on a multiple-user database. Cached objects are continually updated from the database when new objects are defined or when an object definition is saved or restored.

Calendar An object name that defines when one or more resources are available.

Client A user workstation containing the InTrack user interface and programs.

Client/server An architecture of hardware and software where the client (a user or program) makes requests (to the server) for resources or information.

Close A WIP lot transaction that completes the last operation in a route and moves (closes) the WIP lot to inventory.

Collect A WIP lot transaction that collects engineering data along a route to provide in process quality checks for tracking, analysis, and reporting. Information collected during runtime is entered in dataset templates that were defined in ModelMaker.

Column *See* field.

Comma Separated Variable (.CSV) A format used by the clipboard for transfer of text columns and numerical data between applications. A .CSV data item is similar to text, with each variable separated by a comma. Although Microsoft Excel is probably the principal creator of .CSV clipboard data, many DOS applications support this format.

Comment A WIP lot, inventory lot, or bulk inventory transaction that allows the operator to log comments. The comments are stored in the InTrack database.

Complete A WIP lot transaction that moves a WIP lot to the next operation. Issuing this transaction changes a WIP lot's status to "queued."

Configure A WIP lot, inventory lot, or bulk inventory transaction that allows only lots or inventory with certain parameters to be displayed in the appropriate selector.

Configure to order Deviation from a standard way of manufacturing a product to meet specific customer requirements.

Connect A system transaction that establishes a connection to the database.

Connection The graphical line connecting a route step with the next route step (the destination route step). Each connection has a direction indicator to indicate the path of the route. A main exit connection is centered on any side of a route step, and a side exit connection is offset from the center on any side. *See also* main exit, side exit.

Consume A WIP lot transaction that controls and tracks the quantities of input material consumed at an operation.

Consumption *See* material consumption.

Coproduct A valuable output material, but not the final product produced on a route. As coproducts are produced, they are moved off the route, given a different lot number, and are tracked separately on another route. A coproduct does not reduce the quantity of the WIP lot from which it originates. *See also* by-product.

Create A WIP lot transaction that creates a new WIP lot and queues it at the initial step on a manufacturing route.

Customer specifications Special customer requirements for producing a product. Customer specifications are object names referencing a specific set of changes to an operation's attributes, called overrides. *See also* override.

Customization The modifying of a system to meet the specific needs of a user.

Cycle time The processing time necessary for completing an operation.

Data Information associated with a manufacturing system.

Data collection The process of acquiring meaningful data for work in process, usually used for analysis and record keeping. Data is typically collected when an operation is executed and can be alphanumeric, analog, or Boolean. Data is collected in groups controlled by dataset templates, which are defined in ModelMaker.

Dataset template The definition of a group of data items to be collected together at runtime; a form defined in ModelMaker that the operator uses to enter runtime data. When a dataset template is defined and saved, a new runtime table is created to store collected datasets. *See also* dataset.

Dataset Data collected in a dataset template. *See also* dataset template.

Database A system depository of common types of data, sorted by unique identifiers, organized into tables. The InTrack database tables contain both the manufacturing model objects defined in ModelMaker and the lot transaction data created during runtime.

Detail (1) A WIP lot, inventory lot, or bulk inventory transaction that returns attribute information about the lot or inventory. (2) Attributes for a particular activity object. Object details include the defined characteristics of an object, such as the name of the object, the quantity, its quarantine status, any user-defined attributes, etc.

Destination Route Step From a route step, the next step on the route according to a specific disposition code.

Disassemble A WIP lot transaction that controls and tracks the quantities of input material previously consumed (assembled) at an operation that are removed from the assembly.

Disconnect A system transaction that terminates the current connection to the database.

Dispatching The assigning of priority ratings to lots queued at an operation. WIP lots with higher priority ratings usually are worked on before WIP lots with lower priority ratings. Priority ordering is typically based on the due dates or slack times of the WIP lots.

Disposition code A value representing the disposition of a WIP lot once it has completed an operation. The disposition code represents the condition under which quantities of the WIP lot move out of a route operation or are scrapped at a route step.

Due date The date the WIP lot is supposed to be completed at the last operation on a route. The WIP lot may be created using the default due date or a date entered manually by the operator. The default due date is calculated by adding the cumulative cycle times for the route operations to the current date.

Dynamic link library (DLL) A software module loaded into memory at execution time in order to access functions of that model.

Enterprise resource planning (ERP) Software that has expanded the functionality of a manufacturing resource planning (MRP II) system.

ERP *See* enterprise resource planning.

Event The activity in a system initiated by the user in order for processing to take place.

