SIEMENS

SIMATIC PCS 7 OSx

Process Configuration Manual

Order Number: 6ES7 6550XX058BC2
Manual Assembly Number: 2810913–0001
Original Edition

⚠ DANGER

DANGER indicates an imminently hazardous situation that, if not avoided, will result in death or serious injury.

DANGER is limited to the most extreme situations.

⚠ WARNING

WARNING indicates a potentially hazardous situation that, if not avoided, could result in death or serious injury, and/or property damage.

⚠ CAUTION

CAUTION used with a safety alert symbol indicates a potentially hazardous situation that, if not avoided, could result in minor or moderate injury.

CAUTION

CAUTION used without the safety alert symbol indicates a potentially hazardous situation that, if not avoided, could result in property damage.

NOTICE

NOTICE indicates a potential situation that, if not avoided, could result in an undesirable result or state.

MANUAL PUBLICATION HISTORY

SIMATIC PCS 7 OSx 4.1.2 Process Configuration Manual
Order Manual Number: 6ES7 6550XX058BC2
Refer to this history in all correspondence and/or discussion about this manual.

Event	Date	Description
Original Issue	7/02	Original Issue (2810913–0001)

LIST OF EFFECTIVE PAGES

Pages	Description	Pages	Description
Cover/Copyright	Original		
History/Effective Pages	Original		
Trademarks	Original		
iii — xxvii	Original		
1-1 — 1-35	Original		
2-1 — 2-13	Original		
3-1 — 3-67	Original		
4-1 — 4-11	Original		
5-1 — 5-40	Original		
6-1 — 6-31	Original		
7-1 — 7-11	Original		
8-1 — 8-25	Original		
9-1 — 9-29	Original		
10-1 — 10-37	Original		
11-1 — 11-13	Original		
12-1 — 12-41	Original		
13-1 — 13-17	Original		
14-1 — 14-15	Original		
15-1 — 15-7	Original		
16-1 — 16-11	Original		
A-1 — A-27	Original		
B-1 — B-8	Original		
C-1 — C-6	Original		
D-1 — D-12	Original		
Index-1 — Index-12	Original		
Registration	Original		

Trademarks

Contents

List of Figures

List of Tables

Preface

New Features of PCS 7 OSx

SIMATIC PCS 7 OSx Release 4.1.2 supports the following new features:

- **OSx Merge Utility** — This utility allows you to merge the configuration from one OSx system (or a subset of that system) into another, currently running OSx system. This means that you can do major configuration development outside of an OSx system that is running a process, and then add it in without shutting down the process.
- **Remote computer data archiving** — Data archives can be stored on a remote computer. This can be any computer system that can support an FTP server; for example, UNIX, Linux, Windows NT, Windows 2000.
- **Permanent select list** — This feature allows you to choose whether the select list for tag details, graphics, reports, and so on (accessed from the Directory button) remains on the screen until you dismiss it, or disappears when you select an entry.
- **Graphic/tag cross reference report** — A standard report cross-referencing tags in graphics, by tag and by graphic, is available.
- **Internet Protocol netmask configuration** — You are prompted to specify the netmask value or select a default at installation.
- **Save new tag install file to hard disk** — The feature allows saving a tag file to hard disk in addition to MO disk and diskette.
- **SIMATIC Rack PC 840 support** — The Rack PC 840 will now be supported as a system unit.
- **1.3 gigabyte and 640 megabyte MO disk support** — Data Archiving and Backup/Restore will support larger MO disks with the Rack PC 840 hardware platform. Earlier 230 and 540 megabyte MO disks are still supported as well.
- **Additional printer support** — New printers in the Hewlett-Packard DeskJet line are supported.

Conventions Used in the Manual Set

The procedures in the various manuals give you step-by-step instructions about how to carry out tasks. Typically, the last step of any procedure requires that you select the **OK** or **Save** button, press **Enter**, etc. To save space and avoid redundancy, this last step does not appear in the procedure. However, you need to finish each procedure with one of these actions.

OK Saves information that you have entered and closes the window.

Save Saves information that you have entered and does not close the window.

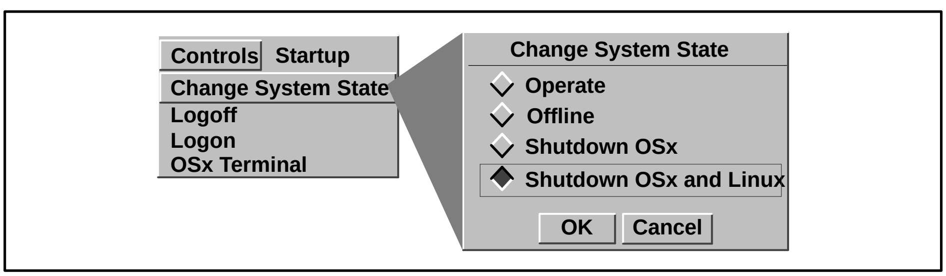
Cancel Closes the window without saving any information that you entered and terminates any action that you initiated.

Dismiss Closes the window without undoing any changes that you have entered. However, if you press **Dismiss** before you press **Enter**, the changes that you made are discarded.

The different fonts used in the manual set have the following meanings.

- Entries that you type from the keyboard are indicated with the `courier font`.
- Items that you select on the screen, or keys that you press on the keyboard, are indicated with this **bolded font.**

Items that you select on a cascaded menu are linked in the manual text with arrows. The first term indicates where to click the main menu bar. For example, **Controls->Change System State** tells you to click **Controls** on the main menu bar, then select **Change System State** from the pull-down menu.



Purpose of This Manual

The *SIMATIC PCS 7 OSx Process Configuration Manual* describes the primary tasks required to configure SIMATIC PCS 7 OSx to control your process.

This manual has been rewritten to reflect the new features. All new information for this release is indicated in this manual by change bars in the margin.

- Chapter 1 describes where to begin configuration, and how to do some basic tasks, such as powering up the system.
- Chapter 2 describes how to link tags, user IDs, alarm groups, and action requests with process group functions.
- Chapters 3, 4, 5, and 6 define the concept of the OSx tag, how to configure a tag and to adjust tag capacities in the system, and how to view tag information when the system is in the Operate state.
- Chapter 7 describes system security.
- Chapter 8 describes how to configure historical and real-time trends.
- Chapters 9 and 10 explain the alarm system and how to configure alarms.
- Chapter 11 describes how to link screen displays so that the operator can access them easily when the system is in the Operate state.
- Chapters 12 and 13 explain how to create action requests. Action requests are messages to the operator that appear in the Operate state.
- Chapter 14 describes system log reports and how to access them.
- Chapter 15 describes the mini-windows feature that displays alarms and action requests associated with a graphic.
- Chapter 16 describes how to create, modify, and display an arrangement of graphic windows as a single entity.

The Other Manuals

The OSx manual set consists of several manuals. If you cannot find the information that you need in the *SIMATIC PCS 7 OSx Process Configuration Manual*, check these other books:

- *SIMATIC PCS 7 OSx System Administration Manual* This manual describes configuring network nodes, and procedures that explain how to configure printers, how to archive data, and how to back up files. Other functions normally carried out by the systems administrator are also described here.
- *SIMATIC PCS 7 OSx Graphical Editor Manual* This manual describes how to create the graphical displays used with OSx.
- *SIMATIC PCS 7 OSx Hardware Manual* This manual describes the various hardware components of the system and how to install them.
- *SIMATIC PCS 7 OSx Reports Manual* This manual describes how to create reports on your process and your OSx configuration.
- *SIMATIC PCS 7 OSx Recipe Manual* This manual describes more advanced configuration tasks involving the creation and use of recipes.
- *SIMATIC PCS 7 OSx Batch Programming Manual* This manual describes more advanced configuration tasks involving the use of BCL, the Batch Control Language, and the creation of batch programs.
- *SIMATIC PCS 7 OSx Operator Manual* This manual is written for the configuration engineer, but it describes how to carry out the various tasks that the process operator must do when the system is in the Operate state.
- *SIMATIC PCS 7 OSx Interface to S5 Controllers User Manual* This manual describes the OSx interface with SIMATIC S5 controllers.
- *SIMATIC PCS 7 OSx Interface to S7 Controllers Manual* This manual describes the OSx interface with SIMATIC S7 controllers.
- *SIMATIC PCS 7 OSx Library Manual* This manual describes function blocks used to program the S7-400 controllers to interface with OSx.

Be sure to check the Readme File for information that did not become available until after the publication deadlines for the OSx manuals. The Readme File also points to important copyright, licensing, and warranty information. Select **Help->About OSx** from the main menu bar, and then click the **Show Readme** button at the bottom of the About OSx dialog box.

Optional SIMATIC PCS 7 OSx Features

The following manuals are available for optional SIMATIC PCS 7 OSx features.

- *SIMATIC PCS 7 OSx Remote Data Transfer Manual* This manual describes the remote data transfer feature. The remote data transfer feature allows you to transmit data collected from the process by an OSx station to an Oracle database on the remote computer for historical records and other purposes.
- *SIMATIC PCS 7 OSx X Terminal User Manual* This manual describes how to connect and operate an X terminal as an extension of an OSx station.
- *SIMATIC PCS 7 OSx @aGlance User Manual* This manual describes how to import OSx data into a Windows application, such as Excel or Lotus 1-2-3, or into another UNIX or VMS application.

If You Need Help

If you have difficulty with your system, contact the Siemens Energy & Automation, Inc., Technical Services Group in the U.S.A. at 800-333-7421. Outside the U.S.A., call 49-911-895-7000.

Chapter 1

Getting Started

1.1 Identifying the Components

Figure 1-1 shows the major hardware components associated with a SIMATIC PCS 7 OSx station.

Keyboards

The SIMATIC PCS 7 OSx system has two keyboards: a factory-hardened membrane keyboard for the operator, and a standard keyboard for the design engineer. The engineering keyboard plugs into the operator keyboard for configuration modifications in the Operate state. A rocker switch on the back of the operator keyboard allows you to set the current operational keyboard.

The operator keyboard is a sealed membrane keyboard and has been optimized for use during process monitoring and control. Operations such as alarm management and screen navigation, are easier to do using the operator keyboard. The engineering keyboard is a standard office-style keyboard that is more convenient to use for system configuration.

Mouse/Trackball

You can use the mouse or the trackball for cursor control. Appendix B lists the keyboard commands to use when a mouse is not available.

Storage Devices

Three data storage units are included: a 3.5-inch diskette drive, a hard disk drive system, and a magneto-optical drive system.

Other Network Nodes

The SIMATIC PCS 7 network includes several other kinds of nodes:

- The control nodes, which are the programmable controllers listed in Figure 1-1, run the process itself.
- You can use the optional SIMATIC APT (Application Productivity Tool) to create the controller program in a structured environment using a high-level graphical programming language. When the APT workstation is connected to the Industrial Ethernet, OSx addresses it as an engineering node. You can download the APT program from the engineering node directly to the programmable controller, and you can download tag data directly to an OSx station.
- You have the option of connecting a management information system to the network to send data for historical records and other purposes using file transfers through TCP/IP or the Remote Data Transfer or @aGlance features. Remote Data Transfer and @aGlance are optional features that can be purchased separately.

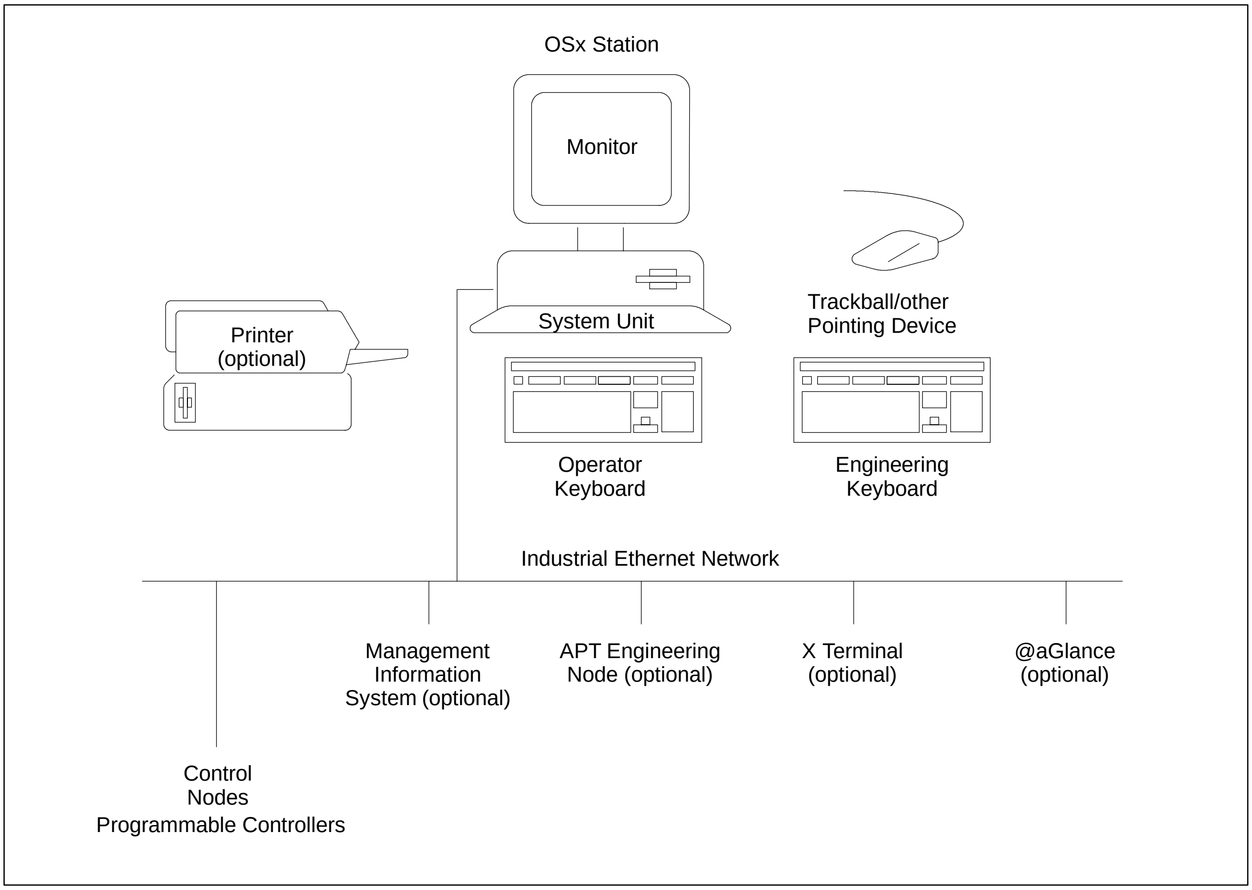


Figure 1-1 SIMATIC PCS 7 OSx Hardware Components

Relationship of SIMATIC PCS 7 OSx System Components

This section gives a brief description of the logical components that comprise a system (Figure 1-2). You can connect up to 16 OSx stations and 32 control nodes to the network. For more information about system components, refer to the *SIMATIC PCS 7 OSx System Administration Manual*.

Control Nodes The programmable controllers that run the application process are designated as control nodes.

OSx Stations The workstations that monitor and control the factory process by communication with the programmable controllers are called OSx stations. Depending on the software loaded on a particular system, an OSx station functions either as a supervisory station or as an operator station.

A supervisory station supports the task of maintaining historical data such as trending and report outputs. It also supports all the services provided by the OSx station. A multiple-station system supports up to four supervisory stations that can assume any of the roles listed below. In a single-station system, the supervisory station can have either of these two roles: primary or out_of_service.

- PRIMARY_ROLE — The station that has the primary role coordinates system functions and maintains the reference copy of the database for the SIMATIC PCS 7 OSx system. Only one OSx station can be assigned the primary role at a given time.
- BACKUP_ROLE — The station that has the backup role automatically assumes the role of primary if the primary fails. Only one supervisory station can be assigned the backup role at a given time.
- SYS_ADMIN — Use the sys_admin role for selected modifications and/or deletions of system configurations. Only one supervisory station can be assigned the sys_admin role at a given time.
- OUT_OF_SERV — Set a station to the out_of_service role before you shut down the station for maintenance or if you need to delete it from the database using Network Setup. You can set a station out_of_service when you need to run diagnostic routines, including testing communications between the station and the control nodes.
- NORMAL_ROLE — A station's standard operational role is normal if you have not assigned it the primary, backup, or sys_admin role.

An OSx station supports the tasks typically needed by process-operator personnel, such as graphics, tag detail, and trend displays. An OSx station can assume the role of normal, primary (if no supervisory stations are available), or out_of_service.

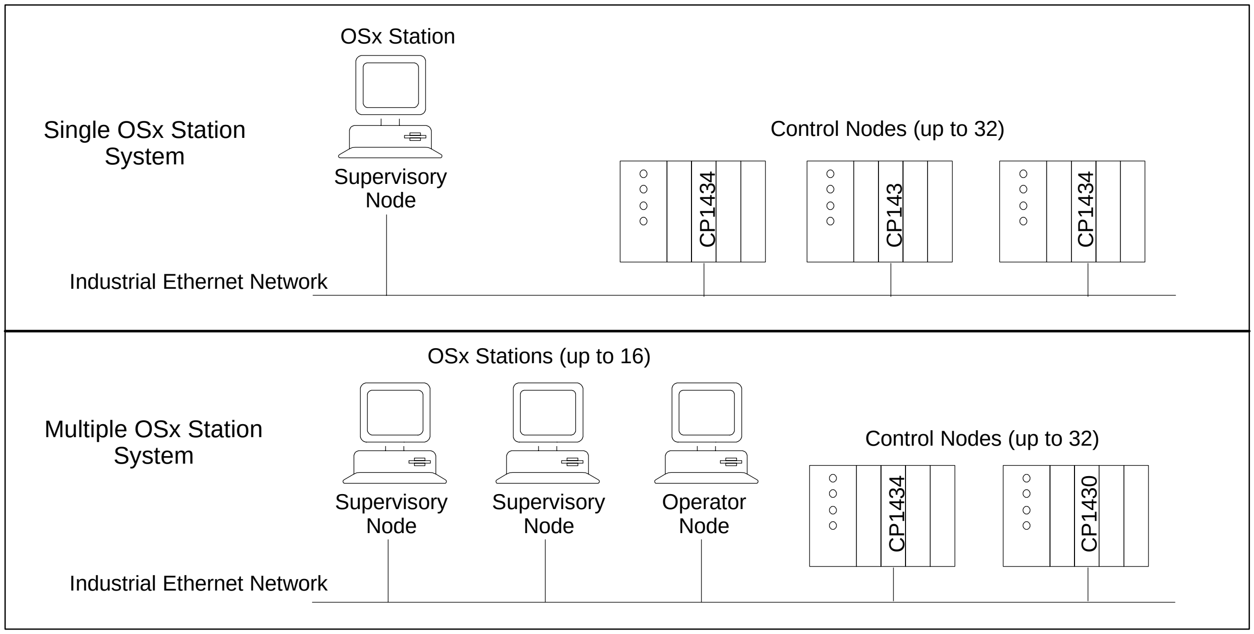


Figure 1-2 Examples of Single and Multiple OSx Station Systems

1.2 Powering Up an OSx Station

Your system administrator must do some preliminary configuration on an OSx station before you can begin entering configuration data for your application process. The *SIMATIC PCS 7 OSx System Administration Manual* describes this preliminary work.

When you power up a system to begin process configuration, an OSx station must complete the boot sequence and display the main screen. If this does not occur, refer to the *SIMATIC PCS 7 OSx System Administration Manual* or consult your system administrator.

NOTE: If a system does not enter the OSx environment at powerup, consult the *SIMATIC PCS 7 OSx System Administration Manual* and your system administrator. You must initialize an OSx station before it can enter the OSx environment.

Whenever you power up, or change states, you must wait for the system to transition completely to the requested state before you interact with OSx using the keyboard or mouse. Attempting to type while a state change is in process could result in the system either going to the Failed state or continually trying to go to the Failed state. The only way to recover at this point is to shut down the system and reboot.

To power up an OSx station, follow these steps.

1. Turn on the monitor.

2. Turn on the printer, if one is connected to the station.

3. Turn on the system unit. After a few moments, the system completes the boot sequence and displays the main screen.

NOTE: When you shut down the system and reboot, if you have a floppy disk that is not a boot disk in the disk drive, the only indication of this is a flashing cursor. The system does not proceed through the boot procedure and no error message appears.

To reboot properly, remove the floppy disk and turn the power switch off and on again.

1.3 Identifying Screen Areas

Main Screen Components

Figure 1-3 shows the elements of the main screen area.

Menu Bar Click any of the choices in the menu bar to display submenus containing additional configuration options.

Main Screen Area This is the main working area of the screen. Dialog boxes, graphics, and various other displays appear in this area.

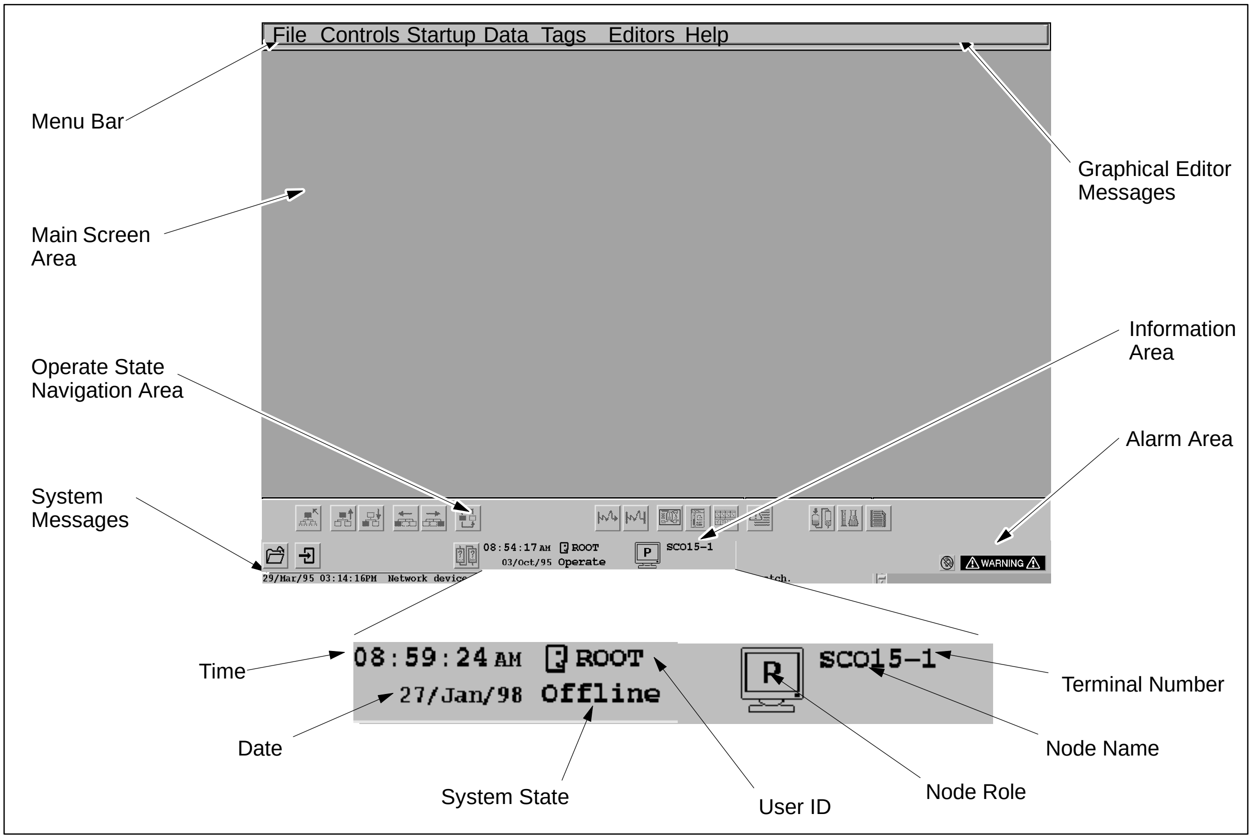


Figure 1-3 OSx Main Screen and Information Area

System Messages When you are using any system functions other than the graphical editor, OSx displays messages in this area. A scroll bar to the right of the system message line allows you to display the last five system messages. These messages are also included in the error file **log.out**.

Graphical Editor Messages When the graphical editor is running in the background mode, the screen displays graphical editor messages in this area.

Information Area This shows station status: User ID, date and time, system state, station name, role, and terminal number. The terminal number is 1 to designate a station, or 2–5 to designate X terminals or remote terminals, if any are connected.

The following codes indicate system information for the station:

P	= Primary
B	= Backup
A	= Sysadmin
Blank	= Normal role
X	= Out_of_service
T	= Transition (primary station only)
RE-SYNC	= Node is out of sync
	= Padlock indicates database is locked (station resynchronizing)
	= User ID

Alarm Area A station displays a flashing alarm icon if upset conditions occur when the system is in the Operate state. The alarm area also displays the alarm silence and action request icons.

Operate State Navigation Area The screen displays this area only in the Operate state. The operator clicks on the pushbuttons to display additional options.

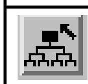
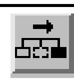
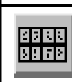
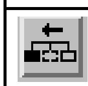
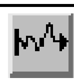
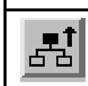
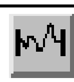
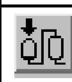
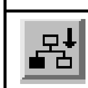
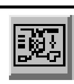
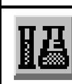
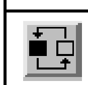
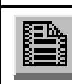
Table 1-1 shows the navigation area icons and the windows that they display. You can configure your system to display these icons with or without text labels defining their functions. Follow these steps.

1. Select **Startup->Event Preferences** from the menu bar. The Event Preferences dialog box appears.
2. In the Navigation Type field, select one of the following:

   Icon Only to display the icons without text (Figure 1-4, top).

   Icon with Text Descriptor to display the icons with text (Figure 1-4, bottom).
3. Click **OK** to save your configuration.

Table 1-1 Navigation Area Icons

Icon	Function	Icon	Function	Icon	Function
	Directory		Overview		Active Batches
	Home		Right		Tag Group
	Left		Real-Time Trend		Alarm Group
	Up		Historical Trend		Batch
	Down		Graphic		Recipe
	Previous		Tag Detail		Report

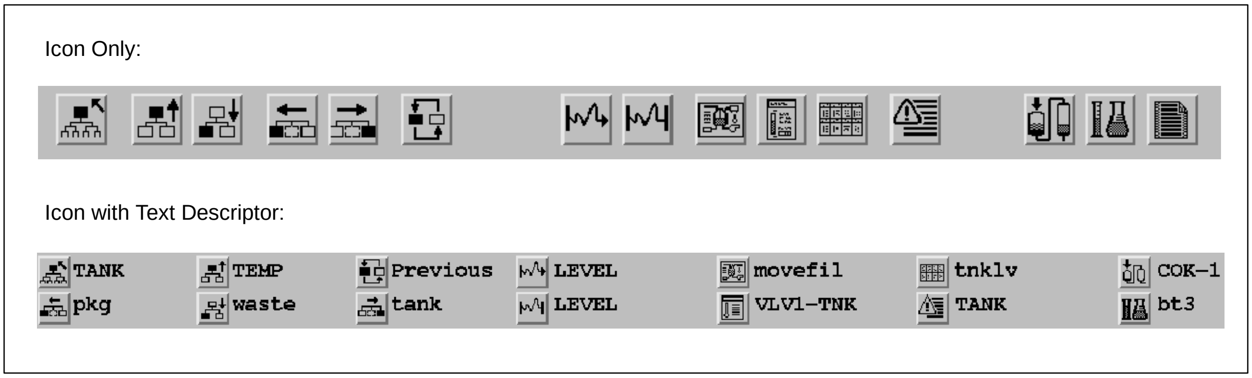


Figure 1-4 Navigation Area Display Options

Displaying the Runtime Directory

To access the runtime directory of OSx display types, the operator must follow these steps:

1. Set the system state to Operate. The screen displays the navigation area.

2. Click the **Directory** pushbutton from the navigation area. The screen displays a directory of OSx display types. (Figure 1-5).

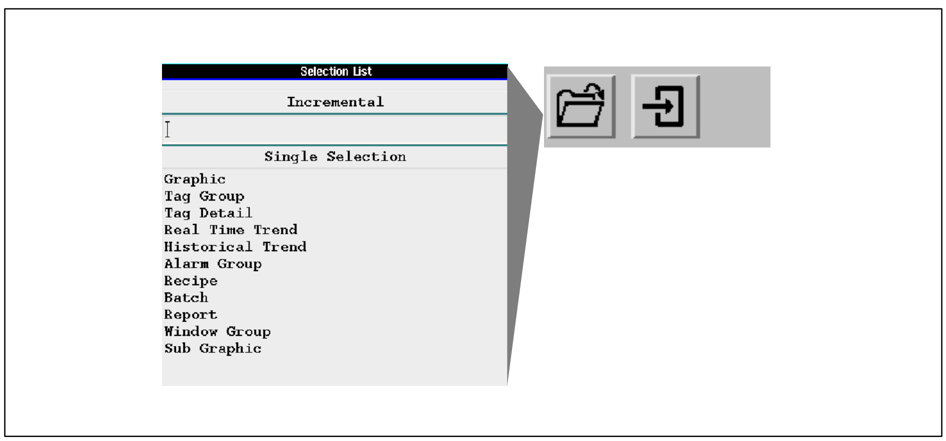


Figure 1-5 Navigation Area Directory

1.4 Working with Screen Elements

Working in Windows

OSx provides numerous navigational tools for working in the screen areas (Figure 1-6). These navigational tools appear as buttons, menus, sliders, or scroll bars.

Window Control Menu Callup Click this tool to display a menu of options. Select one of the following options by clicking on it or by pressing the indicated keys:

- **Restore** expands an icon to a window.
- **Move** allows you to change the position of the window.
- **Size** allows you to enlarge/reduce the window.
- **Maximize** sizes the window to its largest possible size and minimize reduces the window to an icon.
- **Lower** and **Raise Up** move a window above or below all other windows on the screen.
- **Refresh** causes all information on the screen to be redisplayed.
- **Close** dismisses the window.

NOTE: Some of the options listed above are not available in all OSx windows.

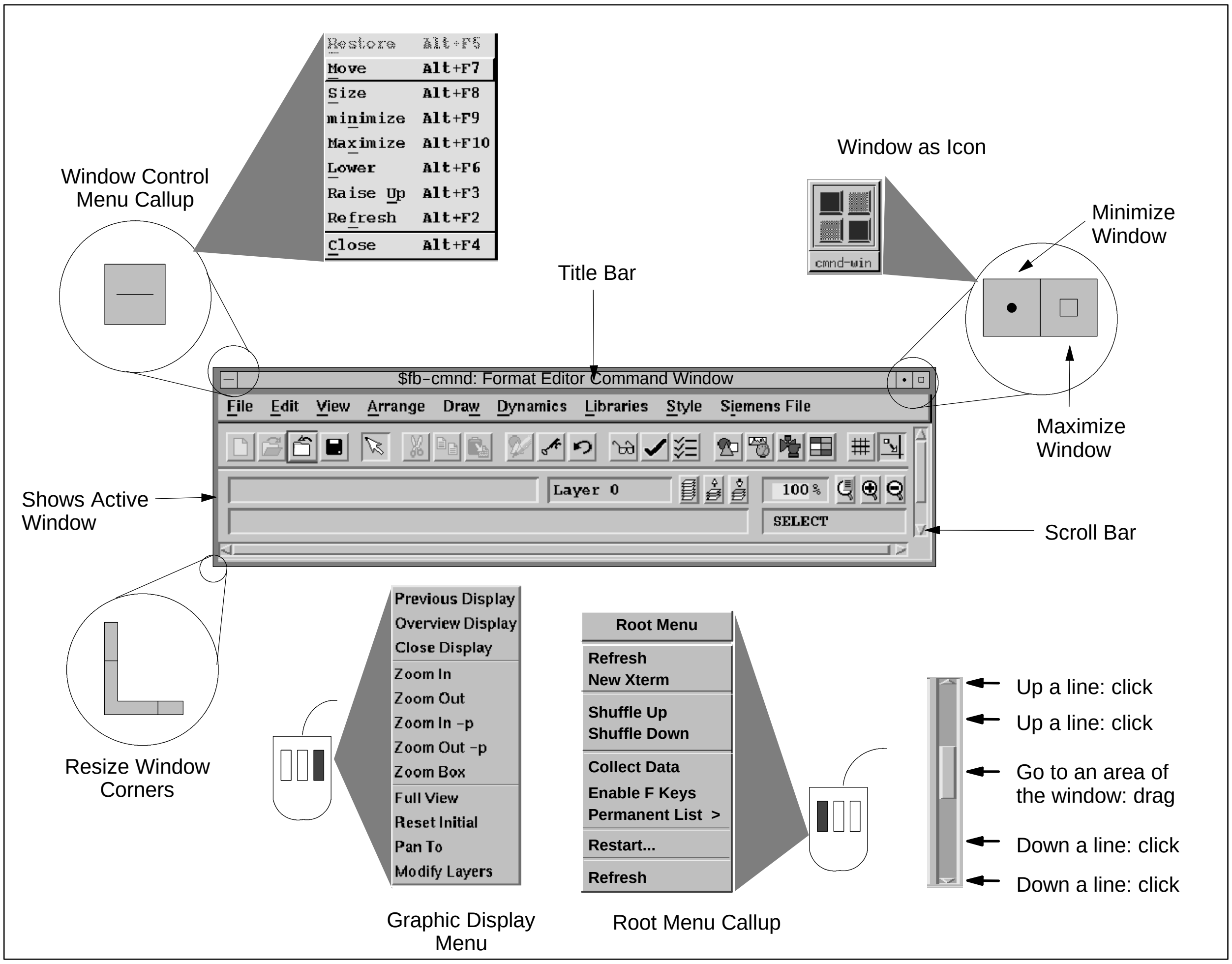


Figure 1-6 Screen Elements

Active Window The border around a window changes color to indicate that the window is active and that both keyboard and pointer operations apply to this window.

Root Menu Callup To display the root menu, place the cursor anywhere in the main screen area, on a point not occupied by another object, and hold the left button down. Select one of the following options by clicking on it:

- **Refresh** redisplays all information on the screen.
- **New Xterm** opens a terminal window. You can log on with an operating system User ID and execute instructions at the command line.
- **Shuffle Up** and **Shuffle Down** move all objects (windows and icons) in the same layer of the screen display up or down a layer. If you cannot see the icon or window that you need, select either **Shuffle Up** or **Shuffle Down** repeatedly until the object appears.
- **Enable F Keys** allows you to map function keys to graphics.
- **Collect Data** allows you to collect diagnostic system information in a file.
- **Permanent List** allows you to choose whether the select list for tag details, graphics, reports, and so on (accessed from the Directory button) remains on the screen until you dismiss it, or disappears when you select an entry:

  Front—the list remains on the screen in front of other objects
  Back—the list remains on the screen, but may be hidden from view
  Off—the list disappears after you select an entry

  For both **Front** and **Back** settings, the list is positioned in the lower right corner of the screen. **Front** is the default setting.

  To move the select list to the front or back at any time, you can use the **Shuffle Up** or **Shuffle Down** commands explained above, or press **Alt F3** (raise window) or **Alt F6** (lower window).
- **Restart** rereads the files that determine the X Window environment. If you change any environment variables, select **Restart** instead of rebooting the system.

Scroll Bar A scroll bar appears on the edge of a window when more information exists in the window than can be displayed at one time. Use it to retrieve information that is above or below the current location in the window.

Minimize Window If a window has this tool, click the tool to reduce the window to an icon. To redisplay the window, click the icon. The window control menu, described on page 1-12, is displayed.

Maximize Window If a window has this tool, click the tool to set the window to its largest possible size.

Resize Window Corners Place the cursor on the corner of the window and hold the left button down while moving the cursor to change the size of the window (Figure 1-7).

Title Bar Some windows have a title bar located at the top. Place the cursor on the title bar and hold down the left button to move the window (Figure 1-7).