Field A column in a database table that describes one characteristic of the entity, such as a WIP lot's primary quantity or current operation, an inventory's location or area, or the date or time a transaction occurred. *See also* record, table.

Finished goods inventory (FGI) A collection of material representing the final product manufactured on a route.

Flow lot A WIP lot queued or in process at more than one operation simultaneously. The individual WIP lot quantities of a flow WIP lot are called sublots.

Function A procedure in programming language.

Graphical route editor (GRE) The visual display in ModelMaker; also known as the Graphical view.

Graphical user interface (GUI) The visual and functional operation of an application that provides the user with the means of interacting with the application at runtime.

Graphical view One of five views of the route document window that visually displays a route and is used for defining route steps. *See also* route views.

GUI *See* Graphical User Interface.

Hold *See* on hold.

Index An ordered list of data records (rows) that are returned as a result of a database query.

Initial step Typically the first step on the route. In runtime, new WIP lots are automatically queued at the initial route step to begin processing.

Input Data entered into the system by the user or another system, device, or program.

Input material The elements, constituents, or substances used for manufacturing a target product; defined as part of the bill of material for that product.

Input parameter Defines the values needed to complete a transaction (e.g., quantity to start, attribute to adjust).

Instruct A WIP lot feature that displays work instructions for a particular operation. *See also* work instructions.

Integer A whole number, such as 0, 50, or 764. Integers can be signed (positive or negative) or unsigned (positive) and are without specified value limits for data collected.

Integrator An application builder, or an application programmer who customizes a program by enhancing and building user interfaces so programs can communicate and share data.

Intermediate material Material that has exited an operation and waits in queue to enter the subsequent operation on a route.

Inventory A collection of material tracked by lot number or storage location. *See also* inventory lot, bulk inventory.

Inventory Lot A collection of material of the same type (product) stored in inventory and tracked by a lot number. *See also* bulk inventory.

Inventory Lot Selector A runtime display object into the InTrack database tables used by the operator for selecting lot inventories to work on. *See also* WIP Selector, Bulk Inventory Selector.

Key A field in the database, such as LotID, or MatlTypeName (material type name), used to locate activity objects. One or more keys constitute the object ID for the activity object. WIP lots, inventory lots, and bulk inventory all require a different set of keys in order to be identified by the InTrack system. *See also* object ID.

Laboratory information management system (LIMS) A system used for data collection in a laboratory environment.

LIMS *See* laboratory information management system.

Location A department or site (physical or logical) responsible for processing lots of work in process or inventory. Examples of locations are physical positions of machines, raw material inventory, and finished goods inventory. A location provides lot ID uniqueness; that is, the combination of location and lot ID must be unique.

Log Tables of data, categorized by common types of information, that reside in a database and use a time stamp as a primary key.

Lot A collection of one or more items of the same material type that are manufactured, stored, and tracked together. There are two types of lots - WIP and inventory.

Machine A device that performs some process on a lot at an operation. Once a machine is defined, you can associate it with one or more operations in the database.

Machine type An object used to group machines that have similar maintenance, failure, and repair characteristics. A machine type lets you update the failure symptoms and maintenance and repair tasks for all the machines in the group just once rather than individually for each machine.

Main exit An exit out of a route step that is associated with the default disposition code. The main exit connection is centered on any side of a route step.

Main path The primary sequence of operations a WIP lot will follow along a route. WIP lots that complete an operation and are assigned the default disposition code progress through the route along the main path. The main path is used in cumulative yield and cycle time calculations.

Maintenance The process of keeping system data current and usable.

Manufacturing Execution System (MES) A set of programs that performs the functionality of plant management, supervisory control, quality management, and plant engineering. MES provides a link between process control systems on the factory floor and the MRP II business system.

Manufacturing Resource Planning (MRP II) A set of programs that plans production to meet demand. MRP II, referred to as a "business system," is the outgrowth of material requirements planning (MRP).

Material The elements, constituents, or substances used for manufacturing a target product (input material), and the coproducts and by-products of the target product (output material).

Material consumption The process of using up input material during an operation.

Material input view One of five types of views in the route document window; used to assign consumable materials to route steps. *See also* route views.

Material output view One of five types of views in the route document window; used to assign by-products and coproducts to route steps. *See also* route views.

Material Requirements Planning (MRP) A system that collects data from other available manufacturing systems (inventory, BOM, WIP, scheduling) to compute material requirements based on sales forecasts.

Material type Identifies and describes items in raw inventory, intermediate material, WIP material, or finished goods; the type of material(s) used in manufacturing a WIP lot.

Merge A WIP or inventory lot transaction that combines two or more lots into one lot along a route. To merge, lots must be (1) queued at the same operation, (2) producing the same product, and (3) have the same status. The system maintains the genealogy of merged lots.

Model The functionality engine of InTrack. The model is a representation of a plant's actual manufacturing processes. The model includes static components, such as operations, and dynamic objects, such as work in process (WIP) lots. The model also includes: materials, the bill of material, work instructions, specifications for data collection, disposition codes for output materials, and locations.

Modeling The process of creating a computer model of plant floor operations. During the modeling process, the paths (routes) that move products through successive stages in manufacturing and the relationship between these routes are defined.

ModelMaker The development environment used to create the manufacturing model. *See also* runtime environment.

Move An inventory lot or bulk inventory transaction that transfers some or all of an inventory to an existing inventory in the same location, or moves the entire inventory lot into another location, storage location, or both. *See* Nonstandard move.

Network A communications infrastructure connecting a group of physically connected computers.

Nonstandard move A WIP lot transaction that changes the WIP lot position to a route step other than the next step on the main path, as defined by the disposition code. A nonstandard move can only be performed if the WIP lot is in queue at an operation.

Object An item or record, usually stored in a database, representing a uniquely definable entity in an application, such as a route, operation, or disposition code.

Object ID Identifier used by the InTrack system to locate objects in the relational database. An object ID points to a specific field(s) in the tables. *See also* keys.

Object model foundation (OMF) The collection of object types representing a manufacturing process; the basic design concept of ModelMaker.

Object report A printout of any or all of the static information displayed in the object viewer area. The object list should always be updated before an object report is generated.

Object selector button A button that appears at the right of every field requiring an object name. When selected, it causes a Select... dialog box to display.

Object type A particular class or category of objects.

OLE .Microsoft's object-based technology for sharing information and services across process and machine boundaries (object linking and embedding).

On hold The status of a WIP lot, inventory lot, or bulk inventory quantity when it is in a unprocessable state for an unspecified length of time, pending release. When a quantity is on hold, it cannot be moved. *See also* released.

Operation An element of work performed during the manufacturing process. An operation can occur at one or more steps on a route and can be associated with one or more routes. An operation can have a work instructions set and one or more dataset templates associated with it.

Operation bitmap In a route step graphic, a bitmap that is an operation attribute and displays once an operation is assigned to a route step.

Output Data processed and formatted for use by an end user, system, device, or program; graded material coming out of an operation.

Output material Material created as the result of an operation. Output materials include coproducts and by-products. *See also* coproduct, by-product.

Output parameter Defines a value to be returned from the database once a runtime transaction is complete (e.g., number of total records found, value of an object attribute).

Overlapped operation An operation that completes the processing of WIP lots in partial quantities, and advances the partial quantities to the next operation on the route. *See also* flow lot.

Override A changed attribute of an operation on a route that manufactures a target product. The operation attribute change is specific to the product or a customer specification associated with that product.

Pane One of the two main sections of the Route window. The Graphical view is displayed in the top pane, and the Step Properties, Material Input, and Material Output views are displayed in the bottom pane.

Parameter An informational element that has a value. Parameters define the values to be written to or returned from the database. *See also* input parameter, output parameter.

Parameter type Specifies how a parameter value is stored in the InTrack relational database (message, integer, date).

Parent lot The original WIP lot that has been divided into sublots.

Parent DataSet Template The definition of a group of data items for which multiple other dataset templates can depend upon for all or part of their definition.

Primary quantity The quantity for a particular WIP lot on a route. The primary quantity defaults to the quantity specified as the standard quantity on the bill of material for the product.

Print A system command that prints a database report.

Product Generally, a type of material produced by a manufacturer. A product's characteristics are defined by the product's material type. During manufacture, a product can be called a part, a component, a product, or a lot. A final product is the primary finished goods produced on a route. *See also* by-product, coproduct, *and* target product.

Product genealogy The complete manufacturing history of a product.

Purge A system transaction that purges from the database WIP lots, inventory lots, and bulk inventory that have a quantity of zero and have exceeded a specified time limit from the last activity date.

Quarantined The status of a WIP lot, inventory lot, or bulk inventory where the lot or inventory is allowed to continue processing, but is prevented from being shipped or consumed. *See also* active.

Query A script statement issued to the database by a client that searches for objects in a database table. Queries can be performed for both activity objects and structural objects.

Query result A script statement that returns the characteristics, or object attributes, of any object located by a database query. A query result can only be issued for one object at a time.

Queue (1) The state in which a WIP lot waits to enter an operation on a manufacturing route. The arrangement of the WIP lots in queue determines the processing priority. (2) The order in which transactions are processed by the InTrack system.