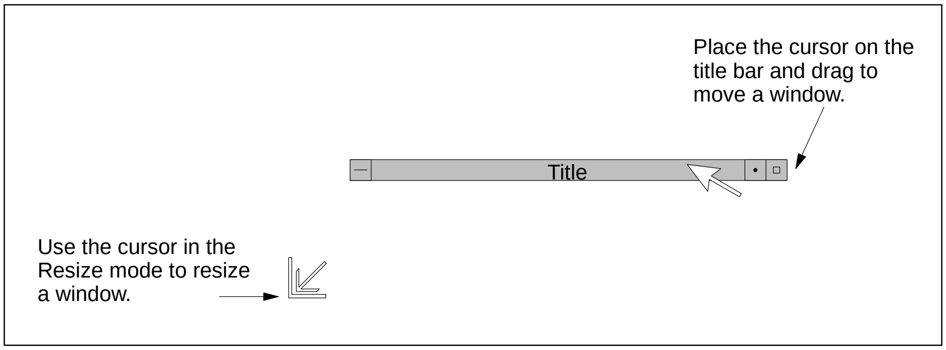


Figure 1-7 Moving and Resizing Windows

Graphic Display Menu To access the graphic display menu, place the cursor within any graphic display and click the right mouse button. Graphic displays include graphics, tag details, the individual faceplates of tag groups, and trends. You can select from the following menu items:

- **Previous Display** allows you to view the graphic display that appeared before the one that is currently being displayed.
- **Overview Display** allows you to view the initial graphic display that appears when you set OSx to the Operate state.
- **Close Display** closes the current graphic display.
- **Zoom In** magnifies (zooms) the graphic display to twice its previously displayed size. The center of the graphic becomes the center of the zoomed area. Scroll bars appear on the graphic display to enable you to move to areas that are outside the current viewing area.
- **Zoom Out** reduces (demagnifies) the graphic display to half its previously displayed size. The center of the previous display is the center of the new display.
- **Zoom In -p** zooms closer, centered on the point marked by your cursor when you access the menu. Scroll bars appear on the graphic display to enable you to move to areas that are outside the current viewing area.
- **Zoom Out -p** zooms back, centered on the point marked by your cursor when you access the menu.
- **Zoom Box** allows you to use a box outline to select an area to zoom on the current display. When you draw the box, the area inside it is scaled and panned, so that the contents of the box fill the graphic display.

  To draw the box, click the left mouse button in the graphic display where you want the upper left corner of the box to start, move the cursor down and to the right until the box encompasses the area that you want to zoom, and click the left mouse button again.

NOTE: Graphics that were created on PCS/OSx systems before Release 3.1.1 do not automatically have the graphic display menu. If you have such graphics and want to add this menu to them, contact the Siemens Energy & Automation, Inc., Technical Services Group in the U.S.A. at 800-333-7421. Outside the U.S.A., call 49-911-895-7000.

- **Full View** rescales the graphic display so that it fits in the display area. For example, if you resize a window to half its original size, only half of the original graphic display is visible. Clicking on **Full View** resizes the original graphic display so that it is visible in the half-size window.

- **Reset Initial** rescales the graphic display to its initial size.

- **Pan To** allows you to select the center point for panning a graphic display. This feature works only when scroll bars appear on the graphic display, indicating that some parts of the display are hidden from view (for example, when you have zoomed in on part of the graphic display).

- **Modify Layers** brings up the Layer Properties dialog box. The Layer Properties dialog box allows you to restore the current layers or all layers of graphic displays on your screen (Figure 1-8). If you click an item in the Layer List to highlight it, you can toggle the **Restore Current Layer** button to make the display visible or invisible.

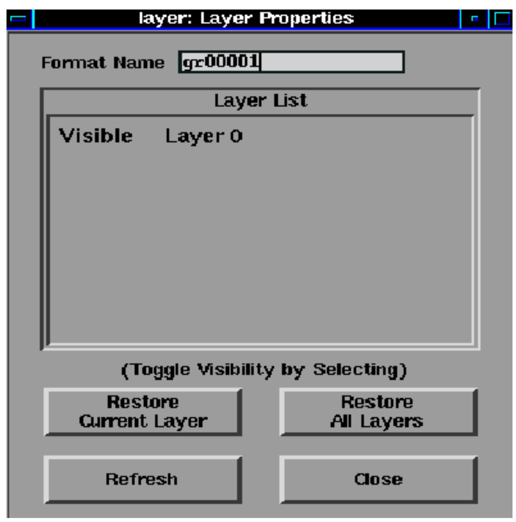
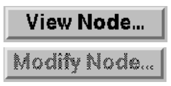


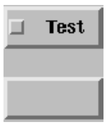
Figure 1-8 Layer Properties Dialog Box

Using Navigational Tools to Enter Data

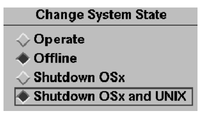
Use the navigational tools to display menus and lists, choose options, and enter data into the system.



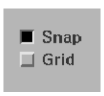
Pushbuttons Place the cursor on a pushbutton and select it to display a menu or another window. When the button text is bold, the pushbutton is active. When the button text is gray, you cannot select the button.



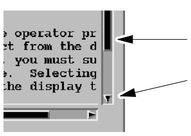
Some buttons have either no text or text adjacent to the button. Clicking on the button causes no action when the button is inactive.



Radio Buttons These are specially linked buttons. When a second button in the group is selected, the first automatically turns off. As a safeguard, you are always required to confirm your selection by clicking on an **OK** button.



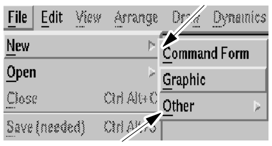
Toggle Buttons These buttons are not linked. Each one must be clicked off and on independently; changing the status of one has no effect on the other.



Text Browsers These tools allow you to see information not visible in a window. Place the cursor on the bar and slide the bar back and forth while holding down the left mouse button. You can also click the pointers at each end of the bar groove to display single lines or columns.

Sliders These tools allow you to see an increase or decrease in an array of values. That is, as you move the slider, the values go up or down. Place the cursor on the bar and slide the bar back and forth while holding down the left mouse button.

Data Entry Field Click a field in which you want to enter information. Some windows require an entry in one field before you can enter information in another field. Therefore, if one field does not appear to be active, move the cursor to the field that precedes it in the window.



Cascaded Menus When menus appear, you choose a given item by clicking on it. Some menu items have cascaded menus that appear when you slide the cursor to follow the triangle indicator. When you cannot select an item because of the current operation or display, the text of the item is gray.

The following five pushbuttons appear in most of the dialog windows used in OSx. Their action does not change from window to window.

OK Saves information that you have entered but not yet saved and closes the window.

Save Saves information that you have entered and does not close the window.

Cancel Closes the window without saving any information that you entered and terminates any action that you initiated.

Dismiss Closes the window without undoing any changes that you have made.

Help Displays a context-sensitive help window.

Short-List Display Tool Place the cursor on the field adjacent to a short-list display tool and click the right button to display a list of items that you can use in that field (Figure 1-9). After the list appears, you can select an item by clicking on the item.

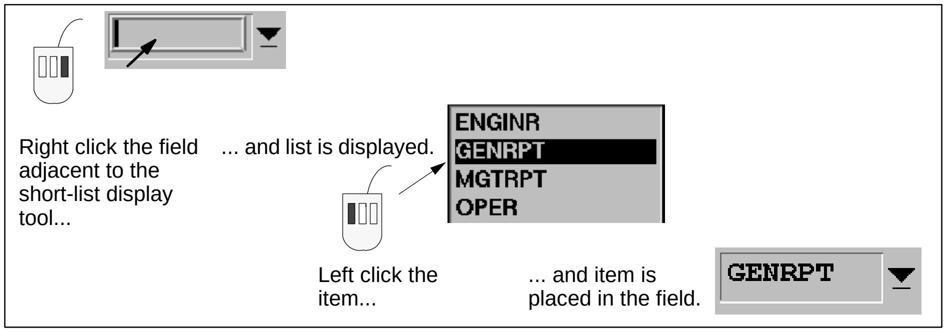


Figure 1-9 Displaying a Short List of Items

NOTE: You can select items on the short-list and long-list display tools and then use the scroll bar to locate other items in the list. Items that you have selected can scroll off the list and remain selected. Any operations that you execute on selected items affect all selected items, including those not currently showing on the list.

Long-List Display Tool Place the cursor on a long-list display tool button and click the left mouse button to display a list of tags that you can use in that field (Figure 1-10). After the list appears, you can select a tag by double-clicking on the tag.

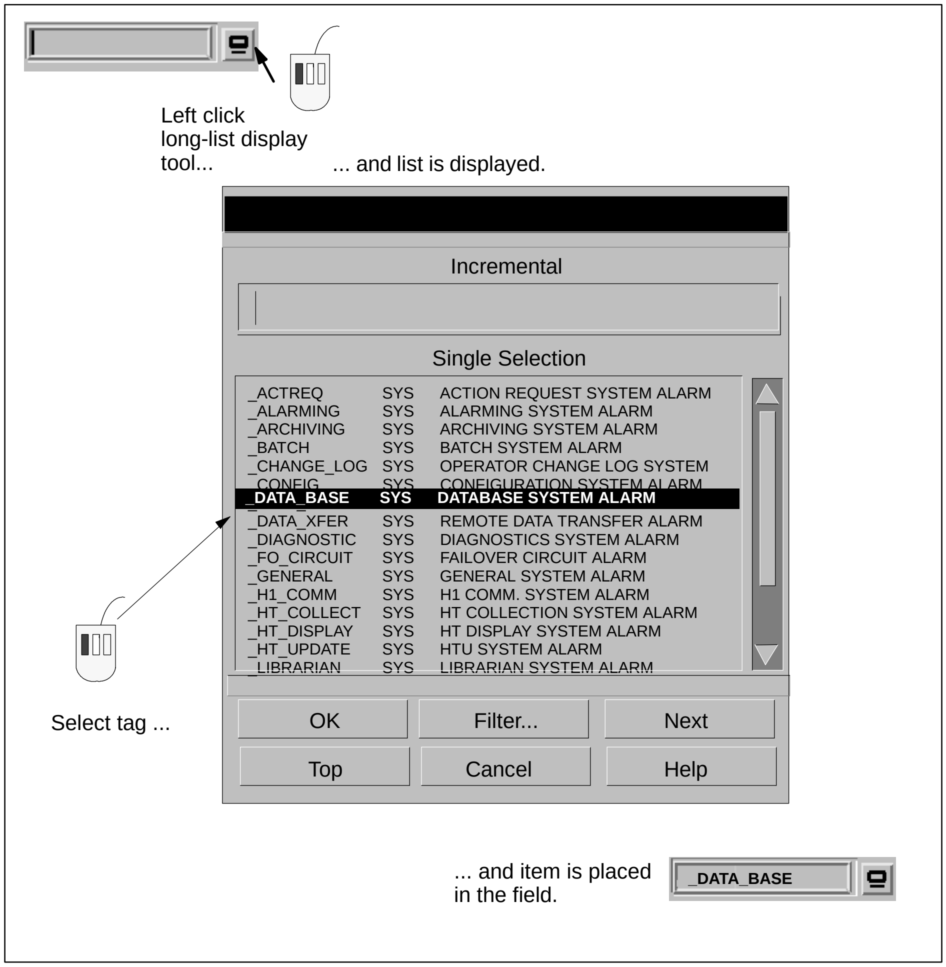


Figure 1-10 Displaying a Long List of Items

When you click a long-list display tool to display a list of tags (Figure 1-11), the list can be a lengthy one of several hundred items or more, sorted alphabetically. Click the **Top** button to reset the search to start at the top of the list.

To locate an individual tag, choose one of two methods: Incremental or Filter. Incremental is the default search operation in a list, except in the graphical editor, where Filter is the default. If you are selecting a tag from a configuration utility, you can click the **Filter...** toggle button to display the Tag Filter dialog box. The following pages describe these search methods.

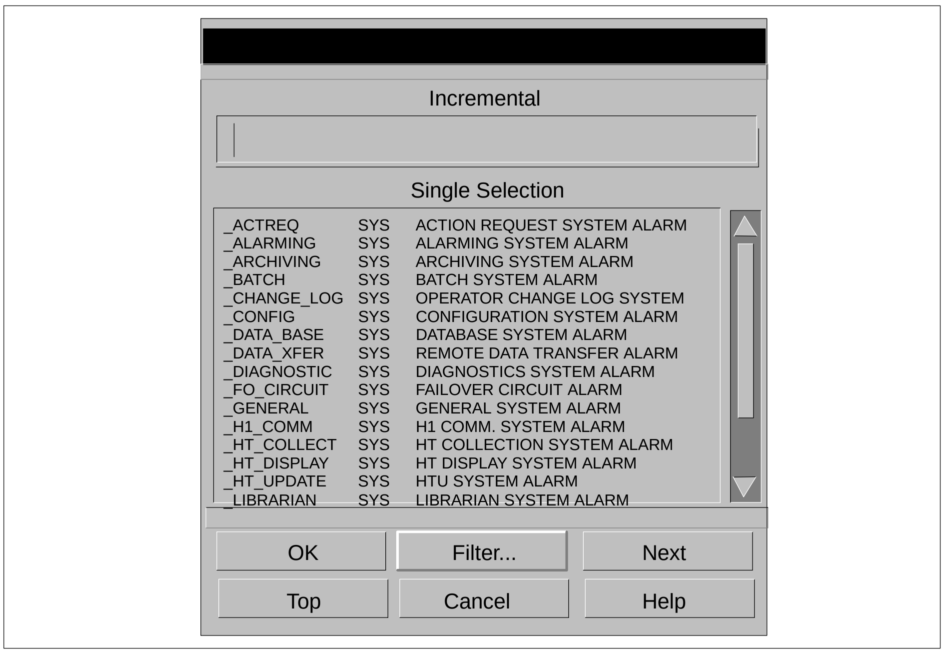
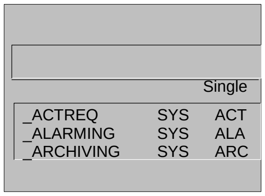


Figure 1-11 Displaying the Long List Default Screen



Long-List Display Tool: Searching Incrementally As you type in characters, the cursor jumps to the first name in the list that begins with the characters entered (Figure 1-12). With each subsequent character that you type, the cursor jumps to the next valid name. The incremental search tool is not case sensitive.

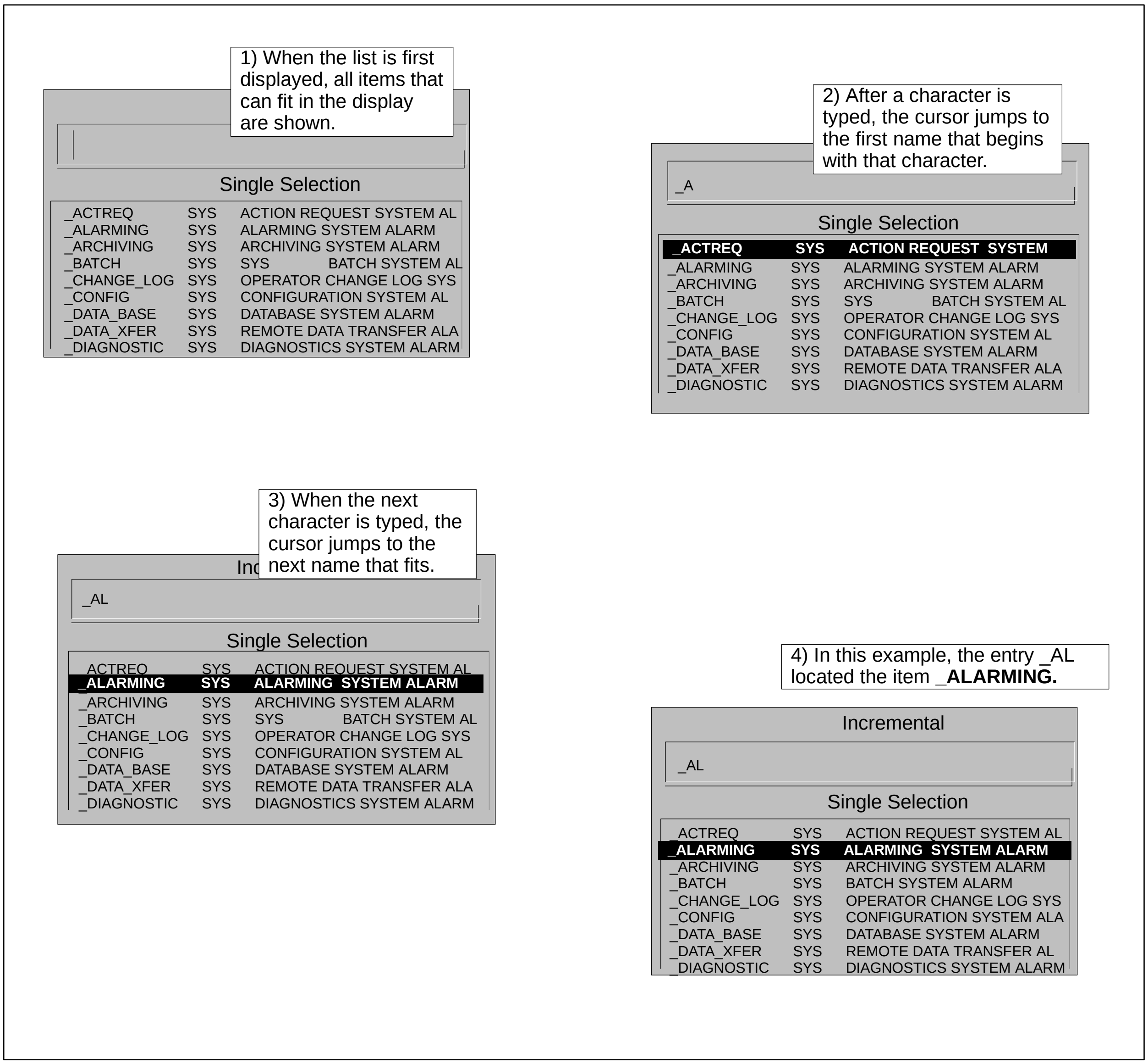
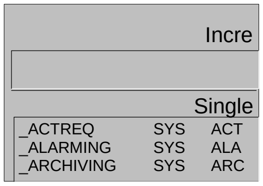


Figure 1-12 Locating an Item in a Long List: Incremental Method



Long-List Display Tool: Using the Tag Filter You can use the Tag Filter dialog box to locate the tag that you need. The tag filter is not case sensitive. To locate a tag or tags using the tag filter, follow the steps below.

1. Click the **Filter...** toggle button at the bottom of the selection list. The Tag Filter dialog box appears (Figure 1-13).

2. To begin a search, select **Name**, **Type**, or **Description**, then select **Ascending** or **Descending**.

3. Type the appropriate text (up to 30 characters) in the field at the bottom of the box, and press **Enter**. The filter process begins.

   For example, to see all the tags that have the letters HT in the name, sorted in descending order by tag name, select **Name,** select **Descending**, and then type `ht` in the search field and click **OK**. The resulting list contains only tag names with the letters `ht` (Figure 1-13). The list is sorted in descending alphabetical order (Z to A) by tag name.

4. To filter the resulting list, choose **Filter...** again, then enter the second search criteria in the Tag Filter dialog box. You can continue this progressive filtering, using different search criteria each time.

   For example, to filter the `ht` list in Figure 1-13 so that only the **_HT_DISPLAY** tag appears, click the **Filter...** toggle button at the bottom of the `ht` selection list. When the Tag Filter dialog box appears, select **Description**, then type `display` in the search field (Figure 1-14), and click **OK**. The filter process begins.

5. To return to the original selection list, click the **Filter...** toggle button at the bottom of the selection list, then select the **Reset** button at the bottom of the Tag Filter dialog box (Figure 1-15).

   Remember that the filter settings remain in effect until you click **Reset**. Always click **Reset** before beginning a new search to ensure that you are accessing the complete tag list.

   To exit the Tag Filter dialog box without filtering the current selection list, select the **Cance**l button at the bottom of the Tag Filter dialog box.

NOTE: The Tag Filter dialog box times out after one minute. The selection list times out after one minute, and the screen returns to the original selection list.

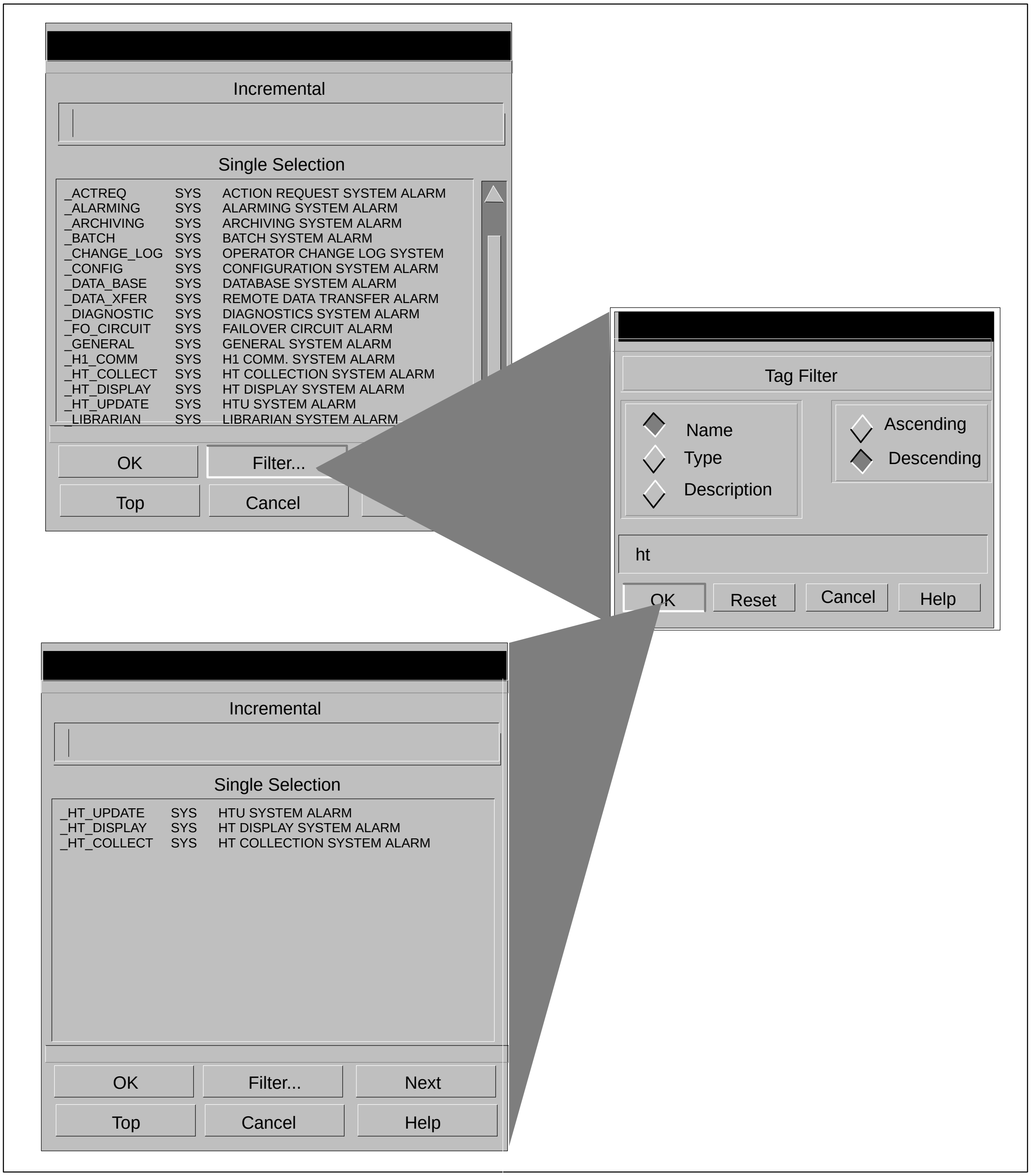


Figure 1-13 Using the Tag Filter

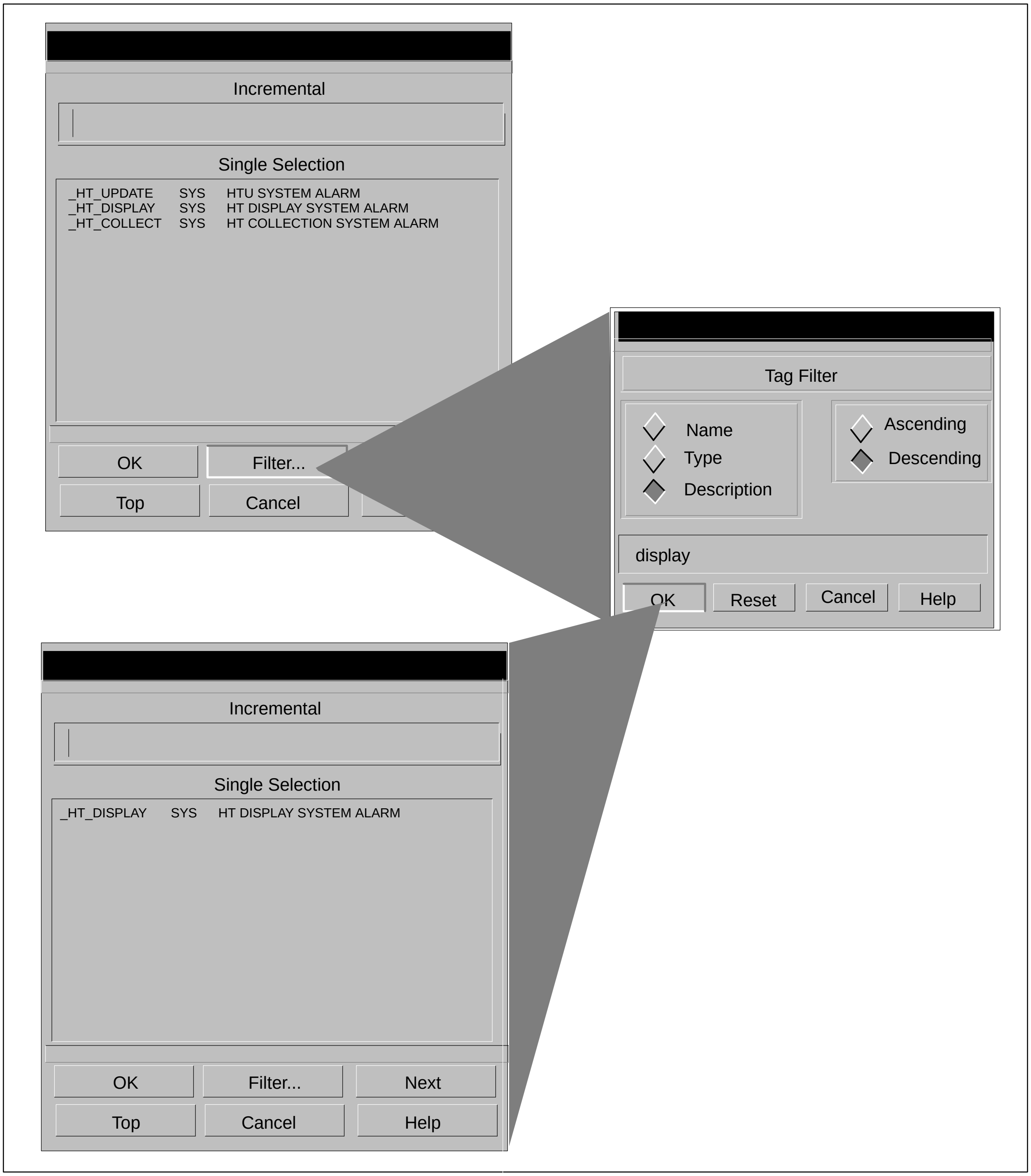


Figure 1-14 Filtering Using a Second Variable

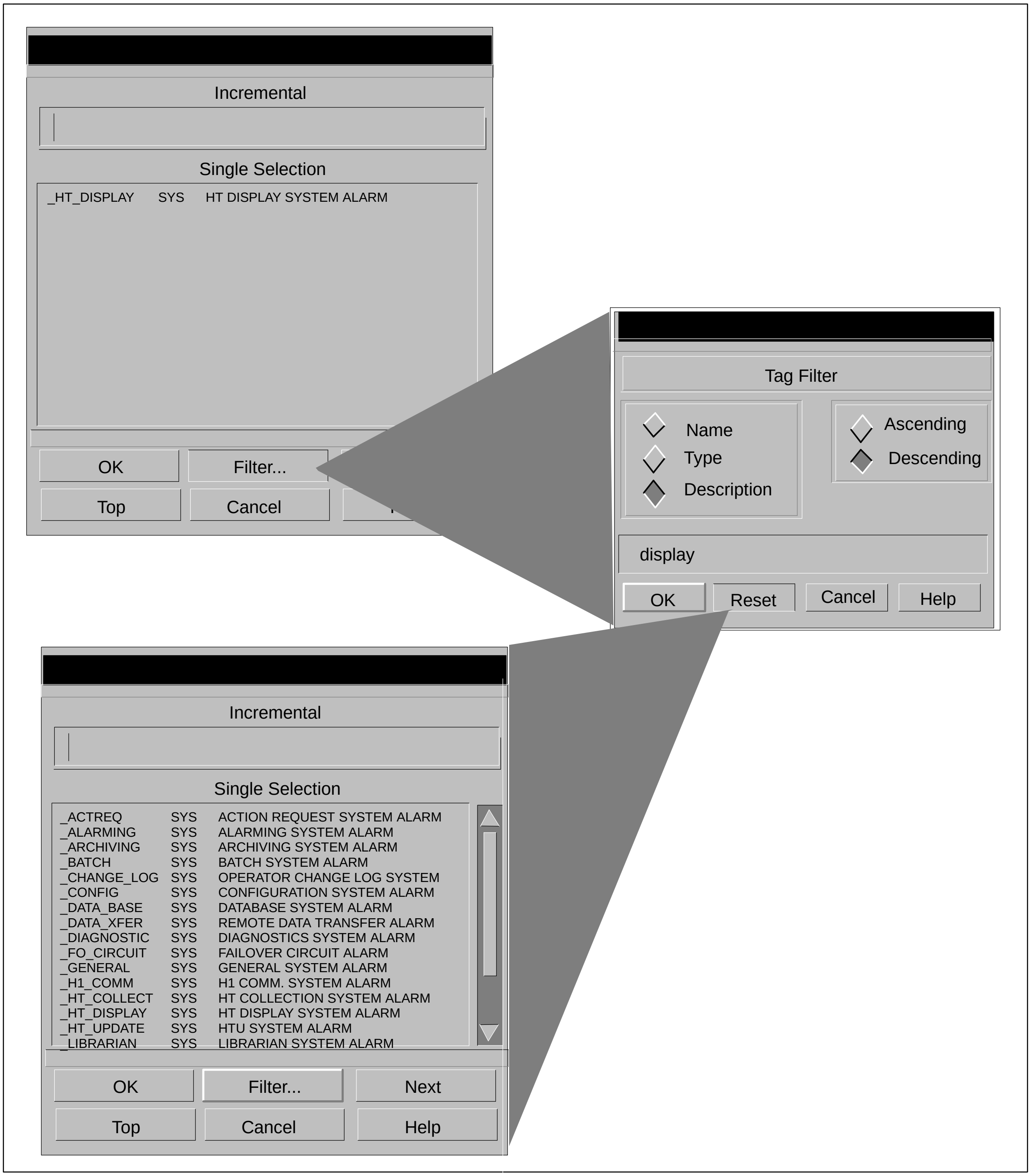
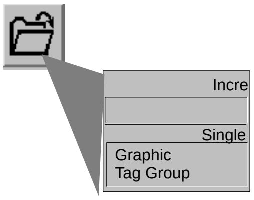


Figure 1-15 Resetting to the Original Tag List



Filtering Directory Lists When you display a list of non-tag entries from the directory, the default search tool is the incremental data entry method. The **Filter** button at the bottom of the selection list operates differently than the **Filter...** button on the long-list display tool tag selection list. To use this filter, click the **Filter** toggle button so that the **Filter** option becomes active. Follow the steps below:

1. Enter the search pattern in the filter search field. This **Filter** option is case sensitive, so be sure upper- and lower-case letters match the entry you are looking for. Figure 1-16 lists the rules for using the filter.
2. Press **Enter**. The cursor jumps to the first occurrence of the pattern.
3. Select the **Next** button to locate the next occurrence of the pattern.
4. When you find the entry that you are looking for, click on the entry to highlight it, and then click **OK**.

To search for a text string that appears anywhere in the name of the item, enter the text pattern in the filter search field. For example, enter `DO_5` to find the first occurrence of the string **DO_5**.	DO_5	DO_48 DO_49 DO_5 DO_50 DO_51
To do a search using a wildcard for a single character, use a period in the search pattern in the filter search field. For example, enter `.TR` to find the first occurrence of any of the following patterns: **CTR_0**, **CTR_1**, **CTR_2,** ...	.TR	CALC_8 CALC_9 CTR_0 CTR_1 CTR_150
To search for a pattern that matches the beginning of a string of text, use a caret in the search pattern in the filter search field. For example, enter `^DI` to find the first occurrence of a string that begins with **DI**. To match the end of a string, use a `$`. To find **AI_2**, enter `_2$` in the search field.	^DI	CHG_VALUE2 CHG_VALUE3 DI_0 DI_1 DI_2
To search for a string that contains any one of several characters, place the characters in brackets. For example, enter `[23]` to find the first occurrence of a string that contains **2** or **3**.	[23]	AI_10 AI_11 AI_12 AI_13 AI_14
To search for a string that contains a range of characters, place the pattern in brackets and indicate the range with a dash character. For example, enter `[2-4]` to find the first occurrence of a string that contains **2**, **3**, or **4**. In the list shown to the right, this filter search string locates DI_2, DI_3 and DI_4.	[2–4]	DI_0 DI_1 DI_2 DI_3 DI_4
To search for a character that appears one or more times in a string, use a plus sign in the search pattern in the filter search field. For example, `[3]+` matches **3**, **33**, **333**, ... [z]+9 matches **z9**, **zz9**, **zzz9**, ...	[3]+	C1AI_31 C1AI_32 C1AI_33 C1AI_34 C1AI_35

Figure 1-16 Locating an Item from the Directory: Filter Method

1.5 Logging On/Off

After powering up the system as described in Section 1.2, you can log on and begin configuration. To log on to OSx, follow these steps:

1. Select **Controls->Logon** from the menu bar (Figure 1-17).
2. Enter your User ID in the **Name** field. Use lower-case letters.
3. Enter your password in the **Password** field.
4. Click the **OK** button. The system logs you on.

NOTE: User IDs and passwords are linked with specific security privileges to prevent unauthorized use of the SIMATIC PCS 7 OSx system. Your system administrator or process configuration engineer can assign you a User ID and password.

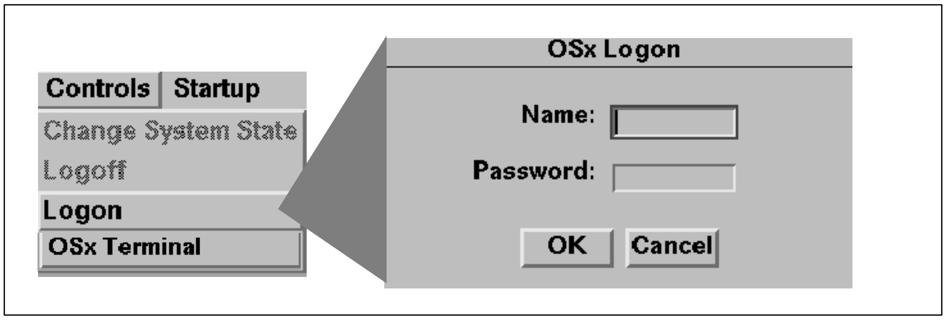


Figure 1-17 Displaying the Logon Menu

When you have finished working, log off the system. If you do not log off, others can use the system under your ID. To log off the system, select **Controls->Logoff** on the the menu bar.

1.6 Recommended Order of Tasks

For the most efficient configuration of your OSx system, do the tasks shown in Figure 1-18 in the order listed.