Queue time The amount of time a WIP lot waits to start an operation.

Raw material Input material that has not been processed on a manufacturing route.

Raw material inventory Inventory of unprocessed material to be consumed during the manufacturing process.

Real A floating point number represented by digits with a fixed base, such as the decimal system. A real number can be made up of either a finite or infinite set of digits.

Receive An inventory lot or bulk inventory transaction that creates new inventory or brings material into existing inventory.

Record In a database table, a row describing a single occurrence of a specific entity, such as a WIP lot, an inventory, or a transaction.

Reinitialize Returning a database to its initial state. Reinitializing erases all rows from InTrack tables (all model objects created using InTrack ModelMaker and all runtime data are removed).

Relational database A database structure that organizes data according to the relationships between the data. In a relational database, relationships between data items are expressed by means of tables.

Relational database management system (RDBMS) A system that manages access, integrity, recovery, and security of data stored in a database. All clients must be able to communicate with the RDBMS before InTrack is installed.

Released The status of a WIP lot, inventory lot, or bulk inventory that releases it from being on hold, enabling it to be processed. *See also* on hold.

Remaining cycle time The time left in the cycle to complete the WIP lot on the route. The remaining cycle time is calculated by adding up the cycle times for the remaining operations on the route.

Restore The process of writing previously archived data back to the RDBMS. See "archive".

Resynchronize Updating local cache memory with current, saved definitions in the database.

Rework A disposition code for a route operation that will move all or part of the WIP lot to another route step or to another route for additional processing. The system does not enforce a predefined meaning for the rework-type disposition code.

Route A network of possible paths containing operations that a WIP lot can follow during manufacturing. A route is made up of route steps, with an operation at each step. The route specifies the sequence of operations that are performed on a WIP lot as it moves along the route.

Route graphic A graphic that depicts a route step and shows details about the properties of the route step. For example, the presence of shading or a border on a route step provide visual representations of the route and route step characteristics.

Route path The logical movement of a WIP lot between two route steps.

Route step On a route, a step associated with one operation.

Route views Multiple types of route displays, used in InTrack to show multiple objects and their attributes and relational dependencies to each other; graphical, tabular, step properties, material input, and material output.

Row *See* record.

Runtime environment Activities performed by an end user (operator) to execute the application once it has been built; the time during which the control unit fetches data and the arithmetic-logic unit performs actual processing. *See also* ModelMaker.

Scale To define input or output materials so that their quantities will always match the ratio of the target product quantity.

Schema The structure of the tables in a database. The schema of the database tables for a manufacturing process can only be changed by a system administrator or by adding UDAs and dataset templates to the model.

Scrap To remove a quantity from the route. Scrapped quantities are no longer tracked.

Script A set of instructions executed within the InTrack runtime environment to automatically perform transactions.

Script function A set of logical commands or functions used at runtime to process WIP lot, inventory lot, or bulk inventory items.

Secondary quantity A trackable quantity for a particular WIP lot on a route in addition to the primary quantity. A secondary quantity is tracked at the WIP lot level; a WIP sublot cannot have a secondary quantity.

Security group An object in ModelMaker that establishes access privileges for an assigned group.

Selector wizard A spreadsheet type display object used for viewing and selecting work items at runtime.

Server The hardware system containing InTrack database(s) and data.

Setpoint The value, such as a target temperature or other measurement setting, that is downloaded to a process controller to define the normal running parameter of a machine.

Ship A lot or bulk inventory transaction that tracks finished goods being shipped to customers and removes the shipped quantity from the database.

Side exit An exit out of a route step that is associated with a disposition code other than the default. A different side exit is used for each disposition code associated with the route step. A side exit connection is offset from the center on any side of a route step.

Slack time The difference between the due date and the standard cycle time for the route. Slack time is created when a manually assigned due date is an overestimate, or when a WIP lot progresses through operations on the route faster than the cycle times specified for the operations.

Split A WIP lot or inventory lot transaction that divides one lot into two lots, called sublots. The system maintains the individual genealogy of a split lot.

Split lot A quantity that has been split off from a WIP lot. The system maintains the individual genealogy of a split lot.

Standard yield *See* yield.

Start A WIP lot transaction that changes the WIP lot status from "in-queue" to "in process."

Statistical process control The method of monitoring and controlling a process by gathering data about the characteristics of the output, analyzing the data, and drawing conclusions based on that data.

Statistical quality control General management of quality control, cause and effect analysis, and compliance to industry standards (ISO 9000).