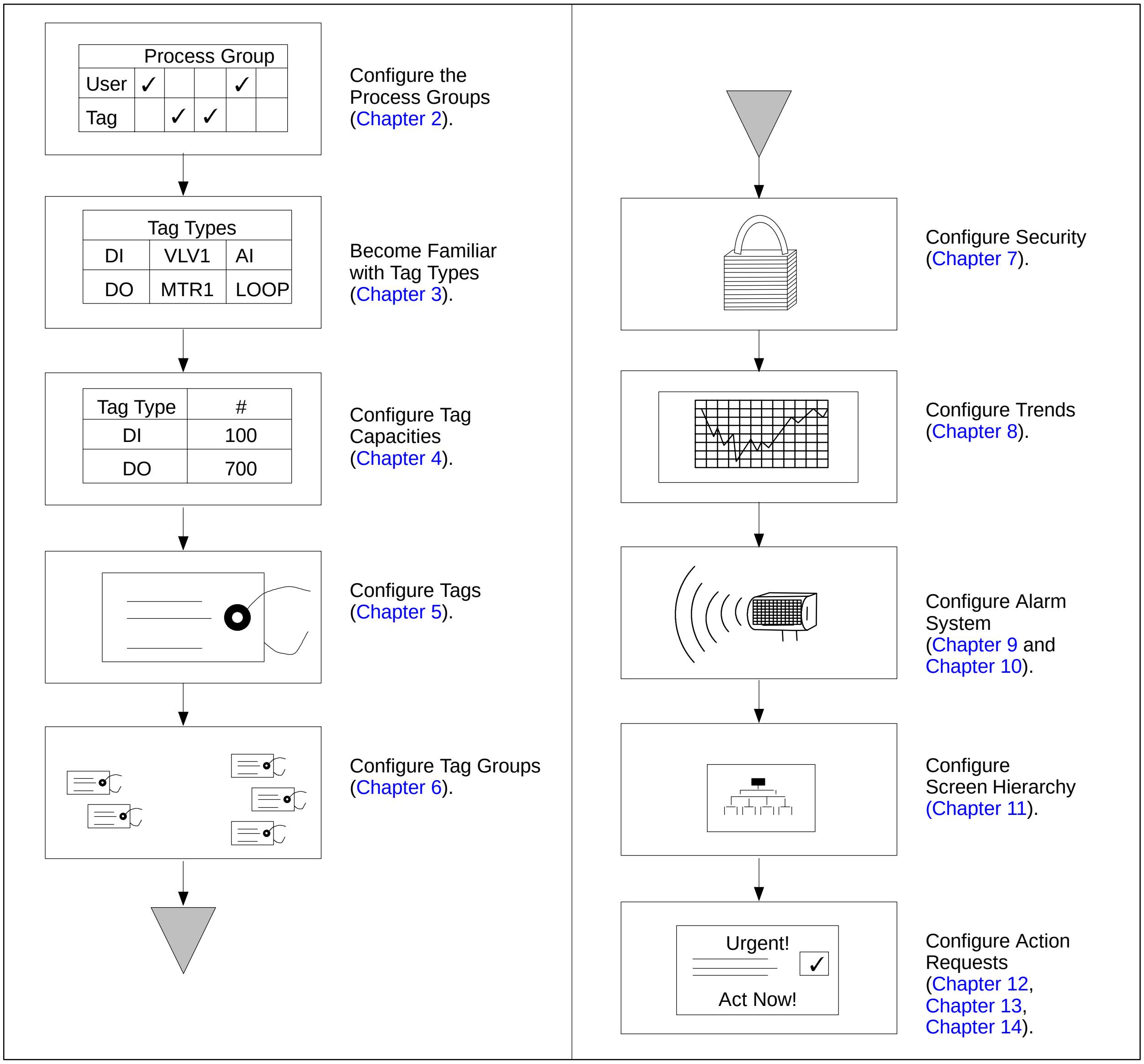


Figure 1-18 Recommended Order of OSx Tasks

1.7 Using the Configuration Planning Sheets

OSx is designed to simplify the process of configuring a system. The software provides easy-to-use menus and screens. Planning sheets, found in Appendix A, provide a means of organizing the configuration and recording system information, thereby saving you a significant amount of time in the data-entry process. Figure 1-19 shows an example of a planning sheet.

Use the planning sheets in conjunction with your review of each chapter in this manual. This exercise should help you comprehend the level of information required for each task and make the overall process more efficient.

PROCESS I/O: DEVICE
DEVICE CONFIGURATION PLANNING SHEET

Sheet ___ of ___

VLV1 (single-feedback valve), VLV2 (dual-feedback valve)
MTR1 (motor), MTR2 (two-speed motor), RMTR (reversible motor)

CONTROL NODE:

TAG NAME:

DESCRIPTION:

PROCESS GROUP:

PARENT:

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	AUTOLOG	INITIAL VALUE
STATUS			YES NO	YES NO	
OVERRIDE			YES NO	YES NO	
MODE_CMD			YES NO	YES NO	
SETPOINT			YES NO	YES NO	
TIMEOUT_1			YES NO	YES NO	
TIMEOUT_2			YES NO	YES NO	

Figure 1-19 Example of a Planning Sheet

1.8 Understanding System States

The system state determines the actions you can do while OSx is running. The default system state is Offline. Four system states are user selectable: Offline, Operate, Shutdown OSx, and Shutdown OSx and UNIX (Figure 1-20 on page 1-34). Two other states, Powerup and Standby, are used as the system changes from one user-selected state to another. The system enters the Resize state automatically when you make changes in the tag capacity. A Failed state occurs in the event of a fatal software or hardware error. The following list shows the system states and their descriptions:

NOTE: When you change system state, all operations stations take the new state except for out_of_service stations. In addition, state changes on an out_of_service station do not affect other stations.

Offline The Offline state supports all configuration functions except tag capacity configuration.

Operate The Operate state supports collecting data, monitoring process states, and all configuration functions except for resizing the database and certain deletions.

Shutdown OSx Select the Shutdown OSx state to shut down OSx. You can still execute operating system commands.

Shutdown OSx and Linux Select the Shutdown OSx and Linux state to shut down both the OSx system and the Linux operating system. Use this option before you power down or reboot the system. For multiple-station systems, if you shut down OSx and Linux on a station that is not out of service, that station exits OSx and Linux and displays the prompt that it is safe to power off or reboot. All the other stations in the system except for out of service stations also exit OSx and Linux when you select this option.

Powerup, **Standby** Powerup and Standby are transition states.

Resize When you change tag capacities, the system starts from the Offline state, enters the Resize state, makes the designated modifications to the system tag capacities, and then returns to the Offline state. On a multiple-station system, you can change tag capacities only on a station with the sys_admin role. On a single-station system, which does not support the sys_admin role, you can change tag capacities on the primary.

Failed The Failed state occurs when a fatal software or hardware error exists. The system allows you to execute diagnostics in the Failed state.

1.9 Powering Down an OSx Station

To preserve the integrity of the process data, you must shut down the system in an orderly manner when you need to turn off the power. To shut down the system, ensure that you are logged on with the appropriate security privilege. Always shut down the primary last and reboot the primary first in a multiple-station system.

Select **Controls–>Change System State** from the menu bar, then select **Shutdown OSx and Linux** (Figure 1-20). On a single-station system, the station exits OSx and Linux and displays a prompt that it is safe to power off or reboot. For multiple-station systems, if you shut down OSx and Linux on a station that is not out_of_service, that station exits OSx and Linux and displays the **Power down** prompt. All the other stations in the system except for those that are out_of_service also exit OSx and Linux. If you shut down OSx and Linux on a station that is out_of_service, only that station exits OSx and Linux.

After you see the **Power down** prompt, follow the steps below.

1. Turn off the system unit.

2. Turn off the monitor.

3. Turn off all external devices, such as printers.

CAUTION
Database information can become corrupted if you do not shut down the system by the procedure described above. **Correcting this situation can require modifications of file information from within the operating system. Factory help may be required.** **Always shut down the system according to the described procedure.**

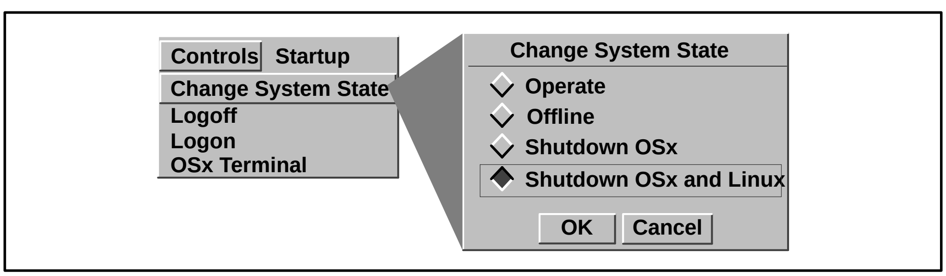


Figure 1-20 Shutting Down the System

Emergency Shutdown

A restricted operating system account has been added that allows you to shut down an OSx station when the main menu is not present. When you log in as `shutdown`, you automatically begin an operating system shutdown. You can continue with a complete operating system shutdown or abort the process. If you choose to abort, the shutdown account times out. If you choose to continue, the shutdown continues to a safe state.

The shutdown account must be activated by the system administrator before it can be used. The account only needs to be activated once. To activate the shutdown account, follow these steps:

1. From **Controls->OSx Terminal** or from a new XTerm window, log in to the target station as `root`.
2. At the # prompt, type `passwd shutdown` and press **Enter**.
3. When the prompt **You can choose whether you pick a password, or have the system create one for you** appears, select the default (1) to assign a password. Do not use a NULL or blank password. The screen prompts you to re-enter the password for verification.
4. Type `exit` to log off. The shutdown account is now enabled.

To use the account, log in from an XTerm window, using `shutdown` as the login ID and the assigned password as the password. As with any account, the password should be given only to individuals entrusted with the shutdown privilege.

Chapter 2

Configuring Process Groups

2.1 Process Group Overview

Defining a Process Group

A process group is a collection of tags, alarms, action requests, printers, reports, log messages, or User IDs that you can associate with each other for reasons of function or security. To create or modify process groups, you must have either the Database Administration or the System Configuration security privilege.

- You can configure the association of alarm annunciation and acknowledgement by plant area. In that way operators are only alerted to alarms that occur in their area of responsibility, and they are not required to acknowledge or silence alarms that are in another area of the plant.
- You can segregate control authorization by plant area or User ID. In that way an operator in one area of the plant is allowed to start and stop equipment in one specific area, but not in other areas.
- You can configure the printing of all reports and log messages from one process or plant area to be sent to a particular printer.

You can assign all tags, alarm groups, User IDs, and action request memberships in a single process group. SIMATIC PCS 7 OSx allows you to configure up to 32 different process groups. The memberships that you assign are not a function of the station at which a user logs on, but rather follow the User ID. An operator can log on a station anywhere in the plant and the permissions and process group memberships assigned to that operator's User ID determine the actions the operator can take.

The process group feature is always operable. You do not enable or disable it. If you do not want to exclude users, tags, or alarm groups from a process group, it is not necessary to enable their membership. By default, all users, tags, and so on, are members of all 32 process groups.

When you use the process group feature, you can assign descriptions to one or more of the 32 available process groups. Then, when you are configuring a particular object, you can choose whether you want to disable the membership in the various process groups.

Configure the process groups before you assign individual tags, User IDs, alarms, action requests, reports, and printers to the process groups. This avoids the necessity of reconfiguring items to assign them to specific process groups.

Follow the steps in Figure 2-1 in the recommended sequence.

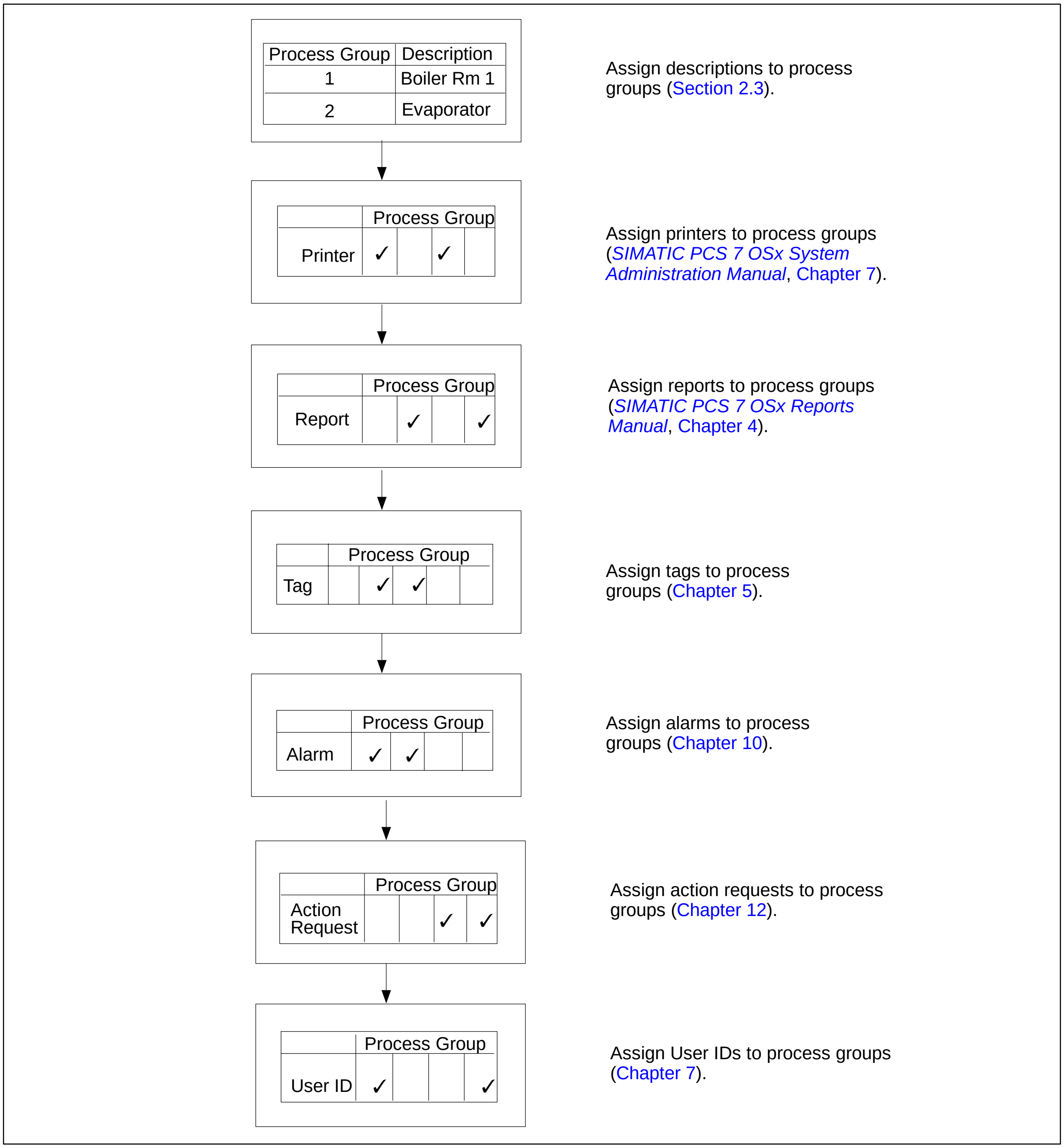


Figure 2-1 Order of Process Group Configuration Tasks

Planning Process Group Links

Before configuring any objects (printers, reports, tags, alarm groups, action requests, or User IDs), create a matrix containing the objects to which you want to assign membership for the various process groups. Examine the matrix carefully to determine which process-group memberships you need to assign. When you have determined all the memberships, you can begin to configure the various objects. As you plan your process group memberships, consider these guidelines:

- Process group membership for tags, alarm groups, action requests, and User IDs are designed for both task segregation and security control, so that operators see and respond only to events in the areas that concern them.
- Process group membership for printers, reports, and log messages are designed strictly for task segregation, so that, for example, all reports and log messages for a particular area of the plant go to one printer. User IDs do not affect printers, reports, and log messages.
- Reports are printed only if they have a process group in common with the printer. If you do not associate a report with any process group, then it will not print. If you do not associate a printer with any process group, you cannot print anything to it from OSx.
- Log messages are printed only if they have a process group in common with the printer. Alarm and action request messages are printed based on the alarm group or action request sharing a process group with the printer. Other log messages that are based on tags are printed if the tag shares a process group with the printer. Batch messages and any log messages that do not have tags default to “all process groups” and will always print to any printer that is configured with at least one process group.
- A tag inherits process groups from its parent tag. If you specify a parent tag for a tag that you are installing, the process groups for the tag will be any process groups that you specify for this tag, plus all the process groups of the parent tag.

Section 2.3 of this chapter describes how to define text descriptions for each of the process groups. However, you create the actual link between a process group and an object when you configure the object, not when you enter a description. As you configure each object, select the appropriate process groups for the object. Refer to Chapter 5 for tags, Chapter 7 for User IDs, Chapter 10 for alarms, Chapter 12 for action requests, and Chapter 14 for log messages. For printers, see Chapter 7 in the *SIMATIC PCS 7 OSx System Administration Manual*. For reports, see Chapter 4 in the *SIMATIC PCS 7 OSx Reports Manual*.

Table 2-1 illustrates how you can assign membership to various objects to provide the kind of control that you want for a process. The example uses six process groups, but you can configure up to 32.

Table 2-1 Linking Users and Tags

User	**Process Group**					
	Prcss. Area 1	**Prcss. Area 2**	**Prcss. Area 3**	**Rcv. Area 1**	**Rcv. Area 2**	**Stock Area 1**
John	X	X	X			
Carol	X		X	X	X	X

Tag						
Boiler_1_lev	X	X				
Boiler_2_lev		X	X			
Boiler_3_lev				X	X	

Alarm Group						
Fire_RM_1	X	X		X		
Fire_RM_2				X	X	X

Action Request						
Add H_2O				X		
Set Temperature					X	X

2.2 Using the Process Group Links

Using the User/Tag Link

You can link specific users and tags so that an operator can only act on those tags that are in a particular area of responsibility. Examine the sections labeled User and Tag in Table 2-1.

In Table 2-1, John can view and modify tags Boiler_1_lev and Boiler_2_lev because they are linked through the process groups listed below:

- John and Boiler_1_lev are members of process groups Process Area 1 and Process Area 2.
- John and Boiler_2_lev are members of process groups Process Area 2 and Process Area 3.

John can view, but not modify, tag Boiler_3_lev because he is not linked to it by any process group.

In Table 2-1, Carol can view and modify tags Boiler_1_lev, Boiler_2_lev, and Boiler_3_lev because they are linked through the process groups listed below:

- Carol and Boiler_1_lev are members of process group Process Area 1.
- Carol and Boiler_2_lev are members of process group Process Area 3.
- Carol and Boiler_3_lev are members of process groups Receiving Area 1 and Receiving Area 2.

NOTE: You can assign a user all the security privileges, but the user cannot access an object if the User ID and the object are not in the same process group. Additionally, if you assign a user and an object to the same process group, the user must still have the required security privileges to modify the object.

Using the User/Alarm Group Link

You can link specific users and alarm groups so that an operator is alerted only to those alarms that occur in a particular area of responsibility.

Refer to Table 2-2, which is a copy of the table shown in Section 2.1, and examine the sections labeled User and Alarm Group.

Table 2-2 Linking Users and Alarm Groups

User	Process Group					
	Prcss. Area 1	Prcss. Area 2	Prcss. Area 3	Rcv. Area 1	Rcv. Area 2	Stock Area 1
John	X	X	X			
Carol	X		X	X	X	X
Tag						
Boiler_1_lev	X	X				
Boiler_2_lev		X	X			
Boiler_3_lev				X	X	
Alarm Group						
Fire_RM_1	X	X		X		
Fire_RM_2				X	X	X
Action Request						
Add H_2O				X		
Set Temperature					X	X

Using the Process Group Links (continued)

In Table 2-2, John is alerted when alarm group Fire_RM_1 goes into alarm and can acknowledge alarms for tags within this group. John can modify tags Boiler_1_lev and Boiler_2_lev because his User ID has been assigned the appropriate security privilege, and because his User ID and these two tags are members of the same process group.

John is not alerted (his OSx station does not display the alarm icon) when alarm group Fire_RM_2 goes into alarm because he and Fire_RM_2 are not members of the same process groups. John cannot modify tag Boiler_3_lev for the same reason.

NOTE: John can view, but not acknowledge, alarms in alarm group Fire_RM_2 by displaying the Alarm Group Display.

In Table 2-2, Carol is alerted when groups Fire_RM_1 and Fire_RM_2 go into alarm, and can acknowledge alarms from both groups. Carol can modify tags Boiler_1_lev, Boiler_2_lev, and Boiler_3_lev because her User ID has been assigned the appropriate security privilege, and because her User ID and these three tags are members of the same process group.

Using Process Groups with System Alarms

By default, all OSx system alarms are placed in the system alarm group. The default configuration assigns this alarm group to all process groups and to all system users who have permission to view and acknowledge alarms.

> **WARNING**
>
> **An unnoticed system alarm could result in death or serious injury to personnel and/or damage to equipment.**
>
> **Do not disable membership for the system alarm group in any of the process groups. This group's membership reduces the possibility that a system alarm could go unnoticed.**
>
> **Ensure that all system users are assigned to the system alarm group to view and acknowledge these critical alarms.**

Using Process Groups with Other Alarms

Carefully consider all options before you assign or remove user IDs for a process group.

To help you monitor alarms with a process group, configure a special group and make it global by making all users of the system members of the process group. Assign important alarm groups and tags to this process group.

> **WARNING**
>
> **If you configure an alarm for a process group, and no currently logged-on user ID has membership to the process group, an alarm will not be displayed. This configuration could result in death or serious injury to personnel and/or damage to equipment.**
>
> **If you have any plant alarms that all users should be aware of, such as toxic fumes sensors, create a special and dedicated alarm group for these alarms and do not disable membership for this alarm group in any of the process groups.**
>
> **To reduce the possibility of one of these global-type alarms going unnoticed, assign all system users to this process group.**

Using the User/Action Request Link

You can link specific users and action requests so that an operator can respond to only those action requests that occur in that operator's particular area of responsibility. Refer to Table 2-3, which is a copy of the table shown in Section 2.1, and examine the sections labeled User and Action Request.

When you assign the request to a process group, the Request field in the Process Group Assignment dialog box displays only the first 7 characters of the action request name. For proper operation ensure that all tags associated with the action requests are assigned to the same process groups.

In Table 2-3, Carol can respond to the action requests. John can view, but not respond to, the action requests. Carol is linked to the action requests through the process groups as follows:

- Carol and the Add H_2O action request are members of process group Receiving Area 1.
- Carol and the Set Temperature action request are members of process groups Receiving Area 2 and Stock Area 1.

Table 2-3 Linking Users and Action Requests

User	Process Group					
	Prcss. Area 1	**Prcss. Area 2**	**Prcss. Area 3**	**Rcv. Area 1**	**Rcv. Area 2**	**Stock Area 1**
John	X	X	X			
Carol	X		X	X	X	X
Tag						
Boiler_1_lev	X	X				
Boiler_2_lev		X	X			
Boiler_3_lev				X	X	
Alarm Group						
Fire_RM_1	X	X		X		
Fire_RM_2				X	X	X
Action Request						
Add H_2O				X		
Set Temperature					X	X

2.3 Assigning Descriptions to Process Groups

Preparing to Configure Process Groups

You can use up to 32 process groups in your process. It is not necessary to use all 32 groups or to use them sequentially. You can assign a text description (up to 30 characters) to each process group that you configure. The characters ; \ " are not allowed in process group descriptions.

You assign the process group descriptions at any of three points during configuration:

- When you use APT, you can choose the **Set Process Group** option while you mark tags for translation. This procedure is described in the APT documentation.
- When you use a spreadsheet or ASCII text editor to configure tags, you can create process-group descriptions. This is described in Chapter 5.
- When you choose the **Process Group** option from the menu bar, you can enter descriptions in the Process Group Configuration display.

Entering Descriptions in the Process Group Configuration Display

To access the Process Group Configuration display, select **Data-> Process Group** from the menu bar (Figure 2-2). For each process group that your process requires, enter a description in the appropriate field to describe the function or characteristic of the group.

Descriptions entered in the Process Group Configuration display are overwritten by descriptions entered during tag configuration. You can choose from the following approaches for entering descriptions.

- Enter process group descriptions when you do tag configuration. These descriptions appear in the Process Group Configuration display after you import the tag data into an OSx station.
- Enter no process group descriptions when you do tag configuration. Enter the descriptions in the Process Group Configuration display after you import the tag data into an OSx station.

You can make changes to existing descriptions in the Process Group Configuration display at any time. Update the descriptions in your APT program or in your spreadsheet database at this time.

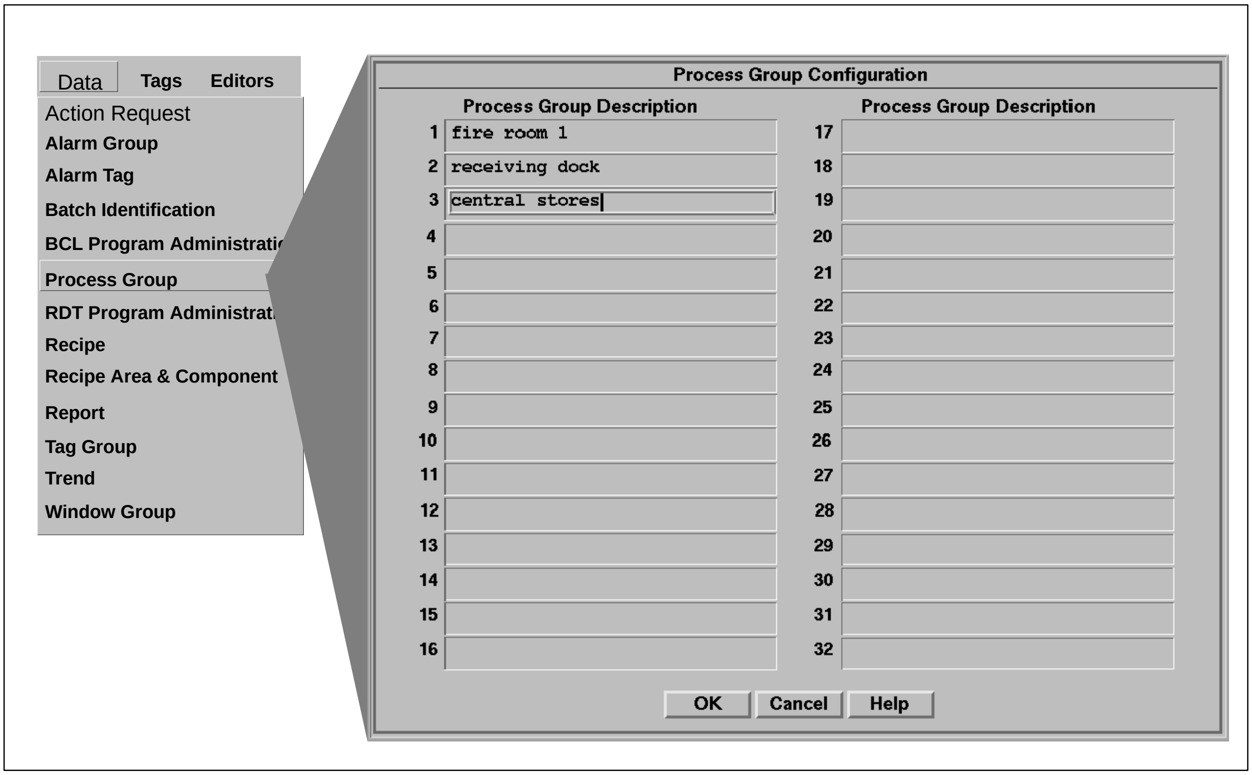


Figure 2-2 Process Group Configuration Display

Chapter 3
Defining Tags

3.1 Using Tags to Identify Values in the SIMATIC PCS 7 OSx System

Defining a Tag

A tag represents a block of data pertaining to an instrument or device associated with the plant process. A tag can be the collection of data that represents a timer, a motor, or a single I/O point. A tag can also be the block of all data that relates to a single PID loop. When the system references a tag, the system accesses all the information represented by the tag.

A tag is identified by a unique group of alpha-numeric characters. Figure 3-1 illustrates the tags used to represent an agitating mixer, control valve, pumps, level-indicator control, and temperature indicator.

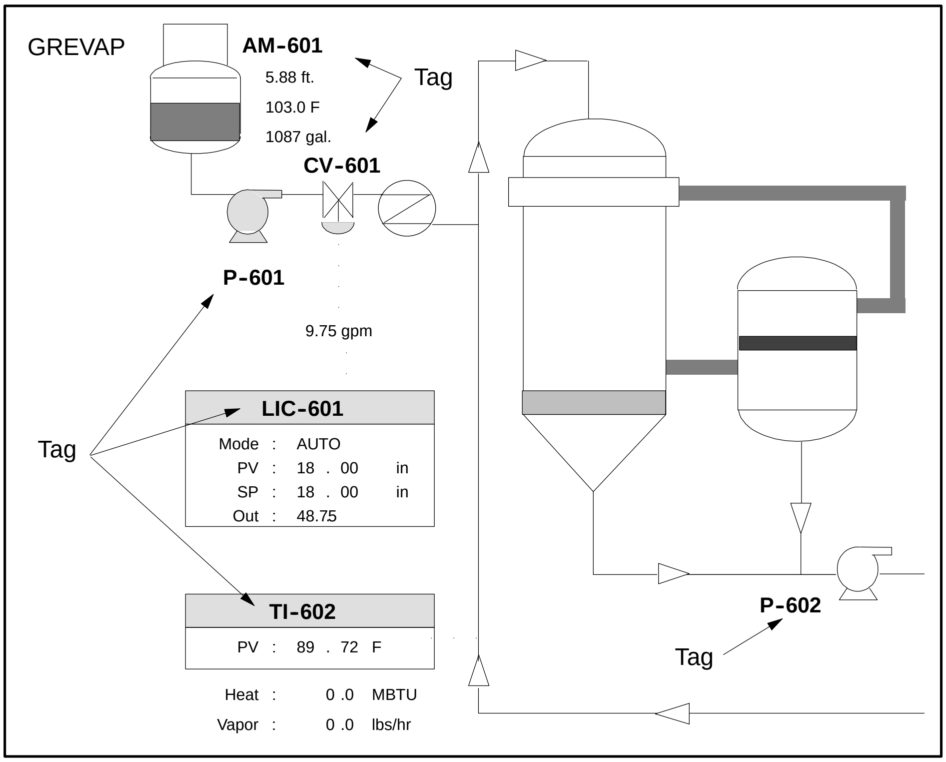


Figure 3-1 Sample Tags

Tag Names

A tag name can contain up to 12 characters. The space, the character combinations -> and <- and the characters \ , ; ” are invalid for tag names. The tag name must begin with an alphanumeric character (A–Z, 0–9) or the underscore (_) character. The tag name can be entered in upper or lower case (Figure 3-2). Do not use tag names that are the same as tag type names. For example, do not assign a VLV1 tag the name VLV1. Use a name like VLV1_3 instead to avoid confusion in the database.

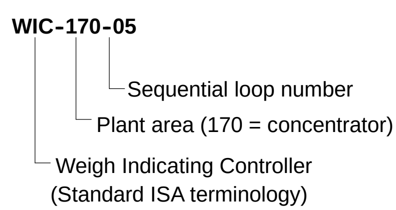


Figure 3-2 Sample Tag Name

Tag Types

The OSx tag types represent related data elements in the controller and determine the specific characteristics of tagged data. OSx tag types are listed in Table 3-1. These tag types are described in detail later in this chapter. The Area tag type is described in the *SIMATIC PCS 7 OSx Recipe Manual* and the Unit tag type is described in the *SIMATIC PCS 7 OSx Batch Programming Manual.*

Tag Attributes

Each tag type consists of a group of associated attributes. An attribute gives the tag its specific characteristics. The value of an attribute determines both its display features and alarm conditions. The attributes of the various tags are described in the sections that follow.

Table 3-1 OSx Tag Types

Tag Types	Tag Types	Tag Types	Tag Types
Analog Input	Digital Input	Loop	Text
Analog Output	Digital Input (10-Bit)	Motor	Timer
Area [1]	Digital Output	Motor (Reversible)	Unit [2]
Calculated Variable	Digital Output (10-Bit)	Motor (2-Speed)	Valve (Single-Feedback)
Counter	Integer Variable	System	Valve (Dual-Feedback)

1 The Area tag type is described in the *SIMATIC PCS 7 OSx Recipe Manual*.
2 The Unit tag type is described in the *SIMATIC PCS 7 OSx Batch Programming Manual*.

3.2 How Tag Elements Are Stored in the Database

Database Tables

The OSx database consists of tables that contain information about your system and process. This section describes two of the tables:

- The Tag_Xref table contains the information that identifies a tag, such as the user-assigned names and descriptions, the type of tag, or a cross-reference key into the tag's Process I/O table.
- The Process I/O table contains the actual data taken from the process. A unique Process I/O table exists for each type of tag.

Structure of the Tag_Xref Table

The Tag_Xref table consists of attributes (fields that identify the contents of a column of the table) and tuples (one or more attributes that make up a row of the table). Table 3-2 lists the attributes of the Tag_Xref table.

The attribute called "Tag" contains the user-assigned name of a tag. Tag acts as a cross-reference into the Process I/O table and allows you to access process data by using the user-assigned tag name. When you refer to a tag by name in your configuration, you access it through this attribute.

Another attribute, called "Pseudo_tag," provides an alternative for accessing process data. Like Tag, Pseudo_tag acts as a cross-reference into the Process I/O table. Typically, you use the attribute called Tag to access process I/O data because the name that you assign to a tag is easier to remember and has more meaning. The system automatically uses the Pseudo_tag attribute, however, since it allows a faster access into the Process I/O tables.

Each attribute has a domain that determines the range and type of value. Domains fall into three categories: set, scalar, or standard. Each set and scalar domain maps the 16-bit word to specified values that are listed with the information about each type of table. The Definition, Range, and Null values for the standard domain are listed in Table 3-3.

- A set domain consists of one or more values in brackets defined for an attribute. The attribute can assume any combination of values from the defined list; the empty set {} is acceptable. The following attribute has a set domain: **di_qual: Set of {Active, Alrm_UnAck, Man_Set, Data_Val}**
- A scalar domain consists of a list of values in parentheses. The attribute can assume the value of only one member of the defined list at any given time. The following attribute has a scalar domain: **loopmode: (manual, auto, cascade)**
- The standard domain is used to represent text and numeric values. The analog input target attribute has a domain of **float 32**.

Table 3-2   Tag_Xref Table Attributes

Attribute	Domain	Definition
Tag	CISTRING[12]	Tag name (a string, 12 characters long, not case-sensitive; all characters except the space and –> <– , ” \ ; are valid).
Pseudo_tag	SINT16	Signed integer identifying tag and used in process I/O tables instead of tag name.
Descriptor	STRING[30]	Tag description (a string 30 characters long; all characters except ” \ ; are valid).
Relation_id	SINT16	Numeric ID of process I/O table.
Status	Set of {Active, Inactive, Error}	Current status of I/O point (ON/OFF scan from network).
Parent_tag	SINT16	Identifies associated tag.
Tag_type	SCALAR of (AI, AO, DI, DO, DI10, DO10, VLV1, VLV2, RMTR, MTR1, CALC, IVAR, TMR, CTR, LOOP, SYS, AREA, MTR2, TEXT, BCH, UNIT)	Identifies tag type.

Table 3-3   Standard Domain Values

Value	Definition	Range	Null
int	Signed 32-bit integer	–2147483648 to 2147483647	–2147483648
long	Signed 32-bit integer	–2147483648 to 2147483647	–2147483648
SINT16	Signed 16-bit integer	–32768 to 32767	–32768
UINT16	Unsigned 16-bit integer	0 to 65535	65535
ushort	Unsigned 16-bit integer	0 to 65535	65535
SINT32	Signed 32-bit integer	–2147483648 to 2147483647	–2147483648
UINT32	Unsigned 32-bit integer	0 to 4294967295	429496729
FLOAT32	32-bit floating point	–1e+38 to 1e+38	–1e+38
BIT16	Set of 16 bits	0 to 0xFFFF	0
BIT32	Set of 32 bits	0 to 0xFFFFFFFF	0
SINT8	Signed 8-bit integer	–128 to 127	0
UINT8	Unsigned 8-bit integer	0 to 255	0
STRING[x]	Character string case sensitive	x characters long	
CISTRING[x]	Character string case insensitive	x characters long	

Structure of the Tag_Xref Table (continued)

Figure 3-3 illustrates the structure of the Tag_Xref table. Each column in this table represents an attribute; each row contains a record, or tuple, of information.

Column=Attribute

Row=Tuple or Record

tag	pseudo_tag	descriptor	relation_id	status	parent_tag	tag_type
T107	1	valve output	32	0x0001	7	1
TMR-04	3	test timer	36	0x0001	2	10
Loop	9	test loop tag	35	0	-32768	12

Figure 3-3 Example of the Tag_Xref Table

Structure of the Process I/O Tables

Each type of tag is represented by a Process I/O table. Table 3-4 lists the various Process I/O tables and the corresponding tag types. The sections that follow describe details about the contents of the Process I/O tables, such as attributes and bit locations, for each tag type.

The database tables always contain the properly scaled values. OSx automatically scales all integer values read from the controller before writing them to the database. If the controller has a special function for scaling, values can be scaled in the controller and written directly to the database. Similarly, floating point values sent to the controller from the database can be unscaled either in the OSx station or by a special function block in the controller.

Table 3-4 Types of Process I/O Tables

Process I/O Table	Tag Type
AI	Analog Input
AO	Analog Output
AREA [1]	Recipe Areas
CALC	Calculated Variable
CTR	Counter
DEVICE	Devices (motors, valves)
DI	Digital Input
DO	Digital Output
DI10	10-Bit Digital Input
DO10	10-Bit Digital Output
IVAR	Integer Variable
LOOP	Loop Controller
SYS	System
TEXT	Text
TMR	Timer
UNIT [2]	Unit

1 The Area tag type is described in the *SIMATIC PCS 7 OSx Recipe Manual*.
2 The Unit tag type is described in the *SIMATIC PCS 7 OSx Batch Programming Manual*.