Status A WIP lot, inventory lot, or bulk inventory transaction used to place a lot or inventory quantity on hold or in quarantine. This transaction is also used to release a lot or inventory quantity on hold or in quarantine.

Status bar A line at the bottom of ModelMaker window that displays information about the current status of the application. The status bar can be displayed or hidden from view.

Step ID The identifier for a single step in a route. The default step ID is a sequential number assigned as the route step is built. The default step ID can be changed to a different number or to a character string.

Step name The identifier for a single step in a route. The step name is a combination of the step ID and the operation name, separated by a colon.

Step properties view — One of five types of views in the route document window; used to assign disposition codes between two route steps. *See also* route views.

Stock room — An inventory location, where material is received into inventory or issued for consumption.

Storage location — A place where an inventory lot or bulk inventory physically resides.

String — A text expression treated as a single data item. A string does not require a special format or syntax.

String selection — A discrete list of possible text choices (such as Red, Green, Blue).

Structural object — Objects created at development in ModelMaker. Structural objects are, for the most part, static components of the manufacturing model. When used together, structural objects provide the "structure," or rules, that support the flow of WIP lots, inventory lot, and bulk inventory and their related data in a manufacturing enterprise. *See also* activity object.

Structured Query Language (SQL) — A language used in relational database systems for defining, searching for, and manipulating data.

Sublots — Quantities of a WIP lot that are spread out among consecutive operations. Sublots share the same lot ID as the "parent," or original, WIP lot.

Substitutes — Materials that can be used in place of previously designated materials in a product's bill of material. Substitutes can be assigned in varying ratios of UOM to the material it is replacing, and in various percentages of the originally designated material.

Table — Group of related data entities and their characteristics. *See also* row, column.

Tabular view — One of five types of views in the route document window; used to assign the operations for each route step. *See also* route views.

TagList — A collection of tagnames referenced by one or more labels or names. TagLists can be used by one or more InTrack commands that will link the tagnames in the list to command parameters so that the system can automatically perform validation, data acquisition, and processing. There are five types of TagLists: Command TagLists, Data Collection TagLists, Object Details (UDAs) TagLists, Diagnostics TagLists, and Query TagLists.

TagList template — A pre-defined set of input and/or output parameters upon which a TagList is based. Each type of TagList has one or more available templates. *See also* TagList, input parameter, *and* output parameter.

Target product The primary finished goods produced on a route. The target product of one manufacturing route can be an input material for another manufacturing route.

Template *See* TagList template.

Toolbar A line of short-cut buttons located directly below the menu bar. ModelMaker toolbar includes the object editor buttons. The toolbar can be hidden from view, or displayed with labeled or unlabeled buttons.

Tracking Monitoring and recording the changing status of material quantities traversing a route, or residing in inventory.

Transaction A collection of one or more OLE script statements that read and/or write to the relational database. A transaction is a request to the InTrack system to find, enter, change, or return information about an activity or structural object the relational database. All transactions are processed at runtime and are performed as a single unit of work for an object.

UDA *See* user-defined attribute.

Unassemble A WIP lot transaction to remove previously assembled components. See "Disassemble".

Unarchive The process of restoring to the InTrack RDBMS previously archived data. See "Archive".

Undo A WIP lot, inventory lot, or bulk inventory transaction that returns the selected lot or inventory to the status it had prior to the most recent transaction .

User A person performing work in InTrack. User access is controlled in InTrack ModelMaker.

User certification Qualifies a worker to perform specific operations and allows the qualification to be tracked in InTrack.

User-defined attribute (UDA) A characteristic of an object that a user defines uniquely (as opposed to built-in attributes). UDAs are contained in fields added to the database using the InTrack ModelMaker.

Version The subdivision of a class of structural objects according to criteria established by the end user; the naming convention used for edited revisions of an object.

Views Types of displays in InTrack that allows a user to view, define, and manipulate objects in a tracking model.

WindowMaker The development environment used to create the display and control components for viewing and manipulating the manufacturing model at runtime.

WindowViewer The runtime environment used to view and manipulate the database for the manufacturing model.

WIP *See* work in process.

WIP lot A collection of the same material type processed and tracked together through manufacturing processes.

WIP Selector A runtime display object into the InTrack database tables used by the operator for selecting WIP lots to work on. *See also* Bulk Inventory Selector, Inventory Lot Selector.

Work in process (WIP) Material undergoing a value-added transformation by completing operations along a route.

Work instructions A set of instructions that indicate to an operator task(s) to perform at an operation, safety procedures, etc.

Yield The percentage of good material coming out of an operation. Yield is used to determine how much material is needed to start a route to produce the target product.

Index