Networked Tag Values

Typically, all tag data values reside in a particular memory location in a controller. Refer to Appendix D for a list of the supported memory locations. For data values that can reside in bit memory (that is, X/Y/C for Series 505 controllers, or F flags for SIMATIC S5 controllers), the S5 controllers provide read access for each bit, but the smallest write element is a byte. Therefore, all writeable tag values must be placed in data block memory for an S5 controller.

In the tag descriptions that begin in Section 3.3, any attributes described in terms of bits (for example, a single bit, two bits) are typically assigned to a Series 505 bit memory location in X/Y/C memory. However, these data values can also be assigned to word memory. In this case, the most significant bit(s) are used to represent the values. For S5 controllers, any tag value described in terms of bits that is capable of being written from OSx is assigned to a data block word and not to the F flag area.

In the OSx database, each tag has a status attribute that is defined as a networked data value. The status attribute is always 16 bits long, and the lower 3 bits are reserved. Note that these 3 bits are used only within the database, and that they are never written to the controller. In fact, in the controller, the status value is typically assigned to bit memory, and only the necessary number of bits is defined. For example, a DI tag status requires only a single bit to represent on/off, and thus, only one bit, not 16 bits, needs to be defined/reserved.

In the OSx database and in the Series 505 controllers, floating point values are stored in IEEE format. Since the S5 controllers store floating point values in the MC5 format, OSx automatically converts the values into the IEEE format whenever floating point values are read or written to the S5 controller. Depending on the magnitude and/or precision of the S5 floating point value, some bit significance may be sacrificed in the conversion.

3.3 AI — Analog Input Tag

An analog input represents a proportional input to an OSx station from a process sensor. An AI tag consists of range information, an input value (such as pressure or temperature), alarm information, and status. The AI tag has the following attributes that must be entered under the attribute heading in the **install.tag** file (Chapter 5).

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , ” ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; ” are illegal in this field.

Low Range/High Range Attributes (l_range, h_range) The low range and high range attributes are read/write values. The value for high range must be greater than the value for low range. Range values allow proportional scaling of bar graphs. The domain for these attributes is FLOAT32. Enter these values in engineering units.

Target Attribute (target) The target attribute is similar to the setpoint on a loop tag and is linked to deviation alarms. Target provides a visual reference on the tag detail display. This reference point allows you to see whether or not the process is operating at the target value. The domain is FLOAT32. Target must be between the values configured for high and low range.

Process Variable Attribute (pv) The process variable attribute is read-only and shows the current value of the variable being monitored. The domain is FLOAT32.

Absolute Alarm Attributes (hh_alm, h_alm, l_alm, ll_alm) The absolute alarm attributes, high high, high, low, and low low are defined for process variables. High high and low low alarms contain values that require critical actions. High and low alarms contain values that require remedial measures. The domain for these attributes is FLOAT32. Absolute alarms are read/write and must be set between the values for high and low range.

Deviation Alarm Attributes (h_dev, l_dev) The high (orange) and low (yellow) deviation alarm attributes define the limits of deviation from a target value that cause an alarm indication. Deviation alarms occur when there is a difference between the normal target value and the current operating value. You can use both absolute and deviation alarm attributes in graphics and for alarms. Alarm limit checking remains in the controller. The domain for these attributes is FLOAT32.

Alarm Deadband Attribute (al_deadband) OSx provides the alarm deadband attribute for alarm detection performed by user-written BCL programs. This attribute cannot be configured as a controller memory location. It is provided as a convenience for BCL programmers. The domain is FLOAT32.

Status Attribute (status) The status attribute, shown in Figure 3-4, consists of 16 bits that are read-only. The domain for this attribute consists of the bit locations shown in Figure 3-4. Code locations are shown in Table 3-5. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag. The Alrm_UnAck bit shows that the alarms have not been acknowledged for a point. The Man_Set bit indicates whether or not the tag has been set to accept manual trending of data. Either the controller program or a hardware device sets the remaining 13 bits.

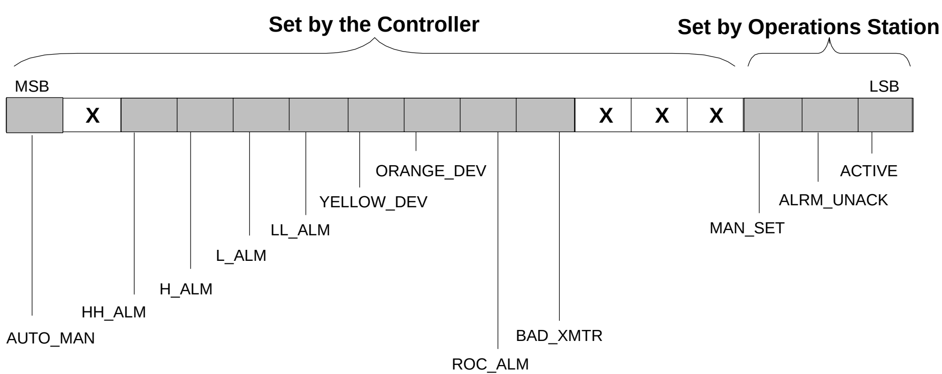


Figure 3-4 Bit Locations for AI Status Attribute

Table 3-5 Hex Code Locations for AI Status Attribute Bits

Bit Name	Definition	Location (Hex)
Auto_Man	Automatic/Manual Set	0x8000
HH_Alarm	High High Alarm	0x2000
H_Alarm	High Alarm	0x1000
L_Alarm	Low Alarm	0x0800
LL_Alarm	Low Low Alarm	0x0400
Yellow_Dev	Low (Yellow) Deviation	0x0200
Orange_Dev	High (Orange) Deviation	0x0100
ROC_Alm	Rate-of-Change Alarm	0x0080
Bad_Xmtr	Bad Transmitter	0x0040
Man_Set	Manual Trending of Data	0x0004
Alrm_UnAck	Alarm Not Acknowledged	0x0002
Active	Tag Is Active	0x0001

Rate-of-Change Alarm Attribute (roc_alm) The rate-of-change alarm attribute is a read/write attribute. This attribute indicates that the value of a process variable is changing more rapidly than the limits specified. The domain is FLOAT32.

Units Attribute (units) The units attribute indicates engineering units, such as tons or liters. The characters \ ; ” are illegal in this field. The domain is STRING8.

Change Attribute (change) The change attribute is the range in percent of span through which the process variable changes before the system updates the value in the database. The domain is FLOAT32.

NOTE: In normal use, analog alarm blocks in a Series 505 controller supply alarm attributes for analog input tags. However, custom programming in other controllers could supply this function.

3.4 AO — Analog Output Tag

An analog output represents a proportional output from an OSx station to a field actuator, such as a control valve or motor speed controller. An AO tag consists of the range information, status, mode, and output. For analog outputs you generally configure user-selectable areas on a graphic display. You can use these areas to call up command forms to perform such operations as setting motor speeds or gate positions. The AO tag has the following attributes that must be entered under the attribute heading in the **install.tag** file (Chapter 5).

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , ” ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; ” are illegal in this field.

Low Range/High Range Attributes (l_range, h_range) The low and high range attributes are values that are read/write. The value for high range must be greater than the value for low range. Range values allow proportional scaling of bar graphs. The domain for these attributes is FLOAT32. Enter these values as a percentage.

Status Attribute (status) The status attribute, shown in Figure 3-5, consists of 16 bits that are read only. The domain for this attribute consists of the bit locations shown in Figure 3-5. Hex code locations are shown in Table 3-6. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag. The Alrm_UnAck bit is not used; it is reserved for future use. The Man_Set bit indicates whether or not the tag has been set to accept manual trending of data.

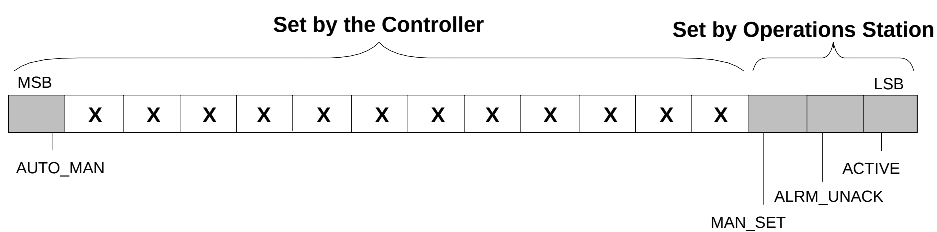


Figure 3-5 Bit Locations for AO Status Attribute

Table 3-6 Hex Code Locations for AO Status Attribute Bits

Bit Name	Definition	Location (Hex)
Auto_Man	Analog Output Mode	0x8000
Man_Set	Manually Trend Data	0x0004
Alrm_UnAck	Reserved	0x0002
Active	Tag Is Active	0x0001

Mode Attribute (mode) The mode attribute is a read/write attribute. The domain is Auto or Manual. The controller controls the process when the bit is set to 1 (auto mode). The operator controls the process when the bit is set to 0 (manual mode). Figure 3-6 shows the bit locations for the mode attribute. Either the controller or an OSx station can set these bits. Table 3-7 lists the values that determine how these bits are set.

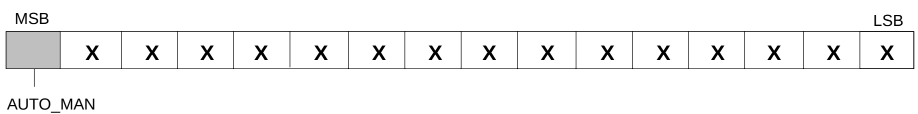


Figure 3-6 Bit Locations for AO Mode Attribute

Table 3-7 Command Values for the AO Mode Attribute

Command	Hexadecimal Value	Decimal Value
Auto	0x8000	–32768
Manual	0	0

Output Attribute (out) The output attribute is a read/write attribute. When reading from the controller, you can use the output value to animate an analog symbol. When writing to the controller, you can send the output floating-point value. The OSx station scales this attribute to an integer value if required. Scaling in the controller is not necessary. The output attribute value must be between the values set for high and low range. The domain is FLOAT32.

Units Attribute (units) The units attribute indicates engineering units, such as tons or liters. The characters \ ; ” are illegal in this field. The domain is STRING8.

Change Attribute (change) The change attribute is the range (in percent of span) through which output varies before the system updates the value in the database. The domain is FLOAT32.

3.5 CALC — Calculated Variable Tag

A calculated variable represents a general purpose floating point storage location. This floating point variable can be the result of a process measurement or of a calculation performed either in the controller or in the OSx station. A CALC tag consists of range information, status, and value. You use the CALC tag for applications when you want to upload floating point numbers and AI, AO, and LOOP tags are not appropriate. A CALC tag is similar to an analog input tag, but without alarming. The CALC tag has the following attributes that must be entered under the attribute heading in the **install.tag** file (Chapter 5).

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , ” ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; ” are illegal in this field.

Low Range/High Range Attributes (l_range, h_range) The low and high range attributes are read/write. The value for high range must be greater than the value for low range. Range values allow proportional scaling of bar graphs. The domain for these attributes is FLOAT32. Enter these values in engineering units.

Status Attribute (status) The status attribute, shown in Figure 3-7, consists of 16 read-only bits. The domain for this attribute consists of the bit locations shown in Figure 3-7. Hex code locations are shown in Table 3-8. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag. The Alrm_UnAck bit is not used and is reserved for future use. The Man_Set bit indicates whether or not the tag is set to accept manual trending of data.

Value Attribute (value) The value attribute is a read/write attribute. You can use it as a secure storage location for coefficients and values that are used in supervisory math. If value is not a networked point, the value you configure for the attribute is recorded only in the database. Value must be between the values set for high and low range. The domain is FLOAT32.

Units Attribute (units) The units attribute indicates engineering units, such as tons or liters. The characters \ ; ” are illegal in this field. The domain for this attribute is STRING8.

Change Attribute (change) The change attribute is the range (in percent of span) through which the calculated value varies before the system updates the value in the database. The domain is FLOAT32.

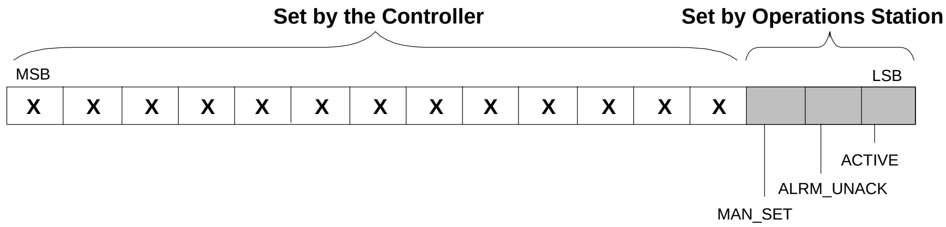


Figure 3-7 Bit Locations for CALC Status Attribute

Table 3-8 Hex Code Locations for CALC Status Attribute Bits

Bit Name	Definition	Location (Hex)
Man_Set	Manually Trend Data	0x0004
Alrm_UnAck	Reserved	0x0002
Active	Tag Is Active	0x0001

3.6 CTR — Counter Tag

A counter represents a counting function within a controller, such as an event counter or a totalizer. A CTR tag consists of the range information, preset, status, and current value. The CTR tag has the following attributes that must be entered under the attribute heading in the **install.tag** file (Chapter 5).

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , ” ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; ” are illegal in this field.

Low Range Attribute (l_range) The low range attribute is a non-networked point. Used with the preset attribute, the low range value allows proportional scaling of bar graphs. You write the value to a controller to set the lower limit of the count. The domain is SINT16. Enter this value in engineering units.

Preset Attribute (preset) The preset attribute is a read/write attribute. If a standard controller counter function is used, counters count up to the preset value. For scaling bar graphs, preset is used in the same manner as high range for other analog tag types. The domain is SINT16.

Status Attribute (status) The status attribute, shown in Figure 3-8, consists of 16 read-only bits. The domain for this attribute consists of the bit locations shown in Figure 3-8. Hex code locations are shown in Table 3-9. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag. The Alrm_UnAck bit is not used and is reserved for future use. The Man_Set bit indicates whether or not the tag is set to accept manual trending of data.

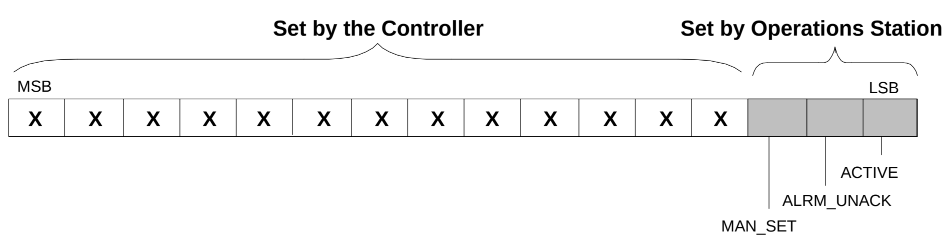


Figure 3-8 Bit Locations for CTR Status Attribute

Table 3-9 Hex Code Locations for CTR Status Attribute Bits

Bit Name	Definition	Location (Hex)
Man_Set	Manually Trend Data	0x0004
Alrm_UnAck	Reserved	0x0002
Active	Tag Is Active	0x0001

Current Attribute (current) The current attribute is a read-only integer. It is similar to an AI process variable in that either items counted or items remaining are reported to the OSx station from the controller. Current must be between the preset and low range values. The domain is SINT16.

Units Attribute (units) The units attribute indicates engineering units, such as tons or liters. The characters \ ; ” are illegal in this field. The domain is STRING8.

3.7 DI — Digital Input Tag

Select the digital input tag type to display or monitor on/off field devices if the operator does not need to send the value to the controller. The DI tag has the following attributes that must be entered under the attribute heading in the **install.tag** file (Chapter 5).

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , ” ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; ” are illegal in this field.

Status Attribute (status) The status attribute, shown in Figure 3-9, consists of 16 read-only bits. The domain for this attribute consists of the bit locations shown in Figure 3-9. Hex code locations are shown in Table 3-10. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag. The Alrm_UnAck bit shows that the alarms have not been acknowledged for a point. The Man_Set bit indicates whether or not the tag has been set to accept manual trending of data. The remaining 13 bits can be set either by logic or by hardware in the controller.

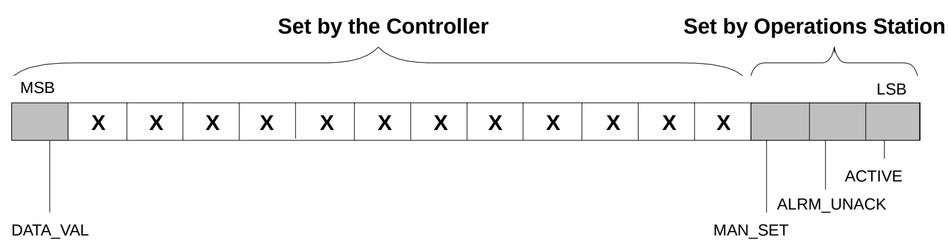


Figure 3-9 Bit Locations for DI Status Attribute

Data for a DI status attribute is stored in one bit. For Series 505 controllers, this is C or X/Y memory. For S5 controllers, this is a data block word location.

Table 3-10   Hex Code Locations for DI Status Attribute Bits

Bit Name	Definition	Location (Hex)
Data_Val	Data Value	0x8000
Man_Set	Manually Trend Data	0x0004
Alrm_UnAck	Alarm Not Acknowledged	0x0002

3.8 DI10 — Ten-Bit Digital Input Tag

If the operator does not need to send the value to the controller, select the 10-bit digital input tag type to display or monitor on/off field devices. The difference between this tag type and the DI tag type is that individual bits of the status attribute can be monitored and used for generic alarming purposes. The DI10 tag has the following attributes that must be entered under the attribute heading in the **install.tag** file (Chapter 5).

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , " ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; " are illegal in this field.

Status Attribute (status) The status attribute, shown in Figure 3-10, consists of 16 read-only bits. The domain consists of the bit locations shown in Figure 3-10. Hex code locations are shown in Table 3-11. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag, and the Alrm_UnAck bit shows that the alarms have not been acknowledged for a point. The Man_Set bit indicates whether or not the tag has been set to accept manual trending of data. The remaining 13 bits can be set either by logic or by hardware in the controller, but only the 10 most significant bits can be used for alarming.

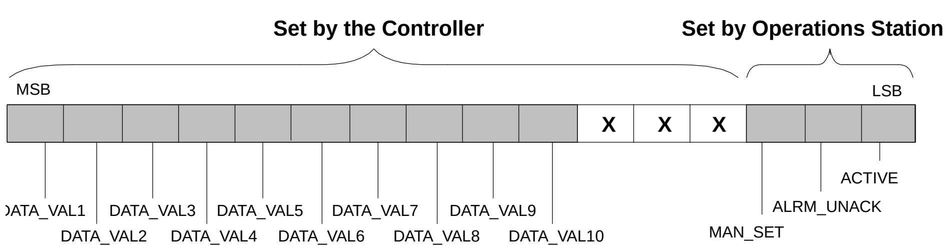


Figure 3-10 Bit Locations for DI10 Status Attribute

Data begins with the most significant bit. If you configure the DI10 tag to collect one bit of data (one location), that bit is the most significant bit. If you configure a DI10 to collect two bits of data and the controller address is C1, C1 is stored in the most significant bit and C2 in the next significant bit, and so on.

Data for a DI10 status attribute is stored in up to 10 sequential bits beginning with the most significant bit. The data must be in contiguous memory locations in the controller. For Series 505 controllers, this is C or X/Y memory. For S5 controllers, the value is assigned to a data block word location.

Table 3-11 Hex Code Locations for DI10 Status Attribute Bits

Bit Name	Definition	Location (Hex)
Data_Val1	Data Value 1	0x8000
Data_Val2	Data Value 2	0x4000
Data_Val3	Data Value 3	0x2000
Data_Val4	Data Value 4	0x1000
Data_Val5	Data Value 5	0x0800
Data_Val6	Data Value 6	0x0400
Data_Val7	Data Value 7	0x0200
Data_Val8	Data Value 8	0x0100
Data_Val9	Data Value 9	0x0080
Data_Val10	Data Value 10	0x0040
Man_Set	Manually Trend Data	0x0004
Alrm_UnAck	Alarm Not Acknowledged	0x0002
Active	Tag Is Active	0x0001

3.9 DO — Digital Output Tag

Select the digital output (DO) tag type when you need to send a command to the field device from an OSx station. The DO tag has the following attributes that must be entered under the attribute heading in the **install.tag** file (Chapter 5).

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , ” ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; ” are illegal in this field.

Status Attribute (status) The status attribute, shown in Figure 3-11, consists of 16 read-only bits. The domain for this attribute consists of the bit locations shown in Figure 3-11. Hex code locations are shown in Table 3-12. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag, and the Alrm_UnAck bit shows that the alarms have not been acknowledged for a point. The Man_Set bit indicates whether or not the tag has been set to accept manual trending of data. The remaining 13 bits can be set either by logic or by hardware in the controller.

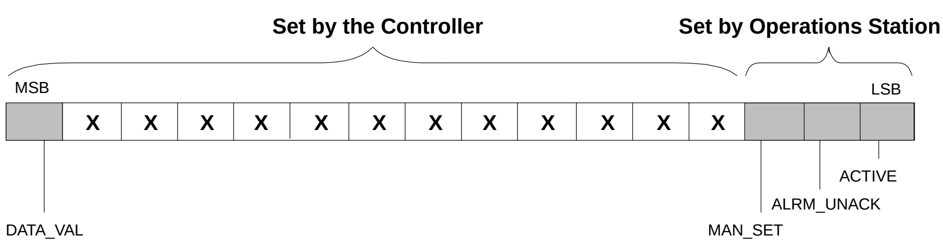


Figure 3-11 Bit Locations for DO Status Attribute

Table 3-12 Hex Code Locations for DO Status Attribute Bits

Bit Name	Definition	Location (Hex)
Data_Val	Data Value	0x8000
Man_Set	Manually Trend Data	0x0004
Alrm_UnAck	Alarm Not Acknowledged	0x0002
Active	Tag Is Active	0x0001

Data for a DO status attribute is stored in up to 13 sequential bits beginning with the most significant bit. The data must be in contiguous memory locations in the controller. For Series 505 controllers, this is C or X/Y memory. For S5 controllers, the value is assigned to a data block word location.

Figure 3-12 shows the DO status attribute stored in 10 sequential bits in C memory where the controller memory address is C104.

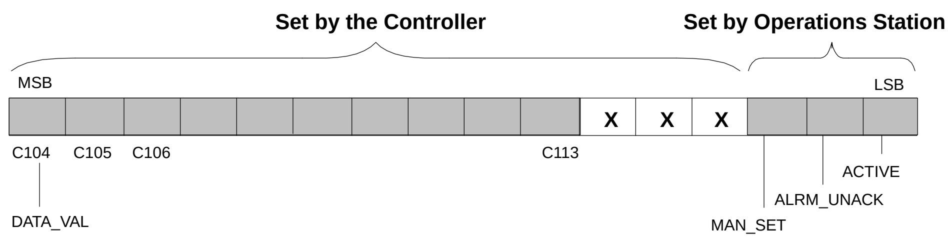


Figure 3-12 DO Status Stored in 10 Bits

Command Attribute (command) The DO command attribute is similar to a device setpoint; DO command is the desired value.

The DO command attribute is stored in sequential bits in the controller (Figure 3-13). For Series 505 controllers, this is 1–16 C or X/Y memory locations. For example, if you designate Memory Type: `C` and No. of Locations: 3, during tag configuration, and the controller memory address is C101, data is stored in C101, C102, and C103. The values shown in Table 3-13 determine how these bits are set.

For S5 controllers, the system does not write to tag values in bit memory areas. Instead, the tag value must be assigned to a data block memory location.

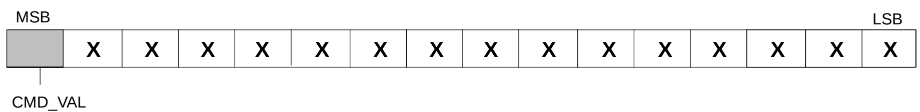


Figure 3-13 Bit Locations for DO Command Attribute

Table 3-13 Command Values for the DO Command Attribute

Command	Hexadecimal Value	Decimal Value
On	0x8000	–32768
Off	0	0

3.10 DO10 — Ten-Bit Digital Output Tag

Select the 10-bit digital output tag type when you need to send a command to the field device from an OSx station. The difference between this tag type and the DO tag type is that you can address, monitor, and command individual bits of the status attribute using the data_val*x* attributes. This feature is useful for alarming. The DO10 tag has the following attributes that must be entered under the attribute heading in the **install.tag** file (Chapter 5).

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , ” ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; ” are illegal in this field.

Status Attribute (status) The status attribute, shown in Figure 3-14, consists of 16 read-only bits. The domain for this attribute consists of the bit locations shown in Figure 3-14. Hex code locations are shown in Table 3-14. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag, and the Alrm_UnAck bit shows that the alarms have not been acknowledged for a point. The Man_Set bit indicates whether or not the tag has been set to accept manual trending of data. The remaining 13 bits are set by logic or hardware in the controller. However, only the 10 most significant bits can be used for alarming.

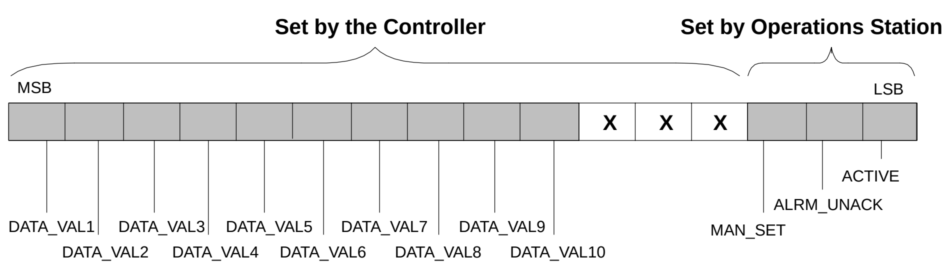


Figure 3-14 Bit Locations for DO10 Status Attribute

Table 3-14 Hex Code Locations for DO10 Status Attribute Bits

Bit Name	Definition	Location (Hex)
Data_Val1	Data Value 1	0x8000
Data_Val2	Data Value 2	0x4000
Data_Val3	Data Value 3	0x2000
Data_Val4	Data Value 4	0x1000
Data_Val5	Data Value 5	0x0800
Data_Val6	Data Value 6	0x0400
Data_Val7	Data Value 7	0x0200
Data-Val8	Data Value 8	0x0100
Data_Val9	Data Value 9	0x0080
Data_Val10	Data Value 10	0x0040
Man_Set	Manually Trend Data	0x0004
Alrm_UnAck	Alarm Not Acknowledged	0x0002
Active	Tag Is Active	0x0001

Data begins with the most significant bit. If you configure the DO10 tag to collect one bit of data (one location), that bit is the most significant bit. If you configure a DO10 to collect two bits of data and the controller address is C1, C1 is stored in the most significant bit and C2 in the next significant bit, and so on. In this case, Data_Val2 is assigned to C2.

Data for a DO10 status attribute is stored in up to 13 sequential bits beginning with the most significant bit. As noted previously, only 10 bits are accessible by the Data_Val*x* bit positions. To prevent confusion, use the 10 most significant bits for this purpose.

Data Value Attributes (data_val1 - data_val10) These attributes contain the commanded values for digital output points Data_Val1 through Data_Val10. The domain for each attribute is On and Off. Table 3-15 shows the values that determine how these bits are set.

Give the data_val*x* attributes the same addresses as the status attribute bits. For example, in Series 505 controllers, if the status of DO10 is assigned to C1 with 10 locations, data_val1 is assigned to C1. Then, data_val2 is assigned to C2, and so on. For an S5 controller you typically program the DO10 tag to have the data value assigned to data block words. Use ladder logic for setting the corresponding bits in the separate status attribute value.

Table 3-15 Command Values for the DO10 Data Val Attributes

Command	Hex Value	Decimal Value
	Data_Val1	**Data_Val1**
On	0x8000	–32768
Off	0x0000	0
.	.	.
.	.	.
.	.	.
	Data_Val10	**Data_Val10**
On	0x8000	–32768
Off	0x0000	0

3.11 IVAR — Integer Variable Tag

An integer variable represents a general purpose integer storage location within an OSx station, such as a current process step indicator or the number of units online. The IVAR tag consists of range information, status, and value. The IVAR tag has the following attributes that must be entered under the attribute heading in the **install.tag** file (Chapter 5).

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , ” ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; ” are illegal in this field.

Low Range/High Range Attributes (l_range, h_range) The low and high range attributes are non-networked points. The value for high range must be greater than the value for low range. Range values allow proportional scaling of bar graphs. Enter these values in engineering units. The domain for these attributes is SINT16.

Status Attribute (status) The status attribute, shown in Figure 3-15, consists of 16 read-only bits. The domain for this attribute consists of the bit locations shown in Figure 3-15. Hex code locations are shown in Table 3-16. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag. The Alrm_UnAck bit is not used and is reserved for future use. The Man_Set bit indicates whether or not the tag is set to accept manual trending of data.

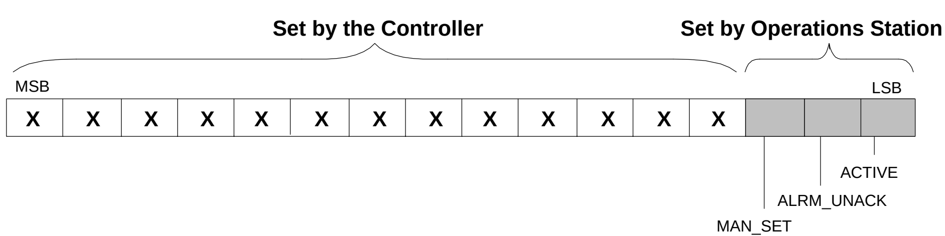


Figure 3-15 Bit Locations for IVAR Status Attribute

Table 3-16 Hex Code Locations for IVAR Status Attribute Bits

Bit Name	Definition	Location (Hex)
Man_Set	Manually Trend Data	0x0004
Alrm_UnAck	Reserved	0x0002
Active	Tag Is Active	0x0001

Value Attribute (value) The value attribute is a read/write attribute. This attribute must be between the values set for high and low range. The domain is SINT16.

Units Attribute (units) The units attribute indicates engineering units, such as tons or liters. The characters \ ; ” are illegal in this field. The domain is STRING8.

3.12 LOOP — Loop Tag

A loop represents a feedback control function within a controller, such as a PID or custom algorithm. A loop tag consists of range information, input value, setpoint, output, tuning constants, status, and alarm information. The LOOP tag has the following attributes that must be entered under the attribute heading in the **install.tag** file (Chapter 5).

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations –> and <– and the characters \ , " ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; " are illegal in this field.

Low Range/High Range Attributes (l_range, h_range) The low and high range attributes are read/write values. The value for high range must be greater than the value for low range. Range values allow proportional scaling of bar graphs. The domain for these attributes is FLOAT32. Enter these values in engineering units.

Process Variable Attribute (pv) The process variable is read only and shows the current value of the variable being monitored. The domain is FLOAT32.

Setpoint Attribute (sp) The setpoint is a read/write attribute. This attribute must be between the values set for high and low range. The domain is FLOAT32.

Output Attribute (out) Output is a read/write attribute that must be between 0% and 100%. The domain is FLOAT32.

Gain, **Rate**, and **Reset Attributes (gain, rate, reset)** These attributes are read/write attributes that define the loop-tuning constants. The domain for these attributes is FLOAT32.

Status Attribute (status) The status attribute, shown in Figure 3-16, consists of 16 read-only bits. The domain for this attribute consists of the bit locations shown in Figure 3-16. Hex code locations are shown in Table 3-17. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag, and the Alrm_UnAck bit shows that the alarms have not been acknowledged for a point. The Man_Set bit indicates whether or not the tag has been set to accept manual trending of data. The remaining 13 bits can be set either by logic or by hardware in the controller.

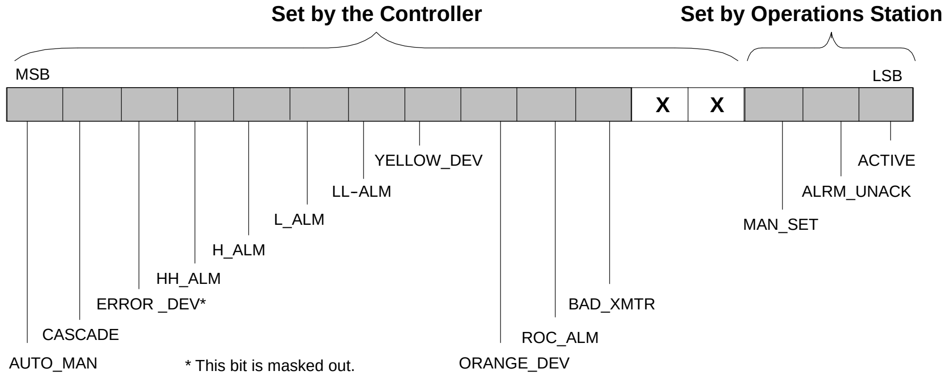


Figure 3-16 Bit Locations for LOOP Status Attribute

Table 3-17 Hex Code Locations for LOOP Status Attribute Bits

Bit Name	Definition	Location (Hex)
Auto_Man	Auto/Manual	0x8000
Cascade	Loop Cascade; Remote Setpoint	0x4000
Error_Dev	Error Deviation (Bit is masked out.)	0x2000
HH_Alarm	High High Alarm	0x1000
H_Alarm	High Alarm	0x0800
L_Alarm	Low Alarm	0x0400
LL_Alarm	Low Low Alarm	0x0200
Yellow_Dev	Low (Yellow) Deviation	0x0100
Orange_Dev	High (Orange) Deviation	0x0080
ROC_Alm	Rate-of-Change	0x0040
Bad_Xmtr	Bad Transmitter	0x0020
Man_Set	Manually Trend Data	0x0004
Alrm_UnAck	Alarm Not Acknowledged	0x0002
Active	Tag Is Active	0x0001

Mode Attribute (mode) The mode attribute is a read/write attribute that allows you to change the loop mode. Mode is also used to update values on OSx command forms and to show the operator the current loop mode. Command forms are configured with user-selectable areas such as **Auto**, **Manual**, or **Cascade**. Bit locations for the mode attribute are shown in Figure 3-17. The domain is Auto, Manual, or Cascade. The values shown in Table 3-18 determine how these bits are set.

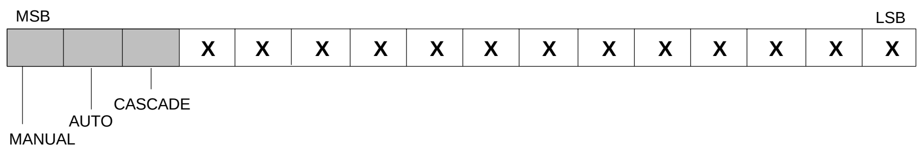


Figure 3-17 Bit Locations for LOOP Mode Attribute

Table 3-18 Command Values for the LOOP Mode Attribute

Command	Hexadecimal Value	Decimal Value
Manual	0x8000	–32768
Auto	0x4000	16384
Cascade	0x2000	8192

Absolute Alarm Attributes (hh_alm, h_alm, l_alm, ll_alm) The absolute alarm attributes, high high, high, low, and low low, are defined for process variables. High high and low low alarms contain values that require critical actions. High and low alarms contain values that require remedial measures. The domain for these attributes is FLOAT32. Absolute alarms are read/write and must be between the values set for high and low range.

Deviation Alarm Attributes (h_dev, l_dev) The high (orange) and low (yellow) deviation alarm attributes define the limits of deviation from a setpoint that cause an alarm indication. Deviation alarms occur when there is a difference between the setpoint and the current operating value. These alarms are read/write attributes. You can use both absolute and deviation alarm attributes in graphics and for alarms. Alarm limit checking remains in the controller. The domain for these variables is FLOAT32.

Alarm Deadband Attribute (al_deadband) OSx provides the alarm deadband attribute for alarm detection performed by user-written BCL programs. These attributes cannot be configured as controller memory locations. They are provided as a convenience for BCL programmers. The domain is FLOAT32.

Rate-of-Change Alarm Attribute (roc_alm) The rate-of-change alarm is a read/write attribute that indicates when the value of a process variable changes more rapidly than the limits specified. The domain is FLOAT32.

Units Attribute (units) The units attribute indicates engineering units, such as tons or liters. The characters \ ; ” are illegal in this field. The domain is STRING8.

Change Attribute (change) The change attribute is the range in percent of span through which the process variable changes before the system updates the value in the database. The domain is FLOAT32.

3.13 TEXT — Text Tag

Select the text tag type when it is necessary to read/write text to or from the controller. The operator can use this tag type to enter a textual comment. The controller can use it to send up to 90 characters partitioned into three 30-character lines to the operator.

NOTE: The text tag attributes (text_1, text_2, text_3) cannot begin with spaces. If you begin input into these attributes by entering spaces, the white space you leave in front of the text disappears when you complete the input. On the other hand, all unused characters at the end of the text attribute appear as spaces at the end of the line. These considerations can be important when you place lines of text side by side on a graphic.

The text tag has the following attributes that require an entry under the attribute heading in the **install.tag** file (Chapter 5).

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , ” ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; ” are illegal in this field.

Status Attribute (status) The text status attribute, shown in Figure 3-18, consists of 16 bits that are read only. The domain for this attribute consists of the bit locations shown in Figure 3-18. Hex code locations are shown in Table 3-19. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag. The Alrm_UnAck bit is not used and is reserved for future use. The Man_Set bit indicates whether or not the tag is set to accept manual trending of data.

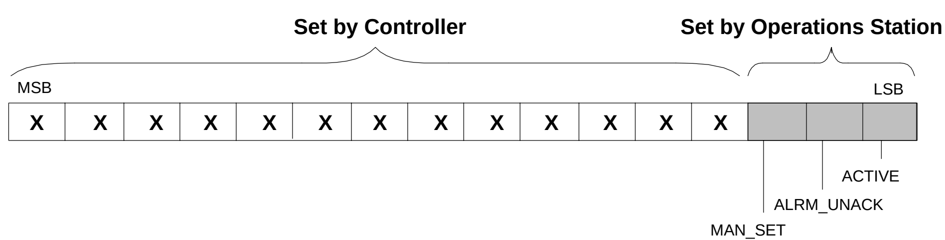


Figure 3-18 Bit Locations for TEXT Status Attribute

String Attributes (text_1, text_2, text_3) These attributes are read/write string attributes. The domain is STRING30 and up to 3 strings can be stored. Therefore, a textual message can be up to 90 characters long. For the Series 505 controllers, text attributes must be configured in V-memory. The default number of memory locations for each text_*x* attribute is one location for every two characters stored. For example, 30 characters take up 15 V-memory locations. For the S5 controllers, configure text attributes in data block word locations.

Units Attribute (units) The units attribute indicates engineering units, such as tons or liters. The characters \ ; ” are illegal in this field. The domain is STRING8.

Table 3-19 Hex Code Locations for TEXT Status Attribute Bits

Bit Name	Definition	Location (Hex)
Man_Set	Manually Trend Data	0x0004
Alrm_UnAck	Reserved	0x0002
Active	Tag Is Active	0x0001

3.14 TMR — Timer Tag

A timer represents a timing function, such as a time-to-complete timer. A TMR tag consists of the range information, status, preset, and current value. The TMR tag has the following attributes that require an entry under the attribute heading in the **install.tag** file (Chapter 5).

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , ” ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; ” are illegal in this field.

Low Range/High Range Attributes (l_range, h_range) The low and high range attributes are read/write values. The value for high range must be greater than the value for low range. Range values allow proportional scaling of bar graphs. You write range values to a controller to control the timer/counter preset (TCP). Enter these values in engineering units. The domain for these attributes is FLOAT32.

Status Attribute (status) The TMR status attribute, shown in Figure 3-19, consists of 16 bits that are read only. The domain for this attribute consists of the bit locations shown in Figure 3-19. Hex code locations are shown in Table 3-20. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag. The Alrm_UnAck bit is not used and is reserved for future use. The Man_Set bit indicates whether or not the tag is set to accept manual trending of data.

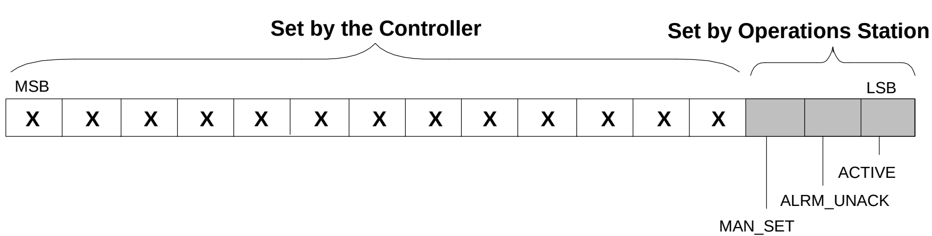


Figure 3-19 Bit Locations for TMR Status Attribute

Table 3-20 Hex Code Locations for TMR Status Attribute Bits

Bit Name	Definition	Location (Hex)
Man_Set	Manually Trend Data	0x0004
Alrm_UnAck	Reserved	0x0002
Active	Tag Is Active	0x0001

Preset Attribute (preset) The preset attribute is a read/write attribute that is similar to a setpoint. If a standard controller timer function is used, the preset is a floating point value that represents tenths of a second. Timers count down from the preset value to zero. The preset value must be between the high and low range values. The domain is FLOAT32.

Current Attribute (current) The current attribute is a read-only attribute that is converted to a scaled floating point value in the OSx station. This attribute is similar to an AI process variable in that it contains the current value of the timer. Current must be between the high and low range values. The domain is FLOAT32.

Units Attribute (units) The units attribute indicates engineering units, such as tons or liters. The characters \ ; ” are illegal in this field. The domain is STRING8.

Change Attribute (change) The change attribute is the range in percent of span through which the current timer value varies before the system updates the value in the database. The domain is FLOAT32.

3.15 Device Tags

Introduction

A device represents a field device, such as a single-feedback valve, dual-feedback valve, single-speed motor, reversible motor, or two-speed motor.

A device tag consists of a commanded value, a transition timeout value, override ability, and status. All these elements are regarded as parts of a single entity. The application program in the controller supports the proper operation of devices.

The device tags have the following attributes that require an entry under the attribute heading in the **install.tag** file (Chapter 5). The domain for each attribute is defined for each device type in the pages that follow.

- The attribute called Tag contains the tag name.
- The Descriptor attribute contains the description of the tag.
- The Status attribute contains information about the current state of the device.
- Use the Mode command attribute to select auto or manual control for the device.
- The Setpoint attribute indicates the condition of a commanded device, for example, open or closed.
- Use the Override attribute to ignore a faulty signal from a travel limit switch.
- The Timeout attributes are integer attributes that can either be uploaded to an OSx station or downloaded to a controller. A timeout is a preset time that indicates an excessive delay in receiving feedback from a travel limit switch during the transition of a device. The domain for these attributes is SINT16.

  Timeout_1 This attribute contains the command failure timeout for the transition from device state true (such as open valve, running motor) to device state false (closed valve, stopped motor).

  Timeout_2 This attribute contains the command failure timeout for the transition from device state false (such as closed valve or stopped motor) to device state true (open valve, running motor).

Single-Feedback Valves — VLV1

The properties specific to the attributes of the Single Feedback Valves are described below.

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , ” ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; ” are illegal in this field.

Status Attribute (status) The status attribute, shown in Figure 3-20, consists of 16 read-only bits. The domain for this attribute consists of the bit locations shown in Figure 3-20.

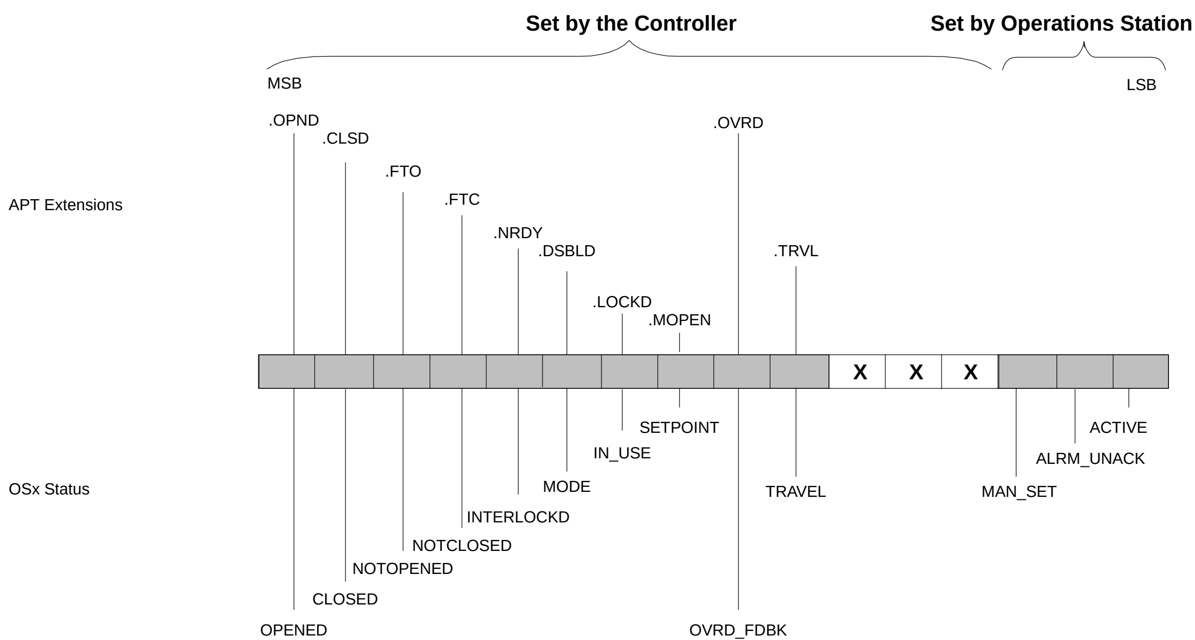


Figure 3-20 Bit Locations for VLV1 Status Attribute

Single-Feedback Valves — VLV1 (continued)

Hex code locations are shown in Table 3-21. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag and the Alrm_UnAck bit shows that the alarms have not been acknowledged for a point. The Man_Set bit indicates whether or not the tag has been set to accept manual trending of data. The remaining 13 bits can be set either by logic or by hardware in the controller.

Table 3-21 Hex Code Locations for VLV1 Status Attribute Bits

Bit Name	Definition	Location (Hex)
Opened	Current device state	0x8000
Closed	Current device state	0x4000
NotOpened	Device state alarm flag	0x2000
NotClosed	Device state alarm flag	0x1000
Interlockd	Flag indicating other control source	0x0800
Mode	Control mode: P/C or operator	0x0400
In_Use	Locks out operators; under controller control	0x0200
Setpoint	Current setpoint	0x0100
Ovrd_Fdbk	Feedback override mode flag	0x0080
Travel	Current device state	0x0040
Man_Set	Manually trend data	0x0004
Alrm_UnAck	Alarm not acknowledged	0x0002
Active	Tag is active	0x0001

Possible device conditions for VLV1 are shown in Table 3-22. Device conditions are valid only if the Active flag is set; the controller must be programmed to support device operation.

Table 3-22 Device Conditions Derived from VLV1 Status Attribute

Bit Value			Description
Opened	**Closed**		**Current Device State**
1	0		Open
0	1		Closed
1	1		Failed
Not Opened	**Not Closed**		**Current Device State**
1	X		Not open alarm
X	1		Not closed alarm
Ovrd Fdbk			**Feedback Override Mode Flag**
0			Do not override feedback
1			Override feedback
Travel	**Setpoint**		**Current Device State**
1	X		Travel
In Use	**Mode**	**Interlockd**	**Current Device State**
0	0	0	Hand
1	0	X	Sequencer control
X	1	X	Operator control
X	X	1	Interlocked

X means value can be either 1 or 0.
Feedback = travel limit switch indicating valve open or valve closed.
Device conditions are valid only if the Active flag is set; the controller must be programmed to support device operation.
If you are not using APT, you must program the controller to change the In_Use bit.

Single-Feedback Valves — VLV1 (continued)

Timeout Attributes (timeout_1, timeout_2) These are integer attributes that can either be uploaded to an OSx station or downloaded to a controller. The domain for the attributes is SINT16.

- Timeout_1 This attribute contains the command failure timeout for the Device True state.
- Timeout_2 This attribute contains the command failure timeout for the Device False state.

Setpoint Attribute (setpoint) This is a single-bit read/write attribute that indicates the condition (open or closed) of the device. The domain is Close and Open. The values in Table 3-23 determine how this bit is set.

Mode Command Attribute (mode_cmd) This is a single-bit read/write attribute that you use to select auto or manual control. Auto mode puts the process under controller control, and manual mode puts it under operator control. The domain is Seq (controller) and Oper (operator). The values shown in Table 3-23 determine how to set this bit. OSx does not support the 0x4000 bit of the mode_cmd attribute, while APT does. Therefore, any changes to this bit are not recorded by the autolog feature.

Override Attribute (override) This is a single-bit read/write attribute that you use to ignore a faulty signal from a travel limit switch. The domain is Close and Open. The values in Table 3-23 determine how this bit is set.

Table 3-23 Command Values for VLV1 Attributes

Attribute	Bit Name	Bit Location	Command	Hex Value	Decimal Value
Setpoint	Setpoint	0x8000	Close Open	0 0x8000	0 –32768
Mode_cmd	Mode	0x8000	Sequencer control Operator control	0 0x8000	0 –32768
Override	Ovrd_Fdbk	0x8000	Ovrd fdbk disabled Ovrd fdbk enabled	0 0x8000	0 –32768

Dual Feedback Valves — VLV2

The properties specific to the attributes of the Dual Feedback Valves are described below.

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , " ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; " are illegal in this field.

Status Attribute (status) The status attribute, shown in Figure 3-21, consists of 16 read-only bits. The domain for this attribute consists of the bit locations shown in Figure 3-21.

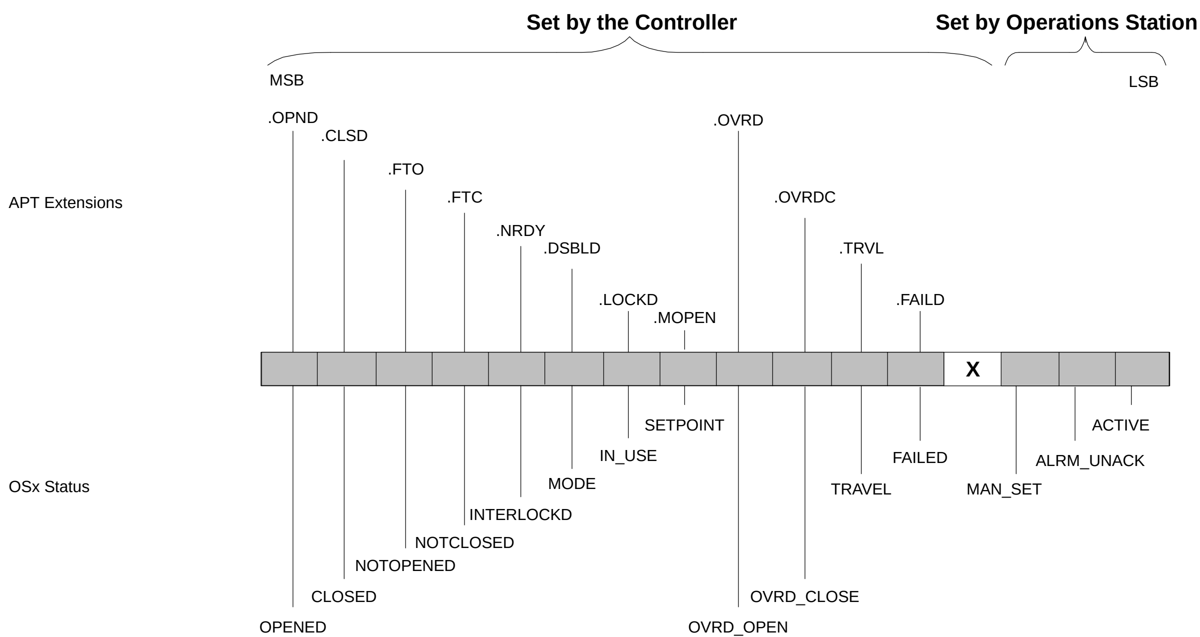


Figure 3-21 Bit Locations for VLV2 Status Attribute

Dual Feedback Valves — VLV2 (continued)

Hex code locations for status attribute bits are shown in Table 3-24. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag, and the Alrm_UnAck bit shows that the alarms have not been acknowledged for a point. The Man_Set bit indicates whether or not the tag has been set to accept manual trending of data. The remaining 13 bits can be set either by logic or by hardware in the controller.

Table 3-24 Hex Code Locations for VLV2 Status Attribute Bits

Bit Name	Definition	Location (Hex)
Opened	Current device state	0x8000
Closed	Current device state	0x4000
NotOpened	Device state alarm flag	0x2000
NotClosed	Device state alarm flag	0x1000
Interlockd	Flag indicating other control source	0x0800
Mode	Sequencer or Operator	0x0400
In_Use	Locks out all operators; under controller control	0x0200
Setpoint	Current setpoint	0x0100
Ovrd_Open	Feedback override mode flag	0x0080
Ovrd_Close	Feedback override mode flag	0x0040
Travel	Current device state	0x0020
Failed	Current device state	0x0010
Man_Set	Manually trend data	0x0004
Alrm_UnAck	Alarm not acknowledged	0x0002
Active	Tag is active	0x0001

Possible device conditions for VLV2 are shown in Table 3-25. Device conditions are valid only if the Active flag is set; the controller must be programmed to support device operation.

Table 3-25 Device Conditions Derived from VLV2 Status Attribute

Bit Value			Description
Opened	**Closed**		**Current Device State**
1	0		Open
0	1		Closed
1	1		Failed
Not Opened	**Not Closed**		**Current Device State**
1	X		Not open alarm
X	1		Not closed alarm
Ovrd Open	**Ovrd Close**		**Feedback Override Mode Flag**
0	1		Override close
1	0		Override open
1	1		Override both
Travel	**Setpoint**		**Current Device State**
1	X		Travel
In Use	**Mode**	**Interlockd**	**Current Device State**
0	0	0	Hand
1	0	X	Sequencer control
X	1	X	Operator control
X	X	1	Interlocked
Failed			**Current Device State**
0			Not failed
1			Failed

X means value can be either 1 or 0.
Feedback = travel limit switch indicating valve open or valve closed.
Device conditions are valid only if the Active flag is set; the controller must be programmed to support device operation.
If you are not using APT, you must program the controller to change the In_Use bit.

Dual Feedback Valves — VLV2 (continued)

Timeout Attributes (timeout_1, timeout_2) These are integer attributes that can either be uploaded to an OSx station or downloaded to a controller. The domain for the attributes is SINT16.

- Timeout_1 This attribute contains the command failure timeout for the Device True state.
- Timeout_2 This attribute contains the command failure timeout for the Device False state.

Setpoint Attribute (setpoint) This is a single-bit read/write attribute that indicates the condition (open or closed) of a commanded device. The domain is Close and Open. The values shown in Table 3-26 determine how this bit is set.

Mode Command Attribute (mode_cmd) This is a single-bit read/write attribute that you use to select auto or manual control. Auto mode puts the process under controller control, and manual mode puts it under operator control. The domain is Seq (controller) and Oper (operator). The values shown in Table 3-26 determine how this bit is set. OSx does not support the 0x4000 bit of the mode_cmd attribute, while APT does. Therefore, any changes to this bit are not recorded by the autolog feature.

Override Attribute (override) This is a two-bit read/write attribute that you use to ignore a faulty signal from a travel limit switch. The domain is None, Close, Open, and Open/Close. The values shown in Table 3-26 determine how these bits are set.

The device override attribute must be stored in 2 sequential bit locations beginning with the most significant bit. The device mode command and setpoint attributes must be stored in one bit location. For Series 505 controllers, this is C or X/Y memory. For S5 controllers, this is a data block word location.

Table 3-26 Command Values for VLV2 Attributes

Attribute	Bit Name	Bit Location	Command	Hex Value	Decimal Value
Setpoint	Setpoint	0x8000	Close Open	0 0x8000	0 –32768
Mode_cmd	Mode	0x8000	Sequencer control Operator control	0 0x8000	0 –32768
Override	Ovrd_Open Ovrd_Close	0x8000 0x4000	Do not ovrd fdbk Ovrd close fdbk Ovrd open fdbk Ovrd open and close fdbk	0 0x4000 0x8000 0xC000	0 16384 –32768 –16384

Motors — MTR1

The properties specific to the attributes of the motors are described below.

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , ” ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; ” are illegal in this field.

Status Attribute (status) The status attribute consists of 16 read-only bits. The domain for this attribute consists of the bit locations shown in Figure 3-22.

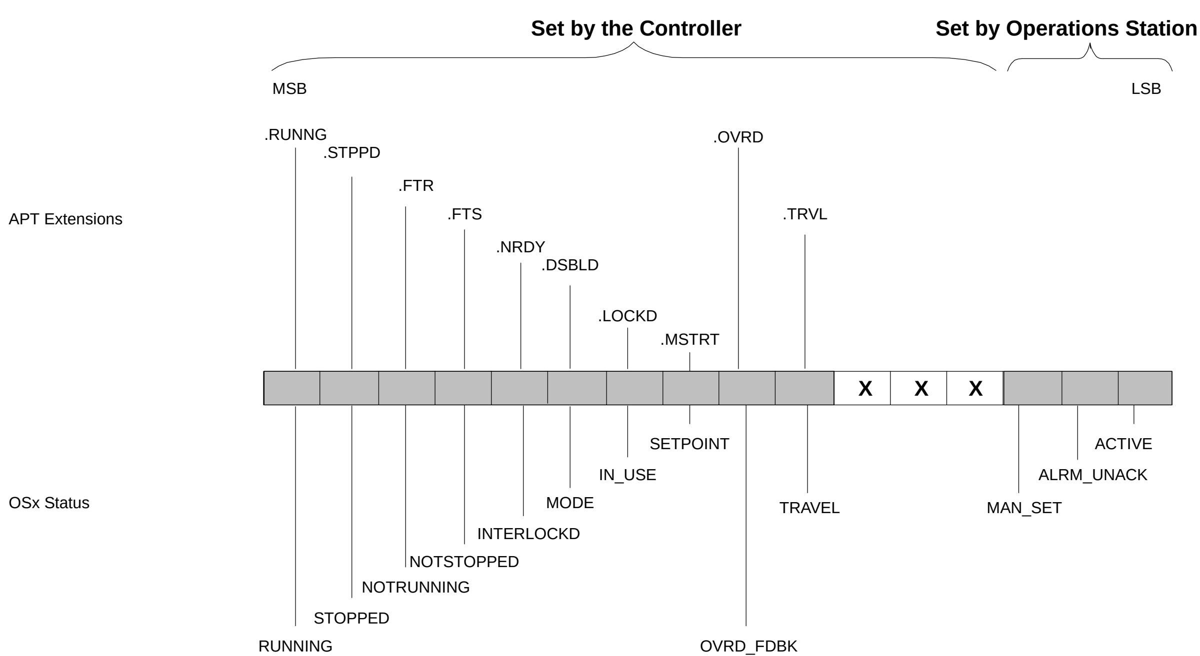


Figure 3-22 Bit Locations for MTR1 Status Attribute

Hex code locations are shown in Table 3-27. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag, and the Alrm_UnAck shows that the alarms have not been acknowledged for a point. The Man_Set bit indicates whether or not the tag has been set to accept manual trending of data. The remaining 13 bits can be set either by logic or by hardware in the controller.

Table 3-27 Hex Code Locations for MTR1 Status Attribute Bits

Bit Name	Definition	Location (Hex)
Running	Current device state	0x8000
Stopped	Current device state	0x4000
NotRunning	Device state alarm flag	0x2000
NotStopped	Device state alarm flag	0x1000
Interlockd	Flag indicating other control source	0x0800
Mode	Sequencer or Operator	0x0400
In_Use	Locks out operators; under controller control	0x0200
Setpoint	Current setpoint	0x0100
Ovrd_Fdbk	Feedback override mode flag	0x0080
Travel	Current device state	0x0040
Man_Set	Manually trend data	0x0004
Alrm_UnAck	Alarm not acknowledged	0x0002
Active	Tag is active	0x0001

Motors — MTR1 (continued)

Possible device conditions for MTR1 are shown in Table 3-28. Device conditions are valid only if the Active flag is set; the controller must be programmed to support device operation.

Table 3-28 Device Conditions Derived from MTR1 Status Attribute

Bit Value			Description
Running	**Stopped**		**Current Device State**
1	0		Running
0	1		Stopped
1	1		Failed
Not Running	**Not Stopped**		**Current Device State**
1	X		Not running alarm
X	1		Not stopped alarm
Ovrd Fdbk			**Feedback Override Mode Flag**
0			Do not override feedback
1			Override feedback
Travel	**Setpoint**		**Current Device State**
1	X		Travel
In Use	**Mode**	**Interlockd**	**Current Device State**
0	0	0	Hand
1	0	X	Sequencer control
X	1	X	Operator control
X	X	1	Interlocked
X means value can be either 1 or 0. Feedback = travel limit switch indicating motor running or motor stopped. Device conditions are valid only if the Active flag is set; the controller must be programmed to support device operation. If you are not using APT, you must program the controller to change the In_Use bit.			

Timeout Attributes (timeout_1, timeout_2) These are integer attributes that can either be uploaded to an OSx station or downloaded to a controller. The domain for the attributes is SINT16.

- Timeout_1 This attribute contains the command failure timeout for the device true state.
- Timeout_2 This attribute contains the command failure timeout for the device false state.

Setpoint Attribute (setpoint) This is a read/write attribute that indicates the condition (stop or run) of the device. The domain is Stop and Run. The values shown in Table 3-29 determine how this bit is set.

Mode Command Attribute (mode_cmd) This is a read/write attribute that you use to select auto or manual control. Auto mode puts the process under controller control, and manual mode puts it under operator control. The domain is Seq (controller) and Oper (operator). The values shown in Table 3-29 determine how this bit is set. OSx does not support the 0x4000 bit of the mode_cmd attribute, while APT does. Therefore, any changes to this bit are not recorded by the autolog feature.

Override Attribute (override) This is a read/write attribute that you use to ignore a faulty signal from a motor run indicator. The domain is Off and On. The values shown in Table 3-29 determine how this bit is set.

The device override, mode command and setpoint attributes must be stored in one bit location. For Series 505 controllers, this is C or X/Y memory. For S5 controllers, this is a data block word location.

Table 3-29 Command Values for MTR1 Attributes

Attribute	Bit Name	Bit Location	Command	Hex Value	Decimal Value
Setpoint	Setpoint	0x8000	Stop Run	0 0x8000	0 –32768
Mode_cmd	Mode	0x8000	Sequencer control Operator control	0 0x8000	0 –32768
Override	Ovrd_fdbk	0x8000	Do not ovrd fdbk Ovrd fdbk	0 0x8000	0 –32768

Reversible Motors — RMTR

The properties specific to the attributes of the Reversible Motors are described below.

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , ” ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; ” are illegal in this field.

Status Attribute (status) The status attribute, shown in Figure 3-23, consists of 16 read-only bits. The domain for this attribute consists of the bit locations shown in Figure 3-23.

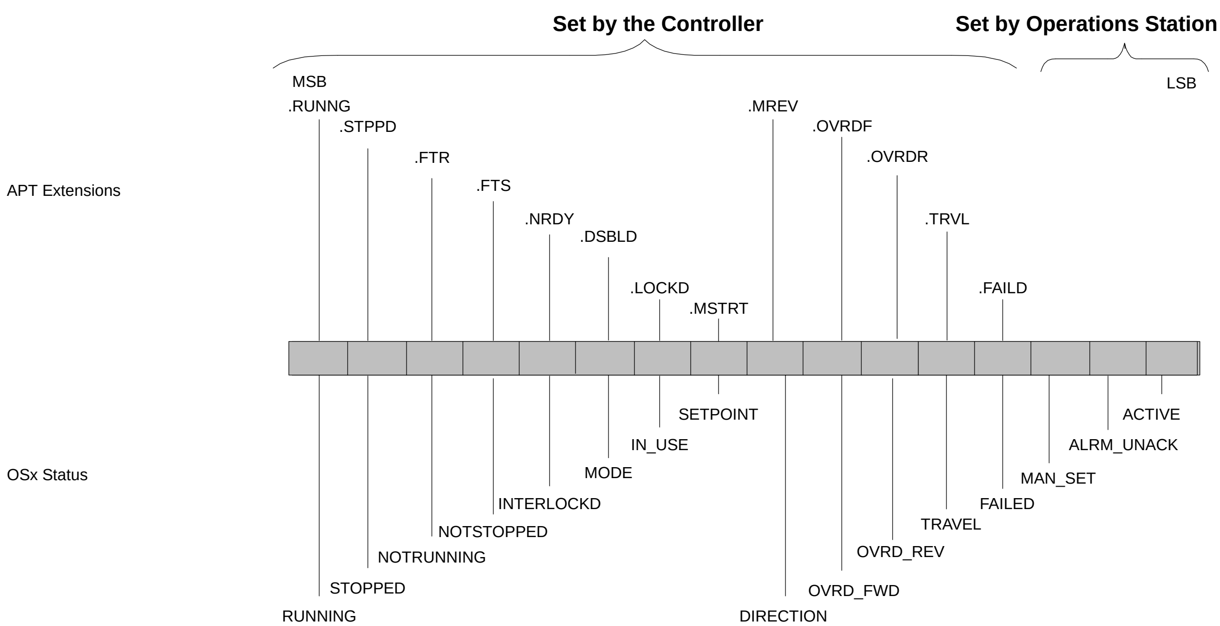


Figure 3-23 Bit Locations for RMTR Status Attribute

Hex code locations are shown in Table 3-30. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag, and the Alrm_UnAck bit shows that the alarms have not been acknowledged for a point. The Man_Set bit indicates whether or not the tag has been set to accept manual trending of data. The remaining 13 bits can be set either by logic or by hardware in the controller.

Table 3-30 Hex Code Locations for RMTR Status Attribute Bits

Bit Name	Definition	Location (Hex)
Running	Current device state	0x8000
Stopped	Current device state	0x4000
NotRunning	Device state alarm flag	0x2000
NotStopped	Device state alarm flag	0x1000
Interlockd	Flag indicating other control source	0x0800
Mode	Sequencer or Operator	0x0400
In_Use	Locks out operators; under controller control	0x0200
Setpoint	Current setpoint	0x0100
Direction	Current direction	0x0080
Ovrd_Fwd	Override forward feedback	0x0040
Ovrd_Rev	Override reverse feedback	0x0020
Travel	Current device state	0x0010
Failed	Current device state	0x0008
Man_Set	Manually trend data	0x0004
Alrm_UnAck	Alarm not acknowledged	0x0002
Active	Tag is active	0x0001

Reversible Motors — RMTR (continued)

Possible device conditions for RMTR are shown in Table 3-31. Device conditions are valid only if the Active flag is set; the controller must be programmed to support device operation.

Table 3-31 Device Conditions Derived from RMTR Status Attribute

Bit Value			Description
Running	**Stopped**		**Current Device State**
1	0		Running
0	1		Stopped
1	1		Failed
Not Running	**Not Stopped**		**Current Device State**
1	X		Not running alarm
X	1		Not stopped alarm
Setpoint	**Direction**		**Current Setpoint**
0	X		Stop
1	0		Forward
1	1		Reverse
Ovrd Fwd	**Ovrd Rev**		**Feedback Override Mode Flag**
0	0		Do not override feedback
1	0		Override forward feedback
0	1		Override reverse feedback
1	1		Override forward and reverse feedback
Travel	**Setpoint**		**Current Device State**
1	X		Travel
In Use	**Mode**	**Interlockd**	**Current Device State**
0	0	0	Hand
1	0	X	Sequencer control
X	1	X	Operator control
X	X	1	Interlocked
Failed			**Current Device State**
0			Not failed
1			Failed
X means value can be either 1 or 0. Feedback = travel limit switch indicating motor running or motor stopped. Device conditions are valid only if the Active flag is set; the controller must be programmed to support device operation. If you are not using APT, you must program the controller to change the In_Use bit.			

Timeout Attributes (timeout_1, timeout_2) These are integer attributes that can either be uploaded to an OSx station or downloaded to a controller. The domain for the attributes is SINT16.

- Timeout_1 This attribute contains the command failure timeout for the Device True state.
- Timeout_2 This attribute contains the command failure timeout for the Device False state.

Setpoint Attribute (setpoint) This is a two-bit read/write attribute that indicates the condition (stop or forward) of a commanded device. The domain is Stop, Forward, and Reverse. The values shown in Table 3-32 determine how these bits are set.

Mode Command Attribute (mode_cmd) This is a single bit read/write attribute that you use to select auto or manual control. Auto mode puts the process under controller control, and manual mode puts it under operator control. The domain is Seq (controller) and Oper (operator). The values shown in Table 3-32 determine how this bit is set. OSx does not support the 0x4000 bit of the mode_cmd attribute, while APT does. Therefore, any changes to this bit are not recorded by the autolog feature.

Reversible Motors — RMTR (continued)

Override Attribute (override) This is a two-bit read/write attribute that you use to ignore a faulty signal from a travel limit switch. The domain is None, Reverse, Forward, and Forward/Reverse. The values shown in Table 3-32 determine how these bits are set.

The device mode command must be stored in one bit location. The device setpoint and override attributes must be stored in two sequential bit locations beginning with the most significant bit. For Series 505 controllers, this is C or X/Y memory. For S5 controllers, this is a data block word location.

Table 3-32 Command Values for RMTR Attributes

Attribute	Bit Name	Bit Location	Command	Hex Value	Decimal Value
Setpoint	Setpoint Direction	0x8000 0x4000	Stop Forward Reverse	0 0x8000 0xC000	0 –32768 –16384
Mode_cmd	Mode	0x8000	Sequencer control Operator control	0 0x8000	0 –32768
Override	Ovrd_Fwd Ovrd_Rev	0x8000 0x4000	Do not ovrd fdbk Ovrd rev fdbk Ovrd fwd fdbk Ovrd fwd and rev fdbk	0 0x4000 0x8000 0xC000	0 16384 –32768 –16384

Two-Speed Motors — MTR2

The properties specific to the attributes of the two-speed motors are described below.

Tag and Descriptor Attributes These attributes identify the tag. The Tag contains the tag name and has a domain of CISTRING12. The space, the character combinations -> and <- and the characters \ , " ; are invalid for tag names. The Descriptor contains the description of the tag and has a domain of STRING30. The characters \ ; " are illegal in this field.

Status Attribute (status) The status attribute, shown in Figure 3-24, consists of 16 read-only bits. The domain for this attribute consists of the bit locations shown in Figure 3-24.

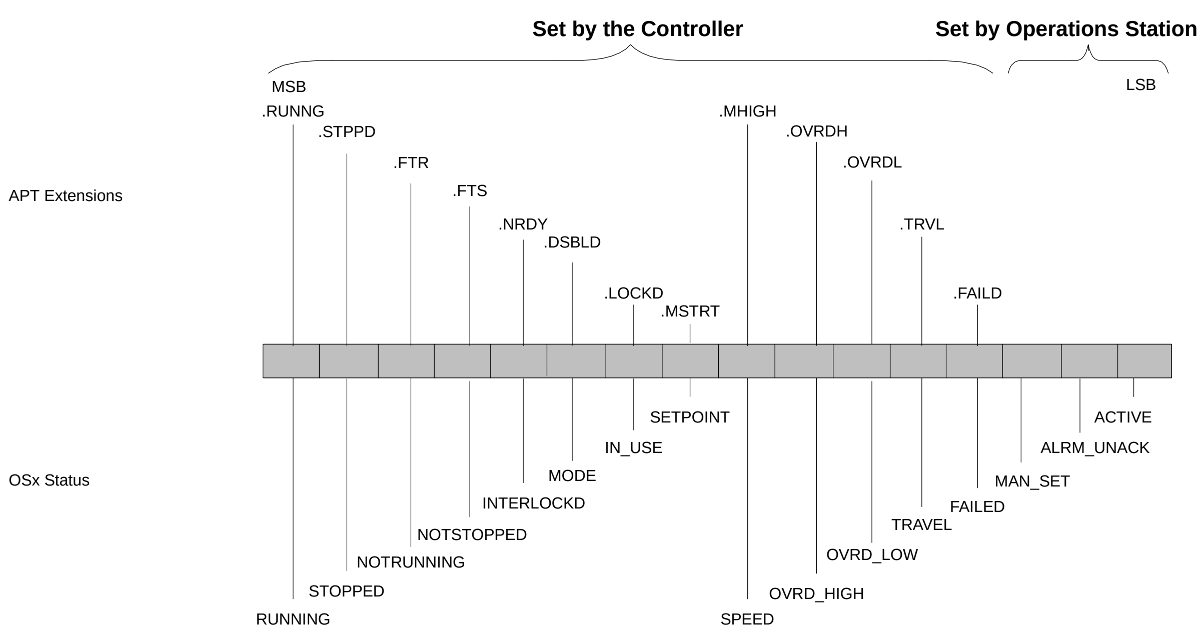


Figure 3-24 Bit Locations for MTR2 Status Attribute

Two-Speed Motors — MTR2 (continued)

Hex code locations are shown in Table 3-33. The 3 least significant bits are reserved for OSx. The Active bit shows the status of network communication for a tag, and the Alrm_UnAck bit shows that the alarms have not been acknowledged for a point. The Man_Set bit indicates whether or not the tag has been set to accept manual trending of data. The remaining 13 bits can be set either by logic or by hardware in the controller.

Table 3-33 Hex Code Locations for MTR2 Status Attribute Bits

Bit Name	Definition	Location (Hex)
Running	Current device state	0x8000
Stopped	Current device state	0x4000
NotRunning	Device state alarm flag	0x2000
NotStopped	Device state alarm flag	0x1000
Interlockd	Flag indicating other control source	0x0800
Mode	Sequencer or Operator	0x0400
In_Use	Locks out operators; under controller control	0x0200
Setpoint	Current setpoint	0x0100
Speed	Current speed; Stop	0x0080
Ovrd_High	Operating high feedback	0x0040
Ovrd_Low	Operating low feedback	0x0020
Travel	Current device state	0x0010
Failed	Current device state	0x0008
Man_Set	Manually trend data	0x0004
Alrm_UnAck	Alarm not acknowledged	0x0002
Active	Tag is active	0x0001

Possible device conditions for MTR2 are shown in Table 3-34. Device conditions are valid only if the Active flag is set; the controller must be programmed to support device operation.

Table 3-34 Device Conditions Derived from MTR2 Status Attribute

Bit Value			Description
Running	**Stopped**		**Current Device State**
1	0		Running
0	1		Stopped
1	1		Failed
Not Running	**Not Stopped**		**Current Device State**
1	X		Not running alarm
X	1		Not stopped alarm
Setpoint	**Speed**		**Current Setpoint**
0	X		Stop
1	0		Low
1	1		High
Ovrd High	**Ovrd Low**		**Feedback Override Mode Flag**
0	0		Do not override feedback
1	0		Override high feedback
0	1		Override low feedback
1	1		Override high and low feedback
Travel	**Setpoint**		**Current Device State**
1	X		Travel
In Use	**Mode**	**Interlockd**	**Current Device State**
0	0	0	Hand
1	0	X	Sequencer control
X	1	X	Operator control
X	X	1	Interlocked
Failed			**Current Device State**
0			Not failed
1			Failed

X means value can be either 1 or 0.
Feedback = travel limit switch indicating "motor running" or "motor stopped."
Device conditions are valid only if the Active flag is set; the controller must be programmed to support device operation.
If you are not using APT, you must program the controller to change the In_Use bit.

Two-Speed Motors — MTR2 (continued)

Timeout Attributes (timeout_1, timeout_2) These are integer attributes that can either be uploaded to an OSx station or downloaded to a controller. The domain for the attributes is SINT16.

- Timeout_1 This attribute contains the command failure timeout for the Device True state.
- Timeout_2 This attribute contains the command failure timeout for the Device False state.

Setpoint Attribute (setpoint) This is a two-bit read/write attribute that indicates the condition of a commanded device, for example, stop or low. The domain is Stop, Low, and High. The values shown in Table 3-35 determine how this bit is set.

Mode Command Attribute (mode_cmd) This is a read/write attribute that you use to select auto or manual control. Auto mode puts the process under controller control, and manual mode puts it under operator control. The domain is Seq (controller) and Oper (operator). The values in Table 3-35 determine how this bit is set. OSx does not support the 0x4000 bit of the mode_cmd attribute, while APT does. Therefore, any changes to this bit are not recorded by the autolog feature.

Override Attribute (override) This is a two-bit read/write attribute that you use to ignore a faulty signal from a motor run indicator. The domain is None, Low, High, and High/Low. The values shown in Table 3-35 determine how this bit is set.

The device mode command must be stored in one bit location. The device setpoint and override attributes must be stored in two sequential bit locations beginning with the most significant bit. For Series 505 controllers, this is C or X/Y memory. For S5 controllers, this is a data block word location.

Table 3-35 Command Values for MTR2 Attributes

Attribute	Bit Name	Bit Location	Command	Hex Value	Decimal Value
Setpoint	Setpoint Speed	0x8000 0x4000	Stop Low High	0 0x8000 0xC000	0 –32768 –16384
Mode_cmd	Mode	0x8000	Sequencer control Operator control	0 0x8000	0 –32768
Override	Ovrd_High Ovrd_Low	0x8000 0x4000	Do not ovrd fdbk Ovrd low fdbk Ovrd high fdbk Ovrd high and low fdbk	0 0x4000 0x8000 0xC000	0 16384 –32768 –16384

3.16 System Tags

OSx provides preconfigured system tags, identified by underscore prefixes (Table 3-36). These system tags generate alarm messages when an error occurs in the system. OSx assigns each of its error codes to a particular system tag, so that you can identify the area of the system in which the error has occurred. See Chapter 9 on configuring alarms for a complete list of system alarms and the conditions that cause them.

When you display tag details for the _ROOT_FILSYS and _OSX_FILSYS tags, you can set alarm limits for the HH Alarm and H Alarm attributes. For example, if you enter 95% for the _ROOT_FILSYS HH Alarm, an alarm occurs when this file system reaches 95% capacity. The value attributes of these tags display the current disk usage in these file systems.

OSx is designed as two file systems, the root (**/**) file system and the **/usr/tistar** file system. The _ROOT_FILSYS alarm tag monitors disk usage on the root file system, and the _OSX_FILSYS alarm tag monitors disk usage on the **/usr/tistar** file system.

The value attribute of the _HT_COLLECT system tag displays a continually updated average of the number of points trended per second.

Table 3-36 Directory of System Tags

System Tags					
_ACTREQ	_DATA_BASE	_HT_COLLECT	_OSX_FILSYS	_ROOT_FILSYS	_TAG_DETAIL
_ALARMING	_DATA_XFR	_HT_DISPLAY	_PAGE_SERVER	_RT_DISPLAY	_TAG_GROUP
_ARCHIVING	_DIAGNOSTIC	_HT_UPDATE	_RECIPE	_SCHEDULER	-TAPE_DRIVE
_BATCH	_FO_CIRCUIT	_LIBRARIAN	_REDUNDANCY	_SECONDARY	_TIME_SYNC
_CHANGE_LOG	_GENERAL	_NODE_SYNC	_REDUND_SYNC	_SPOOLER	_UNIX
_CONFIG	_H1_COMM	_OPERATOR	_REPORTS	_SYS_CONTROL	_VERIFY

When you display the tag detail for the _REPORTS system tag, you can set values for the High High and the High Alarms for the report relation in the database. For example, if you enter 95% for the High High Alarm, an alarm occurs when the database relation containing reports reaches 95% capacity.

Figure 3-25 shows the bit locations for a system tag status attribute. These bits are defined in Table 3-37.

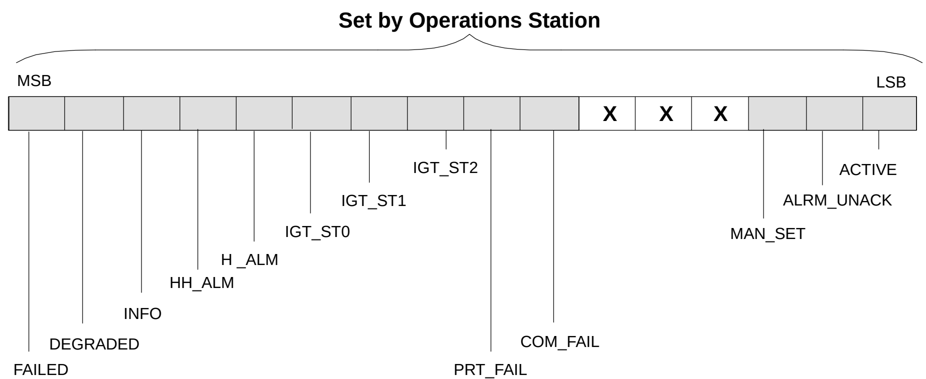


Figure 3-25 Bit Locations for a System Tag Status Attribute

Table 3-37 Hex Code Locations for System Tag Status Attribute Bits

Bit Name	Definition	Location (Hex)
Failed	Subsystem failed	0x8000
Degraded	Degraded performance	0x4000
Info	Information alarm	0x2000
HH_Alm	Disk at capacity	0x1000
H_Alm	Disk near capacity	0x0800
IGT_ST0	Reserved	0x0400
IGT_ST1	Reserved	0x0200
IGT_ST2	Reserved	0x0100
Prt_Fail	Printer fail	0x0080
Com_Fail	Communication lost	0x0040
Man_Set	Manually trend data	0x0004
Alrm_UnAck	Alarm not acknowledged	0x0002
Active	Tag is active	0x0001

3.17 Considerations for Designing the Controller Program

Organizing Data in the Controller

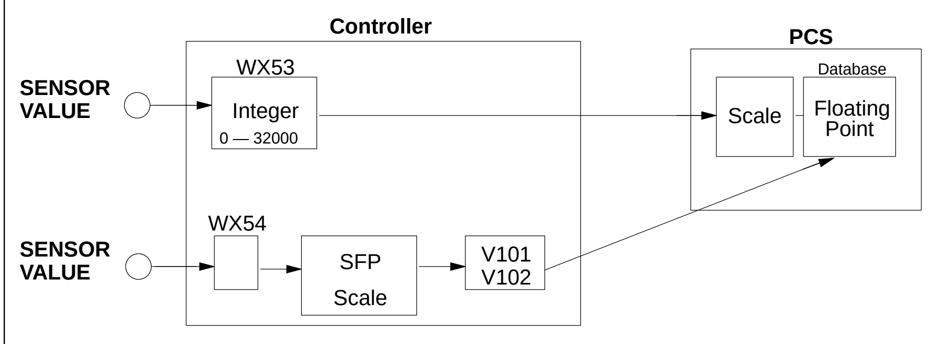
How you organize data in your programmable controllers affects the overall performance level of the SIMATIC PCS 7 OSx system. For maximum network throughput and efficiency, distribute the tags as evenly as possible among the controllers in the system, and group like tag types in contiguous address locations.

NOTE: Failure to group like tag types in contiguous address locations has a significant effect on network throughput.

Scaling Controller Values

The SIMATIC PCS 7 OSx system contains the properly scaled values. The system automatically scales integer values uploaded from the controller before writing them to the database. OSx does not scale floating point values from the controller. If the controller has a special function for scaling, values are scaled in the controller and written directly to the database, as shown in Figure 3-26.

Where you do scaling, whether in the controller or in the OSx station, has a direct effect on performance. Contact your system integrator for assistance in evaluating performance requirements.



$$\text{Scaled Value} = \text{Low Range} + \left[\frac{\text{Value in Controller}}{32000} \times (\text{High Range} - \text{Low Range})\right]$$

Figure 3-26 Scaling Values from Analog and Loop Input

NOTE: For S5 controllers, you must have the controller do scaling operations. For Series 505 controllers, use word values of 0–32000; the OSx station scales the value into the FLOAT32 value stored in the database.

Unscaling Database Values

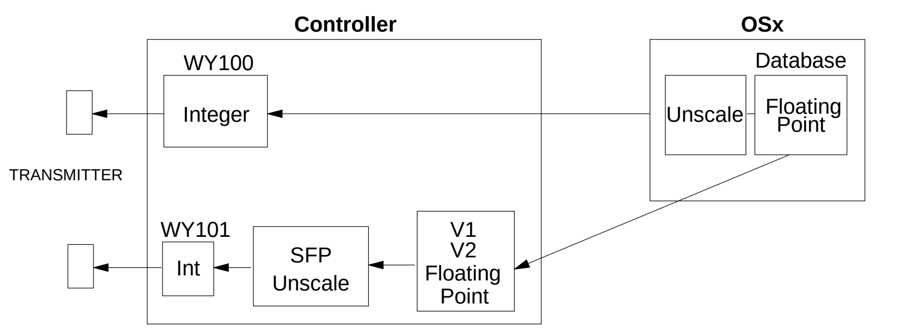
When the system writes scaled values in the database to be stored in a controller as integer values, it unscales them automatically (Figure 3-27).

For example, if you use an analog output to control the speed of a conveyor belt drive motor, the associated AO tag has the following configuration: Low range = 400 rpm; High range = 600 rpm; Output = V-memory location V105. The low and high ranges are, in this situation, used for scaling bar icon symbols, boundary-limit checking of entered values, and so forth.

Assuming no 20% offset, a value of 400 entered at the OSx station sets V105 to 0, an entry of 500 sets V105 to 16000, and an entry of 600 sets V105 to 32000. See Figure 3-27 for the formula used to calculate these values.

If the same analog output tag is configured to a floating point location in the controller (for example, V.105) then V locations 105 and 106 are used together to represent a floating point value. In this case, an entry of 400 at the OSx station sets V.105 to 400.0. Similarly, an entry of 500.6 at the OSx station sets V.105 to 500.6. The floating point value is in IEEE format in the Series 505 controllers. For the S5 controllers, the value is converted to the S5 floating point value.

Loop outputs represent percentages of a given range and are scaled similarly (0–100% only).



$$\textit{Value in Controller} = \left[\frac{\textit{Scaled Value} - \textit{Low Range}}{\textit{High Range} - \textit{Low Range}}\right] \times 32000$$

Figure 3-27 Scaling Values from Analog Output

3.18 Planning the Tag Configuration

Steps to Consider

To ensure a more efficient and error-free tag configuration, read the information presented in Chapter 4 and Chapter 5. Then follow the steps outlined in Figure 3-28.

Use the tag configuration planning sheets in Appendix A when you review Chapter 4 and Chapter 5. All the information presented on the planning sheets is described in these chapters and is required for tag configuration.

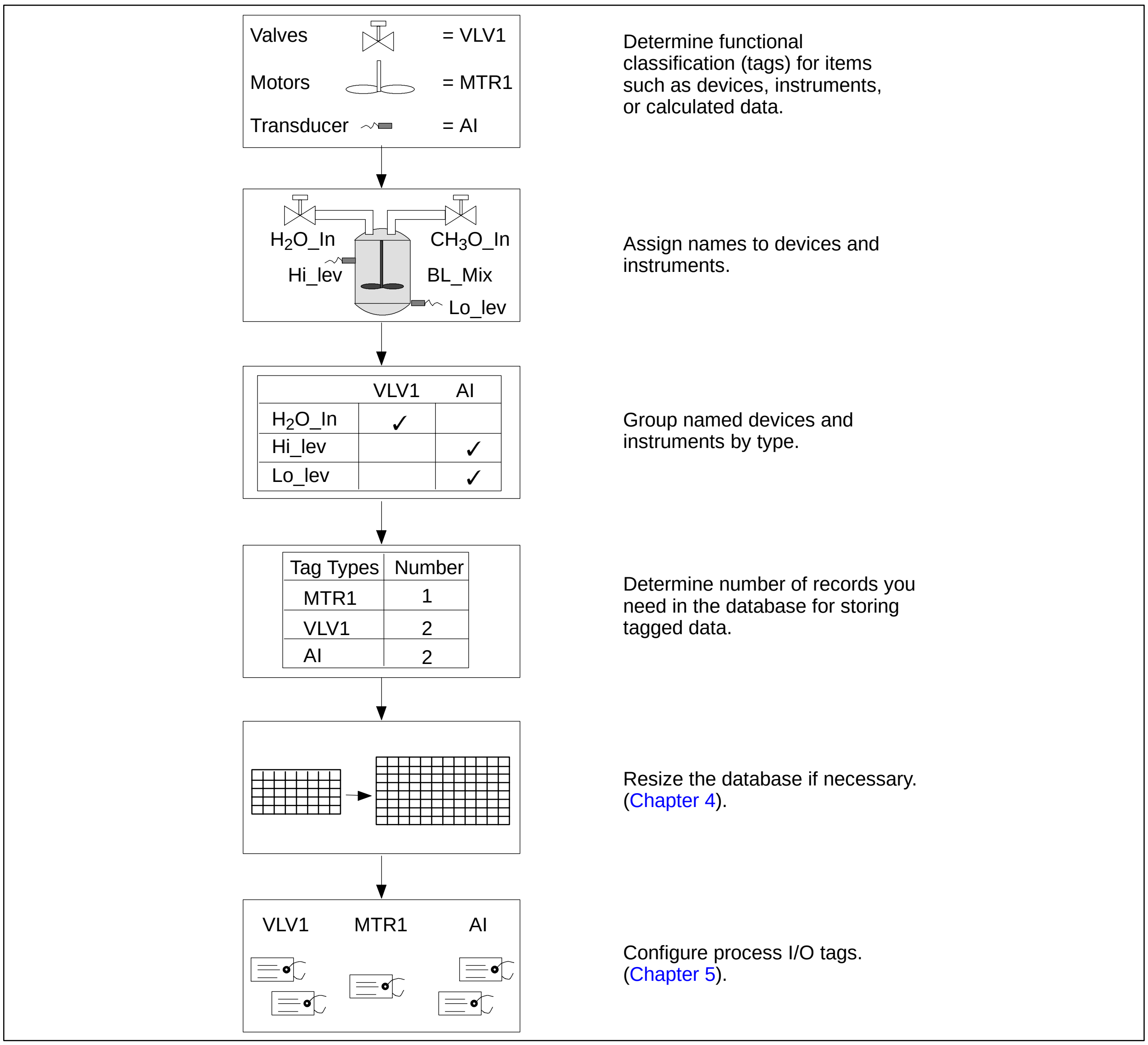


Figure 3-28 Recommended Prerequisites for Defining Tags

Chapter 4

Configuring Tag Capacities

4.1 Determining Memory Requirements for Tag Configuration

Memory Requirements for Each Tag Type

The OSx system operates with a memory-resident database. You can use memory allocated for the database to configure approximately 10,000 tags. Table 4-1 shows the default allocations for a 3,000 tag and nominal 10,000 tag system.

Table 4-1 System Allocations

Tag Type	Number of Tags	
	3,000 Tag System[1]	10,000 Tag System
Analog Input	300	2,000
Analog Output	200	1,000
Digital Input	750	2,500
Digital Output	300	500
Device	400	1,500
Calculated Variable	500	250
Integer Variable	500	250
Timer	20	150
Counter	20	150
Loop	100	1,000
System	50	50
10-bit Digital Input	50	250
10-bit Digital Output	50	250
Text	60	200
Batch	50	100
Unit	50	200
Area	50	50

Trend Groups	OSx[1]
Trend Group	800
Analog/Digital Variables per Group	4/4
Total Variables Trended (Analog/Digital)	3,200/3,200
Sample Rate (fastest)	1/sec
Average Samples	75/sec
Max. Number of Online Historical Samples	15,000,000
Table continued on next page.	

[1] Default configuration, 3,000 tag system

Table 4-1 System Allocations (continued)

Displays	OSx[1]
Point Detail	10,000[2]
Tag Group	1,260
Alarm Group Summary	850
Alarm Group	850

Custom Graphics	OSx[1]
Total Number of User-Defined Graphics	500
Total Number of Live Data Points per Custom Graphic	384
Library Size of Symbols	300
ISA Symbols in Library	21

Reports	OSx[1]
Number of Reports	150
Standard Default	28
Pages per Report (Typical)	2
Variables per Page	80
Pages of Report History Stored	2,000

Recipes	OSx[1]
Number of Recipes	500
Maximum Number of Recipe Components	30,000
Maximum Number of Unique Recipe Components	7,500
Table continued on next page.	

[1] Default configuration, 3,000 tag system
[2] Varies depending on tag mix

Table 4-1 System Allocations (continued)

Action Requests	**OSx**[1]
Number of Action Request Messages	3,000
Action Request History	3,000

Miscellaneous	**OSx**[1]
Alarm Reporting	5,000 Tags
Operator Change Log	8,000 Changes
Diagnostics	Supported
Alarm Message History	25,000
Process Groups	32
Printers	31

[1] Default configuration, 3,000 tag system
[2] Varies depending on tag mix

4.2 Tag Capacity Configuration Overview

Tag capacity configuration enables you to specify the number of records required to store configuration data for each tag type. To configure tag capacities, you must have either the Database Administration or the System Configuration security privilege. You can configure tag capacities for each type of tag within the SIMATIC PCS 7 OSx system.

For a multiple-station system, you configure tag capacities only on an OSx station that has the sysadmin role. In a single-station system, the primary supports this function.

If necessary, you can increase the number of records for each tag type and also allow a reasonable number of extra records for future expansion. As long as a relation has space for another tag, you can add tags without having to increase the capacity for that tag type. If a relation does not have space for another tag, you must increase the tag capacity before you add another tag. Be aware, however, that extra records allocated for one tag type may take up memory required for another type of tag. Tag capacity configuration automatically increases the size of the database memory if necessary to support the increased tag capacities.

Before you can change tag capacities, you must set the OSx station with the sysadmin role in a multiple-station system or the primary in a single-station system to the Offline state. If you use an OSx station with the sysadmin role see the chapter on Multiple-Station Operations in the *SIMATIC PCS 7 OSx System Administration Manual* for further information. When the OSx station begins to resize the database, it changes to the Resize state. During the Resize state, the OSx station logs you off and closes all windows to prevent input while the database is unloading. When resizing is complete, the station changes to the Offline state. You must log on again to resume configuration.

The default tag capacity may allow more tags of a particular tag type than your process requires. When this is true, you can decrease the maximum number of records for a tag type to provide additional memory for other tag types. Follow the same general procedure for decreasing that you use to increase the maximum number of records. The smallest number of records is one.

NOTE: If you need to decrease the maximum number of tags for a particular tag type, you can save time by doing the downsizing operation prior to increasing the maximum number of tags for some other tag type.

4.3 Configuring Tag Capacities

Checking Current Tag Capacities

To check the currently allocated tag capacities, select **Tags->Capacity** from the menu bar (Figure 4-1).

You can view the Tag Capacities Configuration screen while the system is in any state, but OSx must be in the Offline state before you can change the capacities.

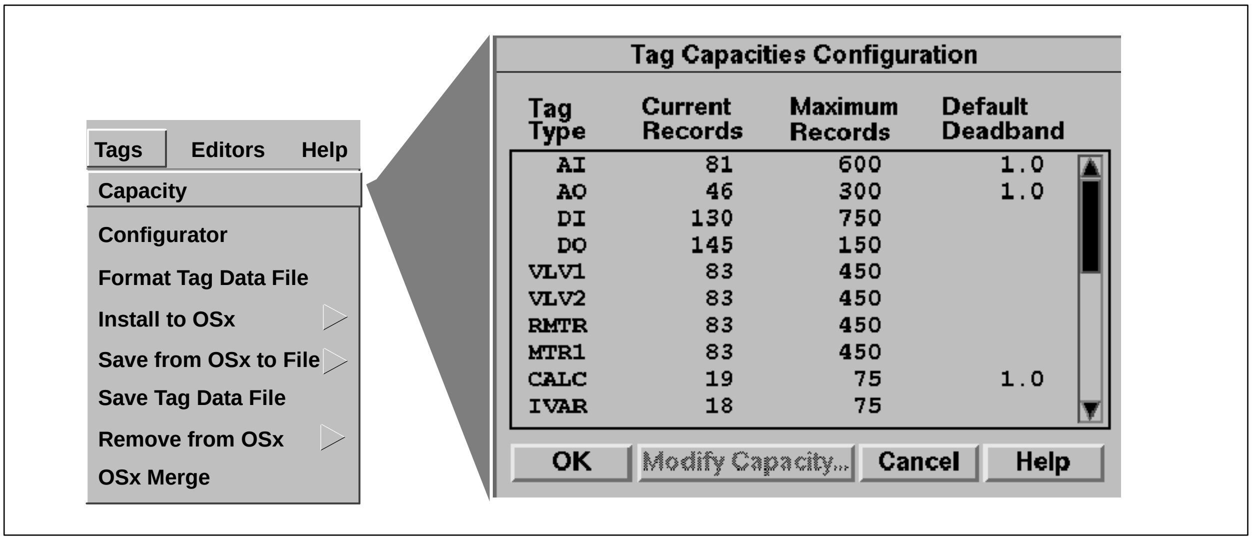


Figure 4-1 Tag Capacity Display

Tag Type Standard tag type names (e.g., AI, DO) and user-defined tag types within the system.

Current Records Number of tags currently programmed into the system for the tag type.

Maximum Records Maximum number of tags that may be configured for the tag type with the memory currently configured for the tag type.

Default Deadband Range (in percent) of span that a measured signal must vary before the value is updated in the database and sent to the OSx stations.

NOTE: Selecting the **OK** button cancels the Tag Configuration choices and returns you to the main menu.

The way the system handles the maximum record number for devices is different from that used for the other tag types. For devices, the Maximum Records column in the Tag Capacity Table lists for each device type the total allowed for all devices (Table 4-2).

In the example below, the value 60 appears in the Maximum Records column for each of the five device types, but the actual number of tags assigned varies with each device as shown in the Tags Assigned column.

Table 4-2 Tag Capacity Table

Device Type	Maximum Records	Tags Assigned
VLV1	60	20
VLV2	60	10
MTR1	60	20
MTR2	60	10
RMTR	60	0
Total Number of Devices		60

Changing Current Tag Capacities

To modify tag capacities, you must have either the Database Administration or the System Configuration security privilege. Make sure the OSx station with the sysadmin role or the single-station primary is in the Offline state, and always exit the graphical editor before modifying tag capacity. If you do not exit the graphical editor when the OSx station transitions to the Resize state, the graphical editor terminates and the system reports an error message.

NOTE: The system may require an hour or longer to resize the database.

To modify tag capacity, follow the steps below.

1. Select the appropriate tag types from the Tag Capacities Configuration display (page 4-6).
2. Select **Modify Capacity...** from the Tag Capacities Configuration display (Figure 4-2).
3. Use the **Next** and **Previous** pushbuttons to page through the tag types you have selected. Enter new values as needed.
4. Select the **OK** button from the Modify Tag Capacity display (Figure 4-2) to display a confirmation window for starting the process. OSx begins resizing the database. If the resizing fails, OSx restores the original database. If the resizing succeeds, OSx creates a new database.
5. To perform additional tasks, you must log onto the system again.

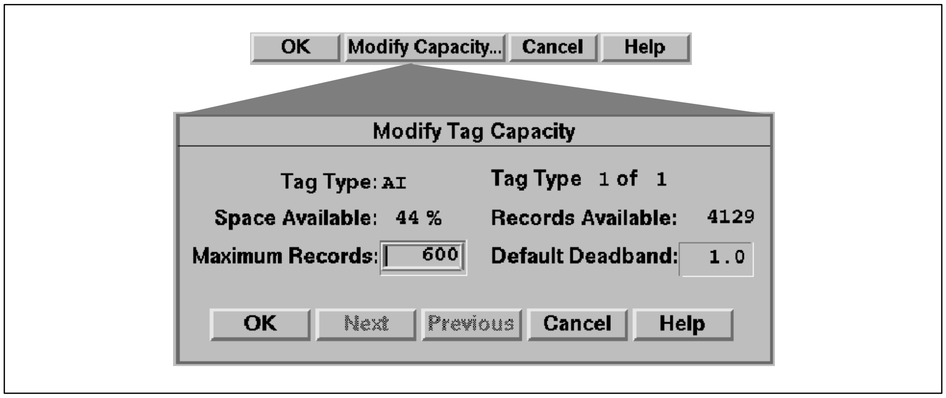


Figure 4-2 Modify Tag Capacity Display

Space Available Percentage of memory available for configuring all tags; based on the maximum number of records defined for all tag types.

Records Available Number of additional tags of the selected tag type that could be configured; based on the percentage of memory available and the maximum records allowable for the selected tag type.

Maximum Records Maximum number of tags that may be configured for the tag type given the memory currently configured for the tag type.

Default Deadband Range (in percent) of span that a measured signal must vary before the value is updated in the database and sent to the OSx stations.

Next and **Previous** Use the **Next** and **Previous** buttons to toggle between multiple tag selections.

NOTE: Modifying tag capacities causes the configured local printer of an OSx station to eject a page and to print a page header for each transition to and from the Resize state. If this is unacceptable, delete the configuration for the local printer before you modify tag capacities. Reconfigure the printer after you complete changes to tag capacities. Network printers and local printers on other OSx stations are not affected.

Changing the Default Database Deadband

A deadband value is the range that the current value of a scanned analog attribute is allowed to vary before the system stores a new value in the database and sends it to the OSx stations. Only five tag types have an analog deadband: CALC, TMR, LOOP, AI, and AO. Deadband is calculated as a percentage of range. Each new current value causes a recalculation of the high and low deadband limits. OSx provides a default deadband of 1% for each tag type that has an attribute for high and low range.

You can change the default database deadband value on a tag-by-tag basis by modifying the value of the tag's change attribute. You do this by marking tags in APT, or by entering a new value for the change attribute in the **install.tag** file that you create from a spreadsheet or ASCII editor. See Chapter 5, which describes how to configure tags and their attributes.

You can change the default database deadband value for all tags of a particular type when you modify tag capacity. After selecting the tag type, enter the new value. The new deadband applies on all subsequently configured tags of the associated tag type and appears on all configuration screens for that tag type.

The example in Figure 4-3 shows a tag with a current analog value of 50° and a deadband of one percent (50° x .01 = .50°). If the newly scanned analog value exceeds the high limit of 50.5° or the low limit of 49.5°, the system records a new current value in the database and displays it on the OSx station screen. This process repeats with the new current value.

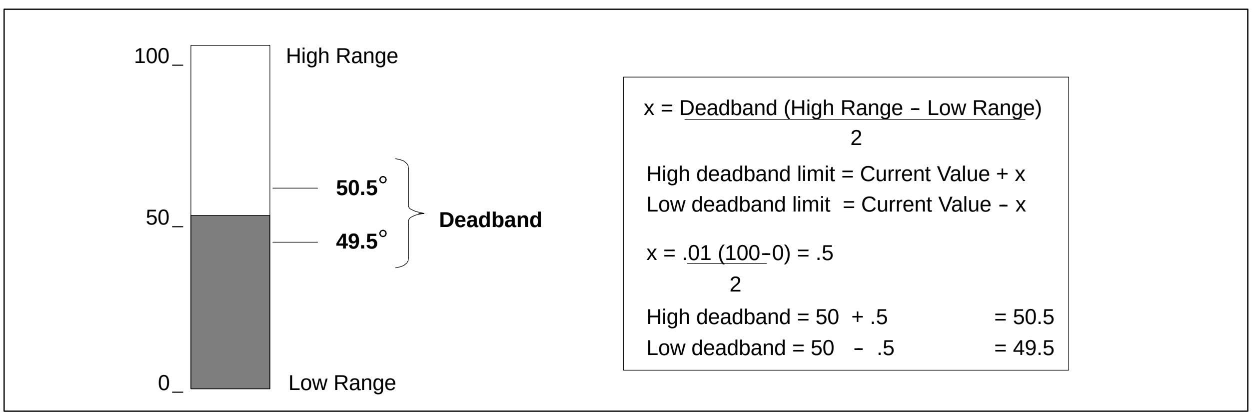


Figure 4-3 Sample Deadband Value

As long as the value stays within the high and low deadband limits, the value remains the same in the database. If the value exceeds the deadband limits, the system records the new value in the database. For example, if the value reaches 51°, the system records that value in the database, and the deadband limits are now 51.5° and 50.5°.

The deadband allows you to match database update to sensor accuracy. Setting the deadband for less than the noise level of the signal reduces system performance significantly.

Using Deadband Values in a Multiple-Station System

When you assign a deadband to the change attribute for a networked tag in a multiple-station system, the database for each OSx station can potentially have differing values (up to the deadband value) for the same tag. For example, station A might read the value of a tag with a deadband of 1% as 16,000. However, if the value has changed to 16,100 when station B reads the tag, station B displays 16,100, but station A continues to display 16,000 because the change is less than the deadband.

NOTE: If it is important for each OSx station to display exactly the same value, assign a deadband of 0% to the tag. Note that when you do not use deadbands, the system updates the database with every change. This can have an adverse effect on network throughput.

Chapter 5

Configuring 505 and S5 Tags

5.1 Choosing Method of Configuring Tags

Figure 5-1 shows the general procedure for configuring and installing 505 and S5 tags for your application. You must have either the Database Admin or the System Config security privilege to install tags.

NOTE: For configuring and installing S7 tags, see the *SIMATIC PCS 7 OSx Interface to S7 Controllers Manual.*

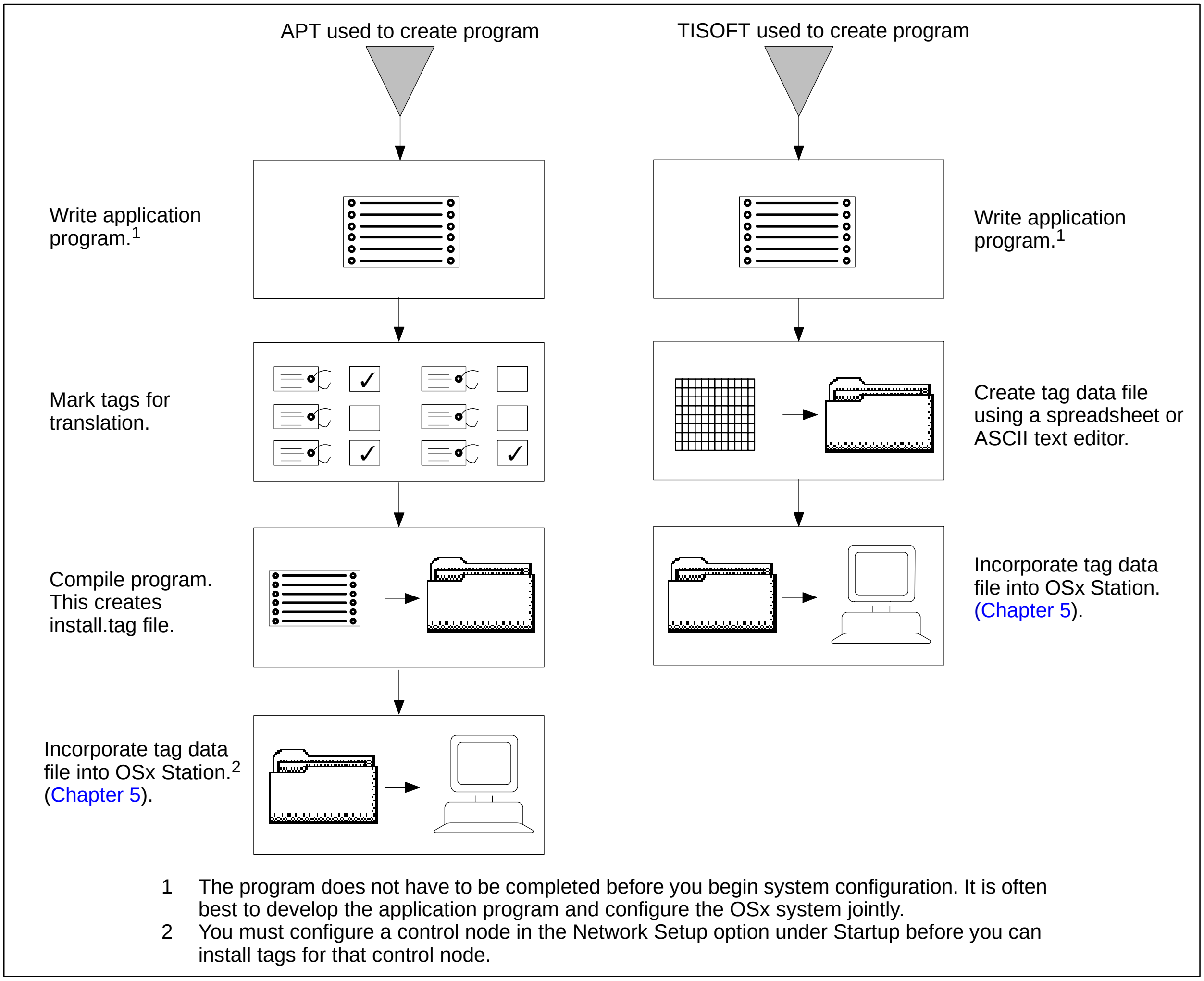


Figure 5-1 Sequence of Tasks in Tag Configuration

You can choose from four methods to configure tags for your process:

Use APT Create the application program using APT. Then mark the names that you assigned to objects for translation into tags (Section 5.3).

Use Tag Configurator The Tag Configurator provides a means for entering tags one at a time. This approach is normally used to add, modify, or delete a few tags after the bulk of tags have been entered into the OSx database using either APT or a spreadsheet.

Create the application program for Series 505 controllers using TISOFT SoftShop or APT. For S5 controllers, use the STEP 5 programming software or APT. Then configure tags using the Tag Configurator (Section 5.4). This method allows you to add, modify, view, and delete tags without leaving OSx. The Tag Configurator can run while OSx is in the Offline or Operate state.

Use a Spreadsheet Program Create the application program using your preferred programming software. Then create a spreadsheet and enter tag information into the spreadsheet that can then be incorporated into the database (Section 5.5). The spreadsheet must allow you to save your data in an ASCII comma-separated value (CSV) format in which the variables are separated by commas.

If you need help creating the file that contains the tag data (**install.tag**), you can examine a template file. For Series 505 controllers, the file, called **install.tag**, is located in the **/usr/tistar/misc** directory, and you can access it by logging in from an OSx Terminal window. For S5 controllers, the file is called **install.s5** and is located in the same directory. Refer to the *SIMATIC PCS 7 OSx Interface to S5 Controllers User Manual* for more information about how to create tag data files for S5 controllers.

Use any ASCII Text Editor Create the application program using TISOFT. Then create an ASCII file and enter tag information into the file that you can then incorporate into the database (Section 5.6).

NOTE: You can use the Tag Configurator or a spreadsheet or an ASCII text editor to create tag data, even when the application program is created in APT. You can also use these tools to edit the tag data file that is created by APT, but changes you make are not incorporated into the APT program.

5.2 Considerations for Configuring Tags

Guidelines

When you configure tags without using APT, follow these guidelines:

- Make sure that the addresses for network points are within the range of memory configured in the controller.
- When you create your own **install.tag** file, place any UNIT tags that have other tags associated with them at the beginning of the **install.tag** file.
- When you are using the Tag Configurator, you must configure any UNIT tags before you configure the tags associated with them.
- If you do not network any of the attributes (if you leave the Memory and Location fields blank), the tag will be saved as a non-networked (OSx) tag.

For Fastest Installation

Installing tags involves significant database activity. If possible, set the system to the Offline state first. If you must install tags with the system in the Operate state, remember that the role of the station on which you install tags affects efficiency, and choose the station according to the following priorities. The first choice provides the fastest method.

- If a station on the system has the sysadmin role, install the tags on this station.
- Install tags on the primary with the other stations set to out_of_service.
- Install tags on the primary with the other stations set to normal.
- Install tags on a station that is not the primary.

For the first two cases, OSx updates only one database, permitting faster tag installation. The new database is propagated to the other stations when they join or synchronize with the primary. For the last two cases, the database on each station is updated as each tag is installed, which causes tag installation to take significantly longer.

Unless you are installing only a few tags, avoid installing tags while the system is in the Operate state.

Considerations for Configuring Tags (continued)

RBE and Tag Installation

Some categories of information are reported by the control node automatically, without a demand scan from an OSx station. Examples include alarms and alarm acknowledgements, the automatic logging of messages, action requests, BCL jobs, and recipe handshaking exchanges. This process is called report by exception (RBE).

Although you can add new tags in either the Offline or the Operate state, installation is faster when the system is Offline. Tags installed during the Operate state are not treated as RBE points, but rather as event-scanned points. The system scans these points in the control node at the Event Scan Period (see the chapter on System Architecture in the *SIMATIC PCS 7 OSx System Administration Manual*). Therefore, the system reads any Status attributes configured for autologging from the control node at the Event Scan Period. If you do install tags while the system is in the Operate state, then the next time you change OSx from Offline to Operate, the autologged Status attributes are treated as RBE points.

RBE Memory Allocation and Series 505 Controllers

The Series 505 controllers have factory-allocated memory to store 20K bytes of RBE definitions. If the number of tags in a given controller requires more than 20K, you must determine the total memory required for all tags and enter the amount in the APT Compiler Control file. If the OSx database defines more RBE tags than the reserved space permits, the controller uses unconfigured user memory to hold additional RBE tag definitions. If you use TISOFT instead of APT to program the controller, it is necessary to leave some memory unconfigured if it is likely that more than 20K bytes is needed for RBE tag definitions.

The memory required for a given tag depends on the tag type (Table 5-1). For a typical OSx tag mix, the average memory requirement is 42 bytes per tag. This means that you can define approximately 490 tags in a given controller before the RBE tag definitions require unconfigured user memory. Depending on the memory requirements of the controller program, it is possible to define a larger number of tags. However, the amount of unconfigured controller user memory available for RBE definitions can change over time if you make changes in the controller program.

If insufficient controller memory is available for RBE, then the system handles tags that could not be stored as RBE definitions in the controller as demand scan tags. The system scans these tags continually at the Event Scan Period. Refer to the System Architecture chapter in the *SIMATIC PCS 7 OSx System Administration Manual* for more information about RBE and demand scan.

When you use APT, you can either view the compiled report that contains Tag Type Usage in Phase 7, “Translate File Builds,” or create a tag report from the object file reports. The tag report lists the OSx tag types and information about the attributes. The report is not as detailed as the **install.tag** file, but APT can display it. These reports list the total number of tags for each tag type that you can use in calculations for memory requirements.

The planning sheet in Appendix A can help you determine memory requirements for tags.

Table 5-1 RBE Memory Requirements for Series 505 Controllers

Tag Type	Controller Memory Requirement (Bytes per Tag)	Tag Type	Controller Memory Requirement (Bytes per Tag)
AI	42	DO	28
AO	42	DO10	28
AREA	42	IVAR	36
CALC	36	LOOP	54
CTR	42	TEXT	48
DEVICE	48	TMR	42
DI	28	UNIT	48
DI10	28		

RBE Memory Allocation and S5 Controllers

The SIMATIC S5 controller programs require a function block package for the RBE process. This function block is configured to hold either 250 or 500 RBE definitions of any tag mix. Refer to the *SIMATIC PCS 7 OSx Interface to S5 Controllers User Manual* for more information.

5.3 Configuring Tags with APT

Use APT for designing and creating the application program. APT is a graphic programming environment that eliminates the need for you to work in relay ladder logic and allows you to develop a structured program. Tag configuration is a simpler and more straightforward process with APT.

Marking Tags

To mark the APT objects that you need to use as tags, follow the procedure below. Refer to your APT documentation for specific details.

1. If you have not already done so, create the matrix described in Section 2.2 of this manual. Using the matrix, assign the tags to the desired process groups.

2. Run APT.

3. Select an APT object for translation, mark it, and then choose the **Set Process Group** option.

4. Enter the description for the process group that you want to associate with the tag. You can also enter descriptions for all the remaining process groups, even if you don't want to associate this tag with them.

   If you prefer, you can assign a tag to a process group and enter the process group description later, as described in Chapter 2. You link tags to process groups by group number, not by the description. The disadvantage to this approach is that your process group descriptions are not documented within your APT program.

5. Select the process groups to be associated with the tag.

6. Return to the **Mark Tags** menu and select the next tag to be marked.

Creating the Tag Data File

The tag data file is created when you compile the APT program. To compile a program, follow the procedure below:

- From within the APT environment, display the **Compiler Control** file.
- Follow the directions in your APT documentation for filling out the **Compiler Control** file. For the **Build Translate Table** option, select **Yes** if this is the first time that you are translating tags. You can select **Append** if you are adding tags to an existing database of tags.
- Compile the APT program. APT creates a file that contains tag data. The file is called **install.tag.**

OSx and APT no longer allow you to install tags from an engineering work station when both APT options **Append tags only** and **Delete unused tags on secondary** are selected at the same time. APT Release 1.8 and later disallows this at the engineering work station. Versions of APT prior to 1.8 still allow selecting both options, but report **PCS Communications Error** when the tag installation is rejected by OSx.

Transferring Tag Data File to an OSx Station

You must transfer the tag data file called **install.tag** to an OSx station. You must have either the Database Admin or the System Config security privilege to install tags.

You can change or delete tags only in the Offline state. You can add new tags in either the Offline or the Operate state, but the Offline state is recommended. The system does not collect data from the network as efficiently when you install tags in the Operate state. This is because the system does not treat the tags as RBE points. When an RBE point changes, each OSx station receives the same changed value. Non-RBE points are scanned by each OSx station individually, and thus each station receives its own value at the time of the scan. Install tags in the Operate state only when you have an immediate need to install tags and when you cannot set the system to the Offline state. See Section 5.2 for more information about tag installation and RBE processing.

You must configure a control node in the **Network Setup** option under **Startup** before you can install tags for that control node. You can transfer the **install.tag** file to an OSx station through the Industrial Ethernet network. Select the APT translate option and follow the prompts in the Translate dialog box. To install the tag data file on an OSx station, follow the procedure described on page 5-10.

To install the tag data file on an OSx station, follow the steps below.

1. Copy the tag data file on your APT computer to a diskette. The APT compile process places the **install.tag** file under the **\apt\program**<program_name> directory, where <program_name> is the name of your APT program.

2. Place the diskette in the disk drive of an OSx station.

3. Select **Tags->Install to OSx->From APT** from the menu bar (Figure 5-2). The Tag Installation Operations dialog box appears (Figure 5-3).

4. To install the APT tag file from a diskette, click the button to the left of **Diskette**.

   If you are installing the APT tag file from your hard drive where it is placed after a network transfer in the APT default directory (**/usr/tistar/misc**) and file (**install.tag**), click the button to the left of **Hard Disk**.

   To install from an APT file that is located somewhere else than the APT default directory and file, use the **From Tag Data File** option (page 5-30).

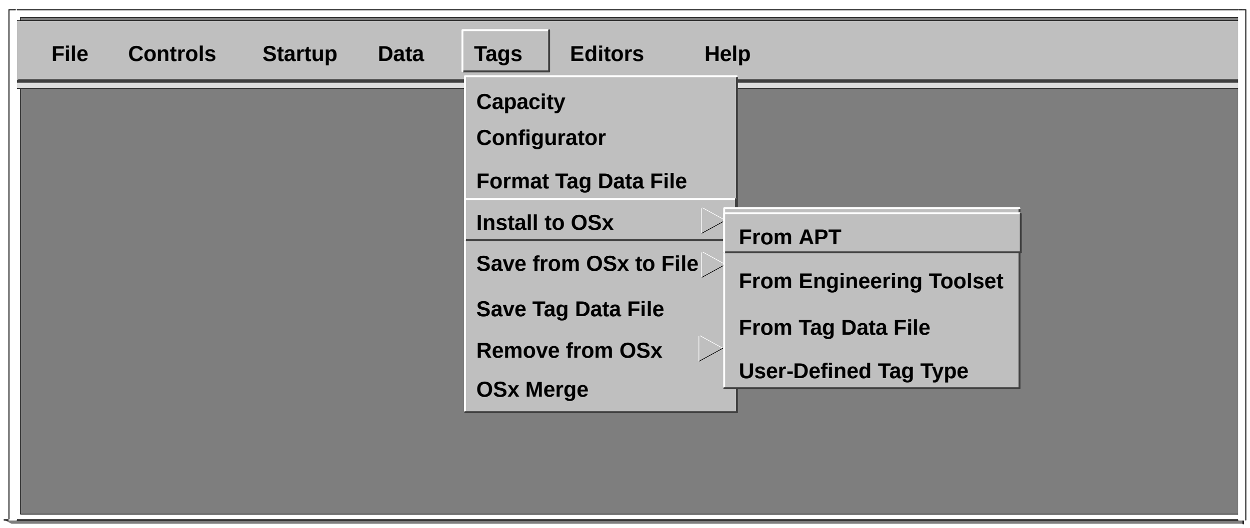


Figure 5-2 Installing Tags from APT

5. Click the **Start** button to begin the tag installation. When installation is complete, the tag installation report is displayed.

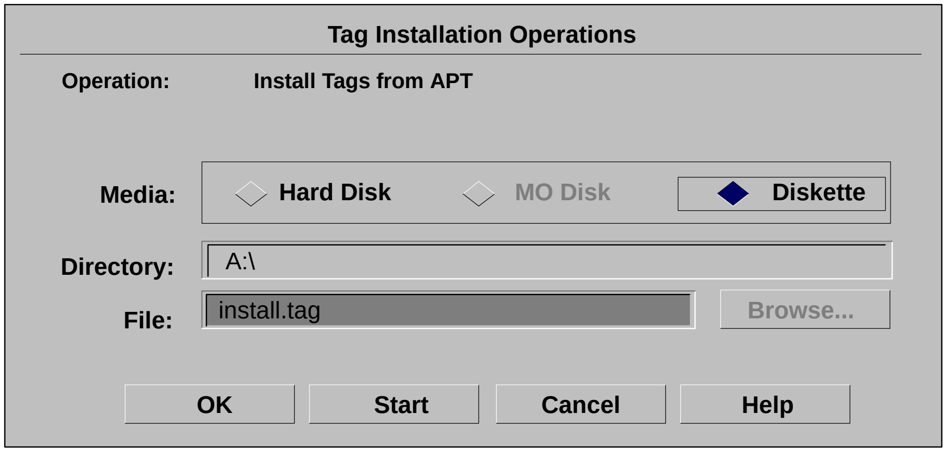


Figure 5-3 Installing Tags from APT

Examining the Tag Data

You can examine these files by using any ASCII text editor. These files appear in the format described in Section 5.6 of this manual.

If the primary is powered down while you are installing tags, incomplete tags may remain in the database. If you re-establish this station as the primary, its database with the incomplete tags is used.

To verify the presence of an incomplete tag, check the **log.out** file in the **/usr/tistar/data** directory for error messages. A message in the Operate state that is preceded by **Error 598: Comm info** may indicate an incomplete tag. You must delete the associated tag and reinstall it. See the OSx station troubleshooting chapter of the *SIMATIC PCS 7 OSx System Administration Manual.*

5.4 Configuring Tags with the Tag Configurator

You can configure tags using the Tag Configurator. This method allows you to add, modify, view, and delete tags without leaving OSx. You must have either the Database Admin or the System Config security privilege to access the Tag Configurator. The Tag Configurator can run while OSx is in the Offline or Operate state. However, deletions are restricted to the Offline state and only certain modifications can be made in the Operate state.

You must configure a control node before you can add a tag to that node. For Series 505 controllers, you must use TISOFT, SoftShop, or APT to write the application program. For S5 controllers, use the STEP 5 programming software or APT. For S7 controllers, use the Engineering Toolset and refer to the *SIMATIC PCS 7 OSx Interface to S7 Controllers Manual*.

NOTE: When you are using the Tag Configurator, you must configure any UNIT tags before you configure the tags associated with them.

Figure 5-4 shows the Tag Configurator while the system is in the Offline state. The fields are described on the following pages.

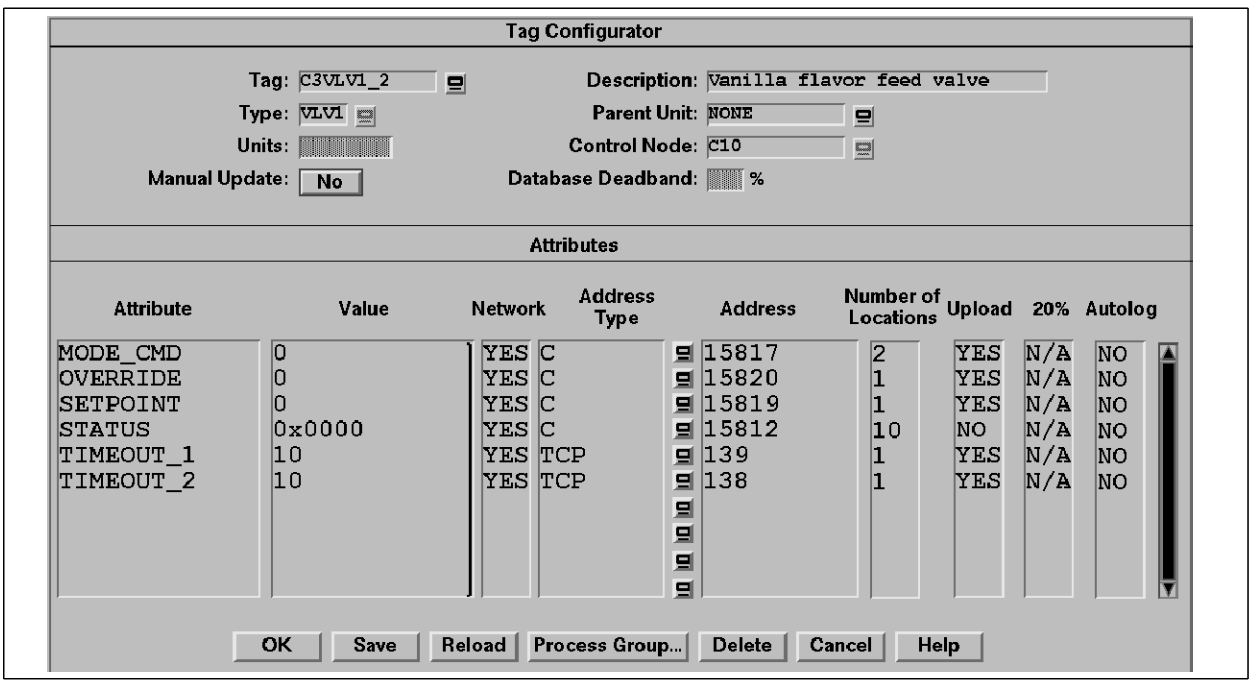


Figure 5-4 Tag Configurator

Tag Enter the name of the process I/O tag that you are configuring. The space, the character combinations –> and <– and the characters \ , ” ; are invalid for tag names. The name must begin with an alphanumeric character (0–9, A–Z) or the underscore (_) character. You cannot use a numeric value such as an integer, float, hex, or scientific notation for tag names. The maximum length of this field is 12 characters. If you are modifying, viewing, or deleting an existing tag, you can click the long-list display tool to the right of the field and select a tag from the list that appears.

Type Click the long-list display tool to the right of the field and select a tag type from the list that appears.

Units Enter the engineering units used for this tag, such as tons or liters. The maximum length for this field is 8 characters. The characters \ ” ; are invalid.

Manual Update Select **Yes** if you want to be able to enter data manually for the tag for trending purposes.

Description Enter a meaningful description of the tag, up to 30 characters in length. The characters \ ” ; are invalid.

Parent Unit Use the long-list display tool to the right of the field to select the unit tag used as the parent unit, if you assign the tag to a unit.

Control Node Use the long-list display tool to the right of the field to select the control node associated with this tag. If you select **OSx**, the tag is a non-networked tag. If you select a control node, but do not network any of the attributes, the tag will be saved as a non-networked (OSx) tag.

Database Deadband Enter the range in percent of span that the current value varies before the system updates the value in the database. The range of this value is from 0 to 100. Use a non-zero deadband to improve system performance by ignoring insignificant value changes (occurring within the specified deadband around the most recent value).

Attribute This is a select list that contains the attribute names.

Value This is an enterable list that contains the attribute values. Numbers that you enter here can scroll left and right off the field of view. If you get the error message **Invalid value type for this attribute**, use the left and right arrow keys to find the hidden numbers.

Network This list contains the network status of the attribute. **Yes** means that the attribute is networked, and **No** means that that attribute is not networked. **N/A** means that the attribute is either not networkable or that the control node is set to **OSx**.

Address Type Before you can select an address type, you must select a control node and set the Network field to **Yes**. Use the long-list display tool to the right of the field to select the address type of the attribute. Examples are shown in the Address Type column of Table 5-2.

Address After you select the address type, you can enter the address of the attribute in this field. This field may require one, two, or three values depending upon the type of control node memory. Examples are shown in the Address column of Table 5-2.

NOTE: S5 addressing is for a word number, which is the STEP 5 convention. S7 addressing is for a byte number, which is the STEP 7 convention. For example, in STEP 7, data block word/real values are referenced by a data block number and a byte offset in the data block.

Number of Locations Enter the number of locations. The number of locations is just an integer. This number can be more than one for an attribute that is used as a bit string or as a text string.

Upload This list contains the upload status of the attribute. The upload command is used to restore current values in the controller when you have reloaded or restarted the controller program. **Yes** means that the attribute is uploaded, and **No** means that the attribute is not uploaded. **N/A** means that the upload is not available. Select **Yes** for ranges and alarm limit values, which rarely or never change. For a Series 505 controller, any upload value is also one of the values downloaded to the controller when you select the **Tag_Download** command from Network Setup.

If you are not using tag alarms in your program, disable the alarm bits individually in APT by marking the box in the CFB, especially for loop and analog input tags. The alarm bits that are typically disabled are H_Dev, L_dev, ROC, and Bad_Xmtr (broken transmitter), but you should inspect each tag individually to determine the bits to disable. If you use broken transmitter alarms, be sure to maintain proper calibration of the signal so that the alarm truly represents a broken transmitter. If the signal is not properly calibrated, nuisance alarms can occur.

20% This list contains the 20% offset status of the attribute. **Yes** means that the attribute is set for 20% offset, and **No** means that the attribute is not set for 20% offset. **N/A** means that the 20% offset is not available. **Yes** is usually selected only for 4–20 ma signals assigned to word memory, such as 505 WX/WY. This setting is applicable for only a few OSx tag attributes, such as AI:PV and AO:OUT.

Autolog This list contains the autolog status of the attribute. **Yes** means that a message is printed whenever a changed value for the attribute is sent to the database. **No** means that the attribute is not set for autolog. **N/A** means that the autolog is not available. Use this feature to log attribute changes that have special significance. Avoid using autolog for attributes that change frequently.

Table 5-2 Example Address Types and Addresses for Controllers

Controller	Controller Memory Address	Memory Type	Address Type	Syntax	Address [1]
505	C101	Bit	C	Element	101
	V.22	Real	V.	Element	22
	K50	Word	K	Element	50
S5	F0.6	Bit	F	Byte.bit	0.6
	DB11:DW5	Word	DBDW	Block:Word	11:5
	DX12:DD7	Double Word	DXDD	Block:Word	12:7
S7	M7.6	Bit	M Bit	Byte.Bit	7.6
	DB11:DBX4.2		DB Bit	Block:Byte.Bit	11:4.2
	MW22	Word	M Word	Byte	22
	DB11:DBW2		DB Word	Block:Byte	11:2
	MD22	Double Word	M DWord	Byte	22
	DB11:DBD4		DBDWord	Block:Byte	11:4

1 A space may be used to separate the address values instead of a specific separator. Only the required number of values are processed; any extra values or input characters are ignored.

Adding a New Tag

To add a new tag to the system using the Tag Configurator, follow the steps below.

1. Select **Tags->Configurator** from the menu bar. The screen displays the Tag Configurator (Figure 5-4 on page 5-12).
2. Enter a unique name for the tag in the Tag field.
3. Configure the tag by filling out the fields in the Tag Configurator as described in the preceding pages.
4. Click the **Save** button. The new tag is added to the system.

Viewing a Tag

To view an existing tag using the Tag Configurator, follow the steps below.

1. Select **Tags->Configurator** from the menu bar. The screen displays the Tag Configurator (Figure 5-4 on page 5-12).
2. Enter the tag name or use the long-list display tool to the right of the field to select the tag that you want to view. The screen displays the current tag information in the fields of the Tag Configurator.

   If new data comes in from the controller, and you want to redisplay the tag information in the database, click the **Reload** button.

Modifying an Existing Tag

To modify an existing tag using the Tag Configurator, follow the steps below.

1. Select **Tags->Configurator** from the menu bar. The screen displays the Tag Configurator (Figure 5-4 on page 5-12).
2. Enter the name of the tag or use the long-list display tool to the right of the Tag field to select the tag that you want to modify.
3. Modify the tag by changing the information in the fields in the Tag Configurator as described in the preceding pages.

   If you make a mistake during modification, and you want to bring back the original data of the tag, click the **Reload** button.
4. Click the **Save** button. The tag modifications are saved.

Modifying H_Range, L_Range, and Deadband

To modify networked H_Range and L_Range attributes and Database Deadband while the system is in the Operate state, you must first set the Scan Status of the tag to **Inactive** on the Tag Detail display. If the tag is currently displayed in the Tag Configurator, you must select the **Reload** button in the Tag Configurator before making changes.

NOTE: For a Counter (CTR) tag, the Preset attribute is treated as the high range attribute. For an Area tag, the Scale_High and Scale_Low attributes are the high and low ranges respectively.

Assigning a Tag to a Process Group

To assign a new or existing tag using the Tag Configurator, follow the steps below.

1. Select **Tags->Configurator** from the menu bar. The system displays the Tag Configurator (Figure 5-4 on page 5-12).
2. Enter or select the name of the tag that you want to assign to a process group.
3. Click the **Process Group** button. The Process Group Configuration dialog box appears.
4. Select **Yes** for each process group that you want to associate with the tag; **No** for all others.
5. Select **OK** to confirm your choices and dismiss the dialog box, then click the **Save** button on the Tag Configurator to save your changes.

NOTE: A tag inherits process groups from its parent tag. If you specify a parent tag for a tag that you are installing, the process groups for the tag will be any process groups that you specify for this tag, plus all the process groups of the parent tag.

5.5 Configuring Tags with a Spreadsheet

You can configure tags using a spreadsheet program, such as Excel. The program must allow you to save your work in an ASCII file (in which commas separate the cell values). You must use either TISOFT SoftShop or APT to write the application program for Series 505 controllers. If you use APT, you can mark objects in the APT environment for use as tags. For S5 controllers, use the STEP 5 programming software or APT. Refer to the user documentation for your spreadsheet program for detailed information about entering data.

NOTE: When you configure tags by a method other than APT, you must place any UNIT tags that have other tags associated with them at the beginning of the **install.tag** file.

You can enter four types of data into the spreadsheet: Tag Data Types, Recipe Data Types, Process Group Data Types, and Delete Tag Data Types. These data types are described on the pages that follow. All the elements of each type are grouped together. A header line separates the data types. Each group of data consists of two or more lines of information. Table 5-7 at the end of this section shows an example of a spreadsheet containing all four types of data.

Tag Data Types

Tag data are made up of three types of information:

- The header line contains field names for the database and marks the beginning of tag information to be incorporated into the database. The header consists of 14 fields, defined in Table 5-3. These fields identify the data that is to be entered for subsequent lines in the tag data section.
- The tag identifier line begins with a T and defines the tag type. Enter values for only six fields identified in the header: Record, ControlNode, Tag Type, Tag Name, Tag Description, and Process Group.
- Depending on tag type, one or more attribute lines follow a tag identifier line. This line begins with an A. Values that you enter in the various fields depend on the type of tag.

Selecting Memory Types You must decide the type of controller memory appropriate for storing the values of specific attributes for each tag. The options for memory type selection vary by tag type, attribute, and associated controller type. See Appendix D, Controller Memory Types, for memory types available in the Series 505 and S5 controllers.

Table 5-3 Tag Header Fields

Field	Definition
Record	Identifies line of tag information. Enter `T` for a tag identifier, or `A` for a tag attribute line.
Control Node	Name of controller, e.g., 555 for a SIMATIC 555 controller, up to 12 characters.
Tag Type	Type of tag, such as LOOP, IVAR, DO, or DI. These are defined in Chapter 3.
Tag	Name of tag, up to 12 characters. All characters are valid except the space character, the character combinations -> and <- and the following characters: , ” \ ; The tag name can begin with an alphanumeric character (A–Z, 0–9) or the underscore (_) character. The tag name cannot be a numeric value, such as an integer, float, hex, or scientific notation number.
Description	Description of tag, up to 30 characters. The characters ” \ ; are illegal in this field. If a comma is included in the tag description, the whole string must be enclosed in double quotes; for example, `“Boiler Room A, Boiler 1.”`
Process Group	Process group number is indicated with a 32-bit hex number. Each bit corresponds to one of 32 process groups. For each process group for which the tag is to be associated, its corresponding bit is set to 1. For example, for a tag to be associated with process group 8, enter the following hex number in this field: `0x00000080`. The binary equivalent is `0000 0000 0000 0000 0000 0000 1000 0000`. For a tag to be associated with process groups 1, 2, and 3, enter this hex number: `0x00000007`. The binary equivalent is `0000 0000 0000 0000 0000 0000 0000 0111`. For a tag to be associated with all 32 process groups, enter `0xffffffff`. The binary equivalent is `1111 1111 1111 1111 1111 1111 1111 1111`. Always set unused process groups to 0. Otherwise, unintended memberships may occur between users, tags, alarm groups, and action requests.
Manual Set	Indicates Y(es) or N(o) whether you can enter data manually for the tag for trending purposes. Only applies to users of the HTU utility.
Parent	If you have structured your application program into units, tags within a unit can all be associated by means of a parent tag. Assign the same parent tag name to tags that pertain to the same unit. The name of a parent tag contains up to 12 characters.
Attribute	Tag attribute, such as Status, Command, or Reset. These are defined in Chapter 3. For custom attributes for tags used in graphics, do not use the character combinations -> and <- or the semicolon (;).
Memory	Memory location in the controller where tag attribute data is located, such as LPV1, V55, or WY41. Refer to Appendix D for more information about the memory types to choose for this field. For non-networked attributes, leave this field blank.
Locations	Number of memory locations required for attribute. This information varies with the tag attribute. See Chapter 3 for details about the number of memory locations required by each attribute. For non-networked attributes, leave this field blank.
Table is continued on next page.	

Table 5-3 Tag Header Fields (continued)

Field	Definition
Upload	Indicate Y(es) for ranges and alarm limit values, which rarely or never change, or for any other data value that is required for use in the control node Tag Download command. This command is used to restore current values in the controller when you have reloaded or restarted the controller program. Since upload values typically do not change, OSx stations collect them from the network when required, only on every other event scan period.
Twenty%	Indicates Y(es) or N(o) whether 20% offset is done on analog values. The 20% offset applies only for scaled 16-bit values, e.g., Series 505 WX memory. A span of 0 to 5.0 volts (0 to 20.0 milliamps) is referred to as a span of 0 to 100%. A span of 1.0 to 5.0 volts (4 to 20.0 milliamps) is referred to as a span of 20% to 100%.
Autolog	Indicates Y(es) or N(o) whether a log message is generated when this attribute value changes in the database. When you delete a tag that is configured for autologging, the system automatically deletes the autolog configuration.
InitValue	Initial value assigned to attribute.

Recipe Data Types

You can define most of the information needed in a recipe from an OSx station. You may prefer to do this rather than to add recipe data types to the spreadsheet. However, you can define the components for a recipe as described below. Note that recipe configuration requires an area tag, and the area tag must still be defined in the Tag Data Type section, whether or not you enter recipe components in the Recipe Data Type section.

Recipe data consist of three types of information:

- The header line marks the beginning of recipe information to be incorporated into the database. The header consists of five fields, defined in Table 5-4. These fields identify the data that is to be entered for subsequent lines in the recipe data section.
- The recipe identifier line defines the recipe and begins with an R. Enter values for only two fields identified in the header: Record and Recipe name.
- Depending on the number of recipe components in a recipe, one or more component lines follow a recipe identifier line. This line begins with a C. Enter values for these fields identified in the header: Record, Component, Tag, and Attribute.

Table 5-4 Recipe Header Fields

Field	Definition
Record	Identifies line of recipe information. Enter R for a recipe identifier line. Enter C for a recipe component line.
Recipe	Name of recipe, such as RCP_1 or RCP_2. Can consist of up to 12 characters.
Component	Name of recipe component, such as Sugar, Water, or Flour. Can consist of up to 12 characters. Components represent ingredients, processes, processing parameters, recipe controls in the recipe itself, and flags in control logic. Components are associated with all the tags and the tags' associated attributes in the recipe area.
Tag	Tag and attribute with which recipe component is associated. The tag and attribute work together to identify the controller memory location that is the destination of the downloaded associated recipe component value. The tag and attribute must match the data type being used, such as real number, integer, or boolean. For example, an IVAR tag cannot be used for real number information.
Attribute	

Process Group Data Types

Use this section of the spreadsheet to enter the descriptions for the process groups that you plan to define for your process. If you prefer, you can enter the process group descriptions later, as described in Chapter 2. You link tags, User IDs, and action requests to process groups by group number, not by the description.

Process Group data consist of two types of information:

- The header line marks the beginning of process group information that you want to incorporate into the database. The header consists of three fields, defined in Table 5-5. These fields identify the data you want to enter for subsequent lines in the process group data section.
- The process group identifier line defines the process group and begins with a P. Enter values for three fields identified in the header: Record, Group Number, and Description.

Table 5-5 Process Group Header Fields

Field	Definition
Record	Identifies line of process group information. Enter P for a process group identifier line.
Group Number	Process group number is indicated with a decimal number (1–32).
Description	String of text up to 30 characters that identifies the process group.

Deleted Tag Data Types

You can delete a tag that you incorporated into the database in a previous tag installation. Make an entry in the Deleted Tag Data Type section. This section consists of two types of information:

- The header line marks the beginning of deleted tag information to be incorporated into the database. The header consists of two fields, defined in Table 5-6. These fields identify the data that is to be entered for subsequent lines in the deleted tag data section.
- The deleted tag identifier line defines the tag to be deleted and begins with a `D` followed by a comma. Enter values for both fields identified in the header: Record and Tag.

You can delete tags individually from within the OSx environment. Refer to Section 5.9 for more information.

Table 5-6 Deleted Tag Header Fields

Field	Definition
Record	Identifies line of deleted tag information. Enter `D` followed by a comma for a deleted tag identifier line.
Tag	Name of tag to be deleted.

NOTE: If you have already installed a tag, and you decide to reinstall it leaving the InitValue field blank, the value in the database remains unchanged.

If you have already installed a tag, and you decide to reinstall it leaving the ManualSet, Upload, Twenty%, and/or Autolog field(s) blank, the system treats the blank as N for No and turns off that value if it was previously on.

Terminating Line

Enter an `E` in the record field of the last line in the spreadsheet to terminate the file. You must include the terminating line.

Table 5-7 shows shows an example of a spreadsheet containing all four types of data. The spreadsheet is 15 cells wide. The first eight cells are shown on page 5-24; the last seven cells are shown on page 5-25.

Table 5-7 Example Spreadsheet

Columns 1-8							
Record	ControlNode	TagType	Tag	Description	ProcessGroup	ManualSet	Parent
T	555	DO	R.INUSE		`0xffffffff`		
A							
A							
T	555	VLV1	VSS		`0xffffffff`		
A							
A							
A							
A							
A							
A							
Record	Recipe	Component	Tag	Attribute			
R	RCP1						
C		Milk	RCP1.R	VALUE			
C		Sugar	RCP.R	VALUE			
C		Timing	R.INUSE	COMMAND			
Record	GroupNumber	Description					
P	6	Group6					
P	5	Group5					
P	3	Group3					
Record	Tag						
D	Loop1						
E							
Table continued on next page.							

Table 5-7 Example Spreadsheet (continued)

Columns 9-15						
Attribute	Memory	Location	Upload	Twenty%	Autolog	InitValue
COMMAND	C29					
STATUS	C29					
TIMEOUT_2	TCP11					
TIMEOUT_1	TCP12					
SETPOINT						
MODE_COMMAND	C95					
STATUS	C90					
OVERRIDE	C98					

Filling Out the Generic Spreadsheet

Refer to the documentation for your spreadsheet program for detailed information about entering data. Follow these steps:

1. Select the spreadsheet parameters that allow you to save the file in an ASCII format, with the data items separated by commas (CSV format).

2. Enter the spreadsheet data, as described in Section 5.5.

3. Save your work in an ASCII file (in which commas separate the cell values). Name the file **install.tag**.

If you need help getting started creating the **install.tag** file, examine a template file. For Series 505 controllers, the file, called **install.tag**, is located in the **/usr/tistar/misc** directory. Access it by logging in from an OSx terminal window. For S5 controllers, the file is called **install.s5** and is located in the same directory. You can copy the file to a diskette and transfer it to an MS-DOS computer for editing with a spreadsheet program. Follow these steps:

1. Copy the template file on the system to diskette with the following command: `mcopy install.tag a:`

   If you are using Excel, use this command instead:
   `mcopy install.tag a:install.csv`

NOTE: To copy from a diskette or an MO, use the `mcopy -a` command instead.

---

2. After editing the file, copy it to a diskette. Name the edited file **install.tag**.

Now you are ready to install the file in the OSx database. See page 5-30.

Transferring Data to an OSx Station

After you have entered all the relevant information into the spreadsheet and saved the file, you need to incorporate the information into the database. You must have either the Database Admin or the System Config security privilege to install tags.

You can change or delete tags only in the Offline state. You can add new tags in either the Offline or the Operate state, but it is recommended that you set the system to the Offline state. The system does not collect data from the network as efficiently when you install tags in the Operate state. This is because the system does not treat the tags as RBE points. When an RBE point changes, each OSx station receives the same changed value. Non-RBE points are scanned by each OSx station individually, and thus each station receives its own value at the time of the scan. Install tags in the Operate state only when you have an immediate need to install tags and when you cannot set the system to the Offline state. See Section 5.2 for more information about tag installation and RBE processing.

You can only install one **install.tag** file at a time. If you start multiple installations from several stations, the system queues up the tasks and executes them sequentially.

You must configure a control node in the **Network Setup** option under **Startup** before you can install tags for that control node. You can transfer **install.tag** to an OSx station by diskette by following these steps:

1. Copy the spreadsheet file **install.tag** to a diskette.
2. Place the diskette in the drive of an OSx station.

Now you are ready to install the file in the OSx database. Use the procedure described on page 5-30 to transfer the tag data to an OSx station.

5.6 Configuring Tags with an ASCII Text Editor

You can configure tags using any ASCII text editor. You must use either TISOFT Softshop or APT to write the application program for Series 505 controllers. If you use APT, you can mark objects in the APT environment for use as tags. For S5 controllers, use the STEP 5 programming software or APT.

Refer to the documentation for your ASCII text editor program for detailed information about entering data.

Types of Data to Enter

Section 5.5 describes the types of information required. The format is shown in Figure 5-5. Be sure to separate all data items with commas.

If you need help getting started creating the **install.tag** file, you can examine a template file. For Series 505 controllers, the file, called **install.tag**, is located in the **/usr/tistar/misc** directory, and you can access it by logging in from an OSx terminal window. For S5 controllers, the file is called **install.s5** and is located in the same directory. You can edit the file using the vi or gvim text editor. You can also copy the file to a diskette and transfer it to a DOS computer for editing with a DOS ASCII editor. Follow these steps:

1. Copy the template file on the system to diskette with the following command: `mcopy install.tag a:`

NOTE: To copy from a diskette or an MO, use the `mcopy -a` command instead.

---

2. After editing the file, copy it to a diskette. Make sure that the name of the edited file is **install.tag.**

Now you are ready to install the file in the OSx database. Use the procedure described on page 5-30 to transfer the tag data to an OSx station.

```
Record,ControlNode,TagType,Tag,Description,ProcessGroup,ManualSet,Parent,Attribute,Memory,Locations,Upload,
Twenty%,Autolog,InitValue
T,545,DO,RCP1.INUSE,,0xffffffff
A,,,,,,,,COMMAND,C29,2,N, ,N,0
A,,,,,,,,STATUS,C29,2,V, ,N,0
T,545,VLV1,VSS,,0xffffffff
A,,,,,,,,TIMEOUT_2,TCP11,1,Y, N,10
A,,,,,,,,TIMEOUT_1,TCP12,1,Y, N,10
A,,,,,,,,SETPOINT,C97,1,Y, ,N,0
A,,,,,,,,MODE_CMD,C95,2,Y, ,N,0
A,,,,,,,,STATUS,C90,10,N, ,N,0
A,,,,,,,,OVERRIDE,C98,1,Y, N,0
T,545,LOOP,LOOP1,,0xffffffff
A,,,,,,,,CHANGE,,,,,,1
A,,,,,,,,STATUS,LSTATUS1,1,N, ,N,0
A,,,,,,,,MODE,LMODE1,1,N, ,N,0
A,,,,,,,,L_RANGE,LPVL1,1,Y, N,0
A,,,,,,,,H_RANGE,LPVH1,1,Y, ,N,100
A,,,,,,,,PV,LPV1,1,N, ,N,0
A,,,,,,,,SP,LSP1,1,N, ,N,O
A,,,,,,,,ROC_ALM,LRCA1,1,Y, ,N,1
A,,,,,,,,L_DEV,LYDA1,1,Y, ,N,0.1
A,,,,,,,,H_DEV,LODA1,1,Y, ,N,0.2
A,,,,,,,,LL_ALM,LLLA1,1,Y, ,N,0.1
A,,,,,,,,L_ALM,LLLA1,1,Y, ,N,0.2
A,,,,,,,,H_ALM,LHA1,1,Y, ,N,0.8
A,,,,,,,,HH_ALM,LHHA1,1,Y, ,N,0.9
A,,,,,,,,RATE,LTD1,1,Y, ,N,0
A,,,,,,,,RESET,LTI1,1,Y, ,N,999.9
A,,,,,,,,GAIN,LKC1,1,Y, ,N,1
A,,,,,,,,OUT,LMN1,1,N, ,N,0
A,,,,,,,,UNITS,,,,,,Liters
Record,Recipe,Component,Tag,Attribute
R,RCP1
C,,Milk,RCP1.R,VALUE
C,,Sugar,RCP1.R,VALUE
C,,Timing,RCP1.INUSE,COMMAND
Record,GroupNumber,Description
P,6,Group6
P,5,Group5
P,3,Group3
Record,Tag
D,Loop1
E
```

Line has wrapped.

Figure 5-5 Example ASCII File

5.7 Installing Tags from a Tag Data File

You can install tags from a tag data file stored on the hard disk, a diskette, or an MO disk. The tag data file may have originated from APT, from the conversion of a set of STEP 7 mapper files, from a spreadsheet, or from a manually edited text file.

To install tags from a tag data file, follow the steps below:

1. From the OSx main menu bar, select **Tags->Install to OSx->From Tag Data File**. The Tag Installation Operations dialog box appears (Figure 5-6).

2. If you are installing the tag data file from your hard drive, click the radio button to the left of **Hard Disk**.

   If you are installing the tag data file from an MO disk, click the radio button to the left of **MO Disk**.

   If you are installing the tag data file from a diskette, click the radio button to the left of **Diskette**.

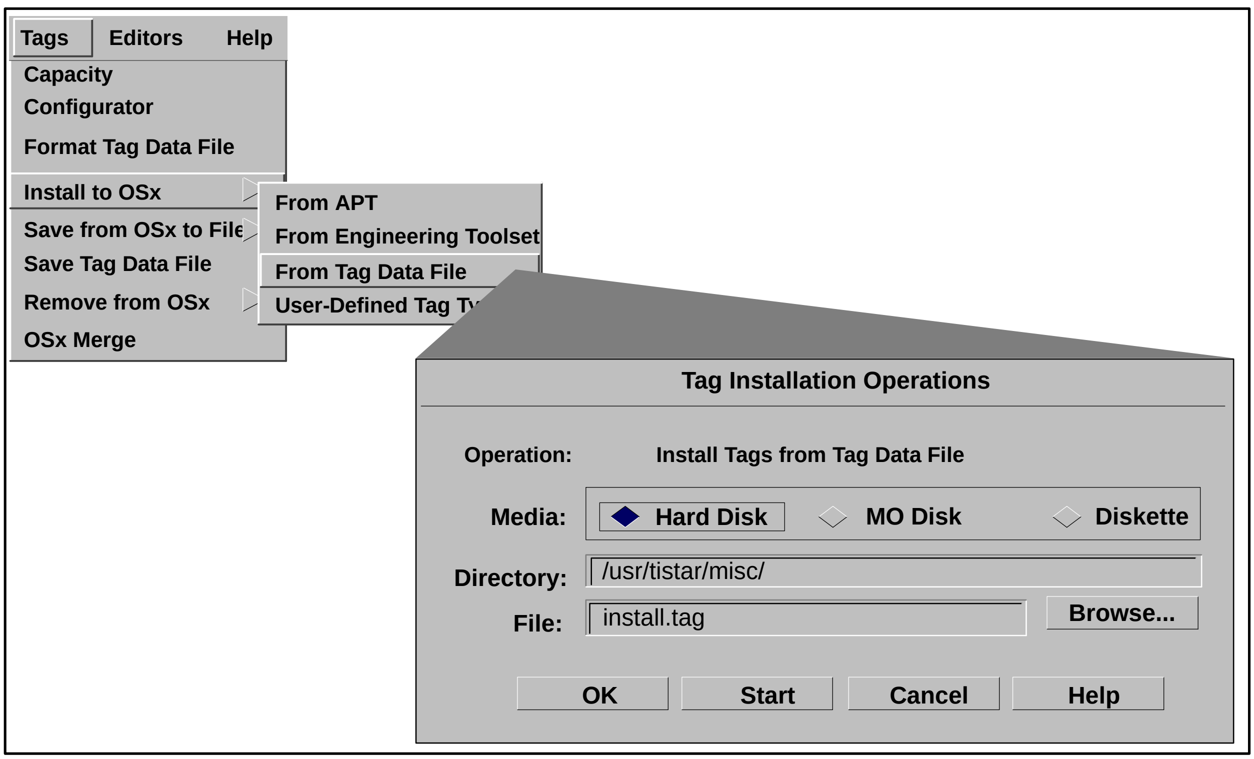


Figure 5-6 Installing Tags from Tag Data File

Installing Tags from a Tag Data File (continued)

3. Click **Browse** to display the File Selection list (Figure 5-7).

4. Use the File Selection list to specify the directory path where your tag data file is located.

5. Click **OK** to confirm the directory path. The directory path and file name appear in the Tag Installation Operations dialog box.

6. In the Tag Installation Operations dialog box, click the **Start** button to begin the tag installation. When installation is complete, the tag installation report is displayed.

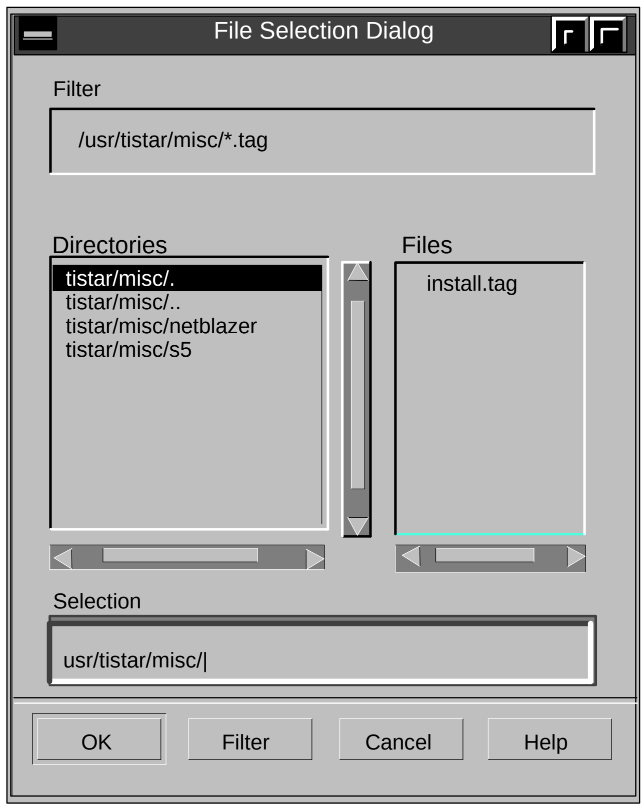


Figure 5-7 Selecting Directory Path for Tag Data File

5.8 Saving Tags

You can save tags to a removable media, such as a diskette or MO disk. You can specify the tags that you want to save in several different ways:

- **All Tags** — generates tag data files for all tags currently configured. One file is generated for each configured control node. All files are saved to removable media.
- **By Tag Name** — generates a tag data file for selected tags and saves the file to removable media. You can select the tags from a selection list of all tags. The default name for the file is **install.tag**.
- **By Tag Type** — generates a tag data file for selected tag type and saves the file to removable media. You can select the tag types from a selection list of all tag types. The default name for the file is **install.tag**.
- **By Control Node** — generates a tag data file for selected control nodes and saves the files to removable media. You can select the control nodes from a selection list. The default name for the file is **install.tag**.

Save Tag Data File — copies the specified tag data file from the hard disk to removable media (Figure 5-8). You can select the file from a selection list.

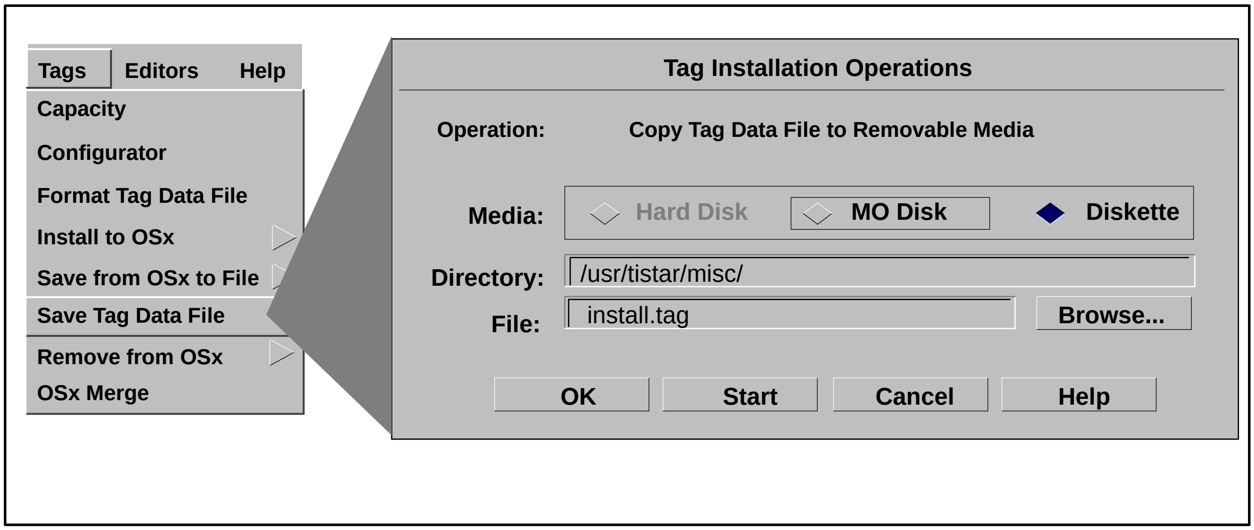


Figure 5-8 Copy Tags to Removable Media

To save tags to a removable media (diskette or MO disk), follow these steps:

1. From the OSx main menu bar, select **Tags->Save from OSx to File** and then choose an option from the Save menu. The Tag Installation Operations dialog box appears (Figure 5-9).

2. If you are saving tags to a diskette, click the radio button to the left of **Diskette**. Be sure that you have inserted a diskette in the floppy disk drive.

   If you are saving tags to an MO disk, click the radio button to the left of **MO Disk**. Be sure that you have inserted an MO disk in the MO drive.

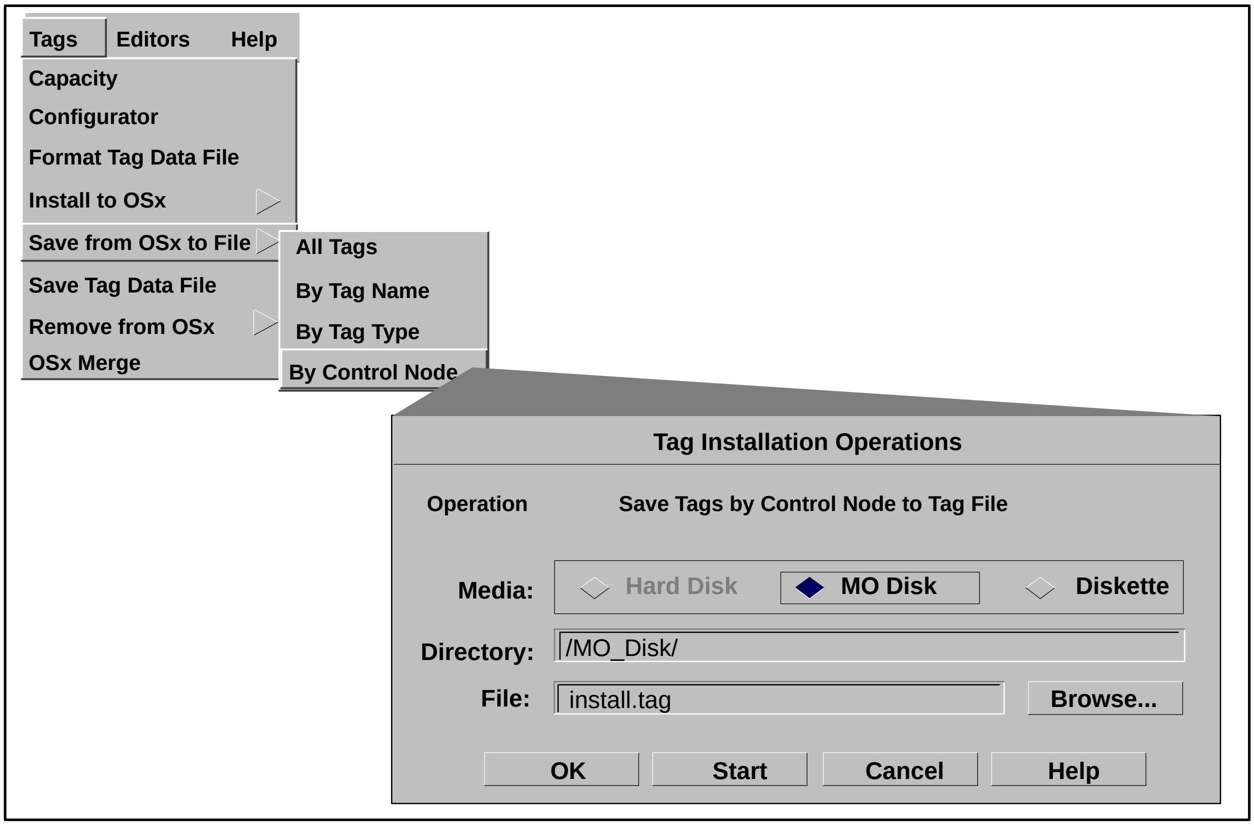


Figure 5-9 Saving Tags from OSx to Tag File

This procedure continues on the next page.

3. Click **Browse** to display a Selection List dialog box of tags, tag types, or control nodes, depending on the save option that you have selected. Figure 5-10 shows a selection list of tag types.

4. Click to highlight the items representing the tags that you want to save to removable media.

5. Click **OK** on the Selection List dialog box.

6. Click **OK** on the Tag Installation Operations dialog box to save the tags to removable media.

NOTE: The tag selection list displays OSx system tags in addition to other tags. (System tags begin with the underscore character; for example, _HT_UPDATE.) Although system tags can be selected from the list, they cannot be saved to a removable media and will be ignored in the Save by Tag Name operation.

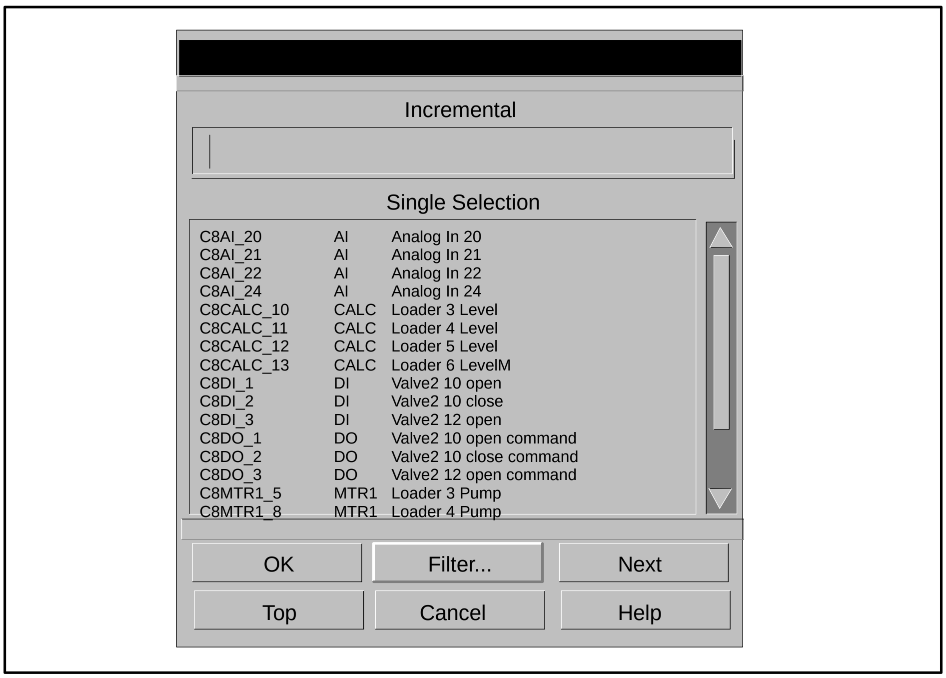


Figure 5-10 Selecting Tag Types to Save

5.9 Deleting Tags

Guidelines

You can delete individual tags from within the OSx environment. You must have either the Database Admin or the System Config security privilege to delete tags, and the system must be in the Offline state. The following rules also apply.

- The system deletes any alarm tags or alarm suppression tags that you specify. You cannot delete a tag that is an alarm group state tag. Doing so generates an error. You must first remove the tag from the alarm group in the Alarm Group Configuration dialog box. Then you can delete the tag from the system.
- When you delete a tag, the system automatically removes the tag from any trend or tag groups.
- When you delete a tag that is configured for autologging, the system automatically deletes the autolog configuration.
- When you delete a tag that is a component of a recipe area, the system automatically removes the component from the recipe area. However, attempting to delete a tag under the following conditions generates an error:

  The tag is a component of a recipe area that currently has a production recipe. To delete the tag, you must first remove its recipe from production. See the *SIMATIC PCS 7 OSx Recipe Manual* for the procedure.

  The tag is a recipe area tag that currently has a production recipe. To delete the tag, you must first remove the recipe in that area from production. See the *SIMATIC PCS 7 OSx Recipe Manual* for the procedure.
- You cannot delete a tag that is configured in an action request. This includes trigger, reset, answer, or embedded tags. If you attempt to delete a tag that is in an action request, the system generates an error. You must first either delete the action request or replace the tag in the action request with another tag. In the case of a nonessential embedded tag, you can simply remove the tag from the action request. Then you can delete the tag from the system.
- If the tag to be deleted is used in a report, graphic, BCL, or RDT program, you must remove the tag from these items manually. Then you must recompile (for programs), revalidate, and reinstall the item.

You can also make entries in the **install.tag** file to delete tags. Refer to page 5-23 for more information.

Deleting with the Tag Configurator

Within OSx, you can delete one or more individual tags either with the Tag Configurator (below) or with the Tag Installation dialog box (page 5-38). To delete individual tags using the Tag Configurator, follow these steps.

1. Set the system to the Offline state.
2. Select **Tags–>Configurator** from the menu bar.
   The screen displays the Tag Configurator (Figure 5-11).
3. Enter the name of the tag or use the long-list display tool to the right of the Tag field to select the tag you want to delete.
4. Click the **Delete** button, and click **Yes** when the Confirm Delete dialog box appears. The tag is deleted from the system.

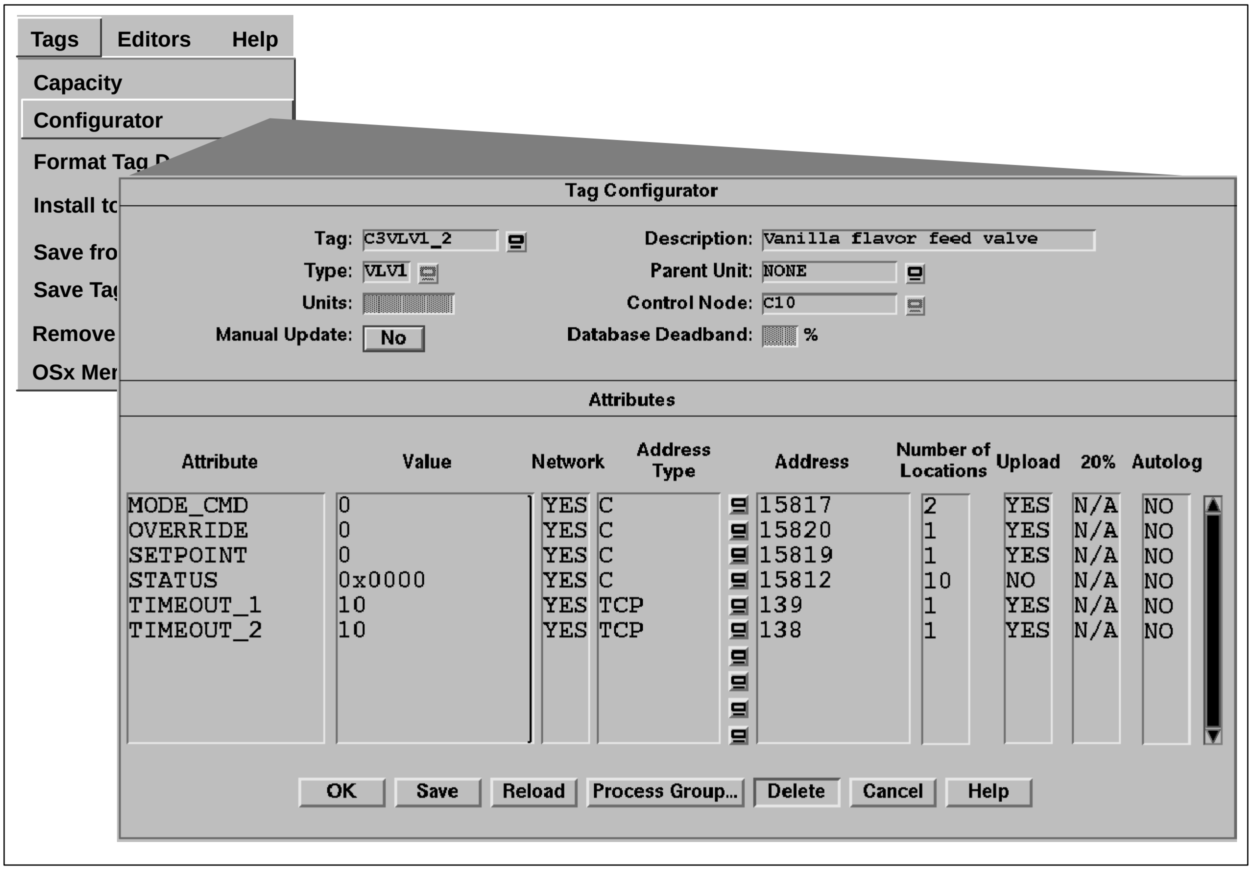


Figure 5-11 Deleting Tags with the Tag Configurator

Deleting Tags Using the Tag Installation Dialog Box

You can remove (delete) tags from your OSx system in two ways from the Tag Installation dialog box:

- **By Tag Name** — deletes tags from a selection list of all tags.
- **From Tag Data File** — deletes tags specified in a tag data file stored on the hard disk, a diskette, or an MO disk. See page 5-23 for information on how to create a tag data file for deleting tags.

To delete tags from your OSx system, follow these steps:

1. Set the system to the Offline state.
2. From the OSx main menu bar, select **Tags–>Remove from OSx** and then choose an option from the Remove menu. The Tag Installation Operations dialog box appears (Figure 5-12).
3. If you are removing tags specified in a tag data file on a diskette, click the button to the left of **Diskette**. Be sure that you have inserted the diskette in the floppy disk drive.

   If you are removing tags specified in a tag data file on an MO disk, click the button to the left of **MO Disk**. Be sure that you have inserted the MO disk in the MO drive.

   If you are removing tags specified in a tag data file on your hard drive, click the button to the left of **Hard Disk**.
4. Click **Browse** to display a Selection List dialog box for the tags or the tag data file of tags that you want to remove (Figure 5-13). The Selection List times out after five minutes. If you are removing a large number of tags, consider using a tag data file instead of selecting the tags individually.
5. Click to highlight the items representing the tags that you want to remove.

To delete a user-defined tag type from your OSx system, see the *SIMATIC PCS 7 OSx Interface to S7 Controllers Manual*.

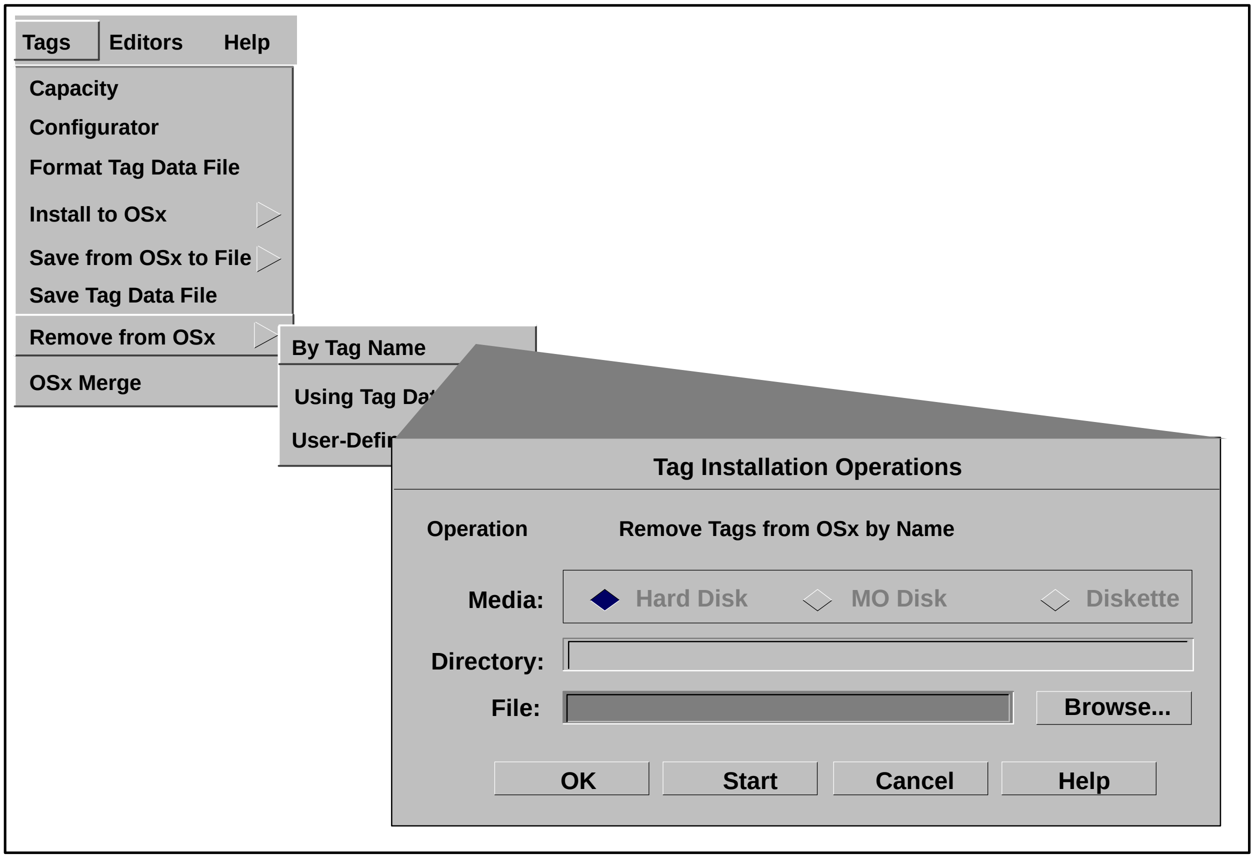


Figure 5-12 Removing Tags from OSx

This procedure continues on the next page.

Deleting Tags (continued)

6. Click **OK** on the Selection List dialog box to confirm your choices.

7. Click **OK** on the Tag Installation Operations dialog box to remove the tags. When installation is complete, the tag installation report is displayed.

NOTE: The tag selection list displays OSx system tags in addition to other tags. (System tags begin with the underscore character; for example, _HT_UPDATE.) Although system tags can be selected from the list, they cannot be deleted and will be ignored in the Remove by Tag Name operation.

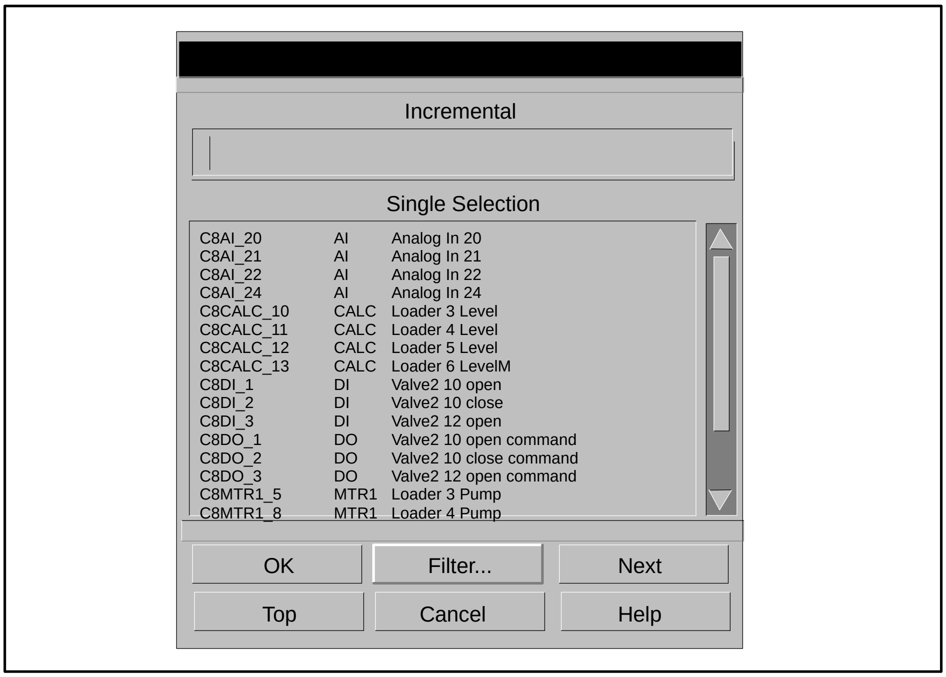


Figure 5-13 Selecting Tags to Delete

Chapter 6

Viewing Tag Information

6.1 Tag Detail Displays

SIMATIC PCS 7 OSx provides a standard display format, called the tag detail, for all the tag types except the recipe area tags and the tags that represent the system nodes. In the Operate state, the tag detail shows the attribute information associated with a tag. The operator can display a tag detail and observe the current values for the tag's attributes.

NOTE: A point on a tag detail may change to magenta, and the animation may stop, under the following circumstances:

- The OSx station loses communication with the control node.
- You deactivate the scan for a tag from a tag detail display.
- A tag contains no points that are configured for network scan.
- The controller returns invalid floating point data for tag types with floating point attributes.
- You have assigned the data to an invalid controller memory location and OSx cannot collect the data.
- You have sent the **Scan_Off** command to the OSx station.

Displaying a Tag Detail

A Tag Detail Directory lists the configured tags (except the recipe area tags and the tags that represent the system nodes) and their corresponding descriptions. The directory lists one tag detail name on each line.

To access the directory of tag detail displays, follow these steps:

1. Set the system state to Operate. The screen displays the navigation area.
2. Select the **Directory** pushbutton from the navigation area. The screen displays a directory of OSx display types (Figure 6-1, middle).
3. Select the **Tag Detail** option. The screen displays a list of tags (Figure 6-1).
4. Select the tag that you need to see. The screen displays the detail for the tag.

The figures on the following pages illustrate examples of tag detail displays that the operator can view and change. Since the security privileges required can vary, they are listed individually in tables accompanying each tag detail.

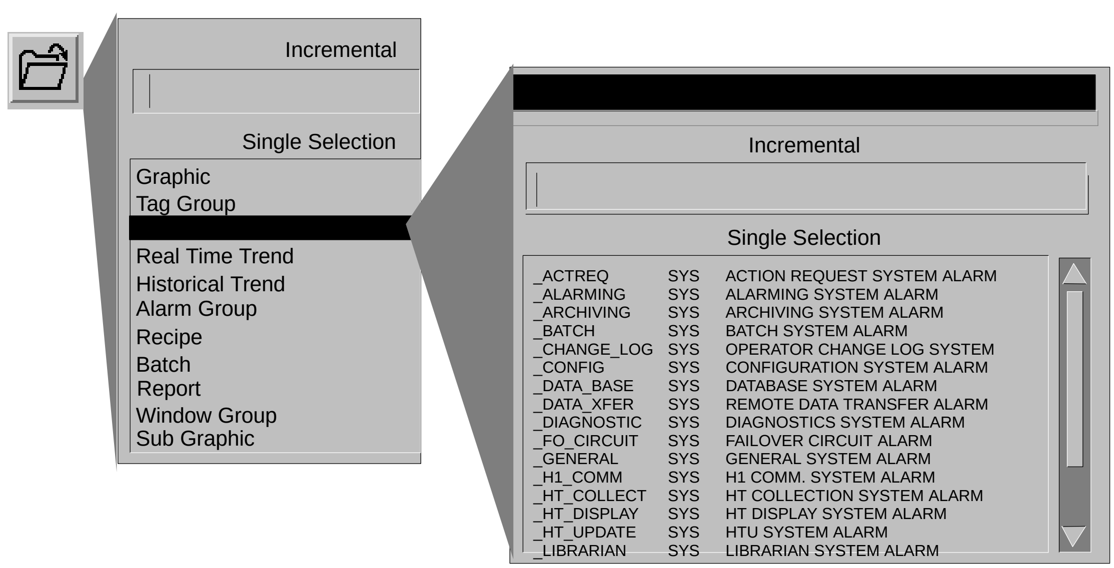


Figure 6-1 Tag Detail Directory

Changing Tag Detail Values in the Operate State

While the OSx system is in the Operate state, the operator may need to interact with the controller (for example, to change a setpoint, adjust alarm limits, enable or disable alarms, or start or stop a pump or motor). For further information on enabling/disabling alarms from a tag detail, see Chapter 9, Guidelines for Configuring Alarms. An operator who has the appropriate security privileges and process group membership can do this by entering new values in a tag detail. Figure 6-2 shows a typical tag detail display.

On the tag detail, the shadowed or indented boxes indicate values that you can change. Click the box to bring up a command form at the bottom of the screen (Figure 6-3).

NOTE: If you display a command form for one of the tag details, make a change to a field, and the change does not succeed, the value in the database does not change. However, the command form does display the incorrect value. The tag detail continues to display the correct, unchanged value. When you redisplay the command form, it contains the original, unchanged value.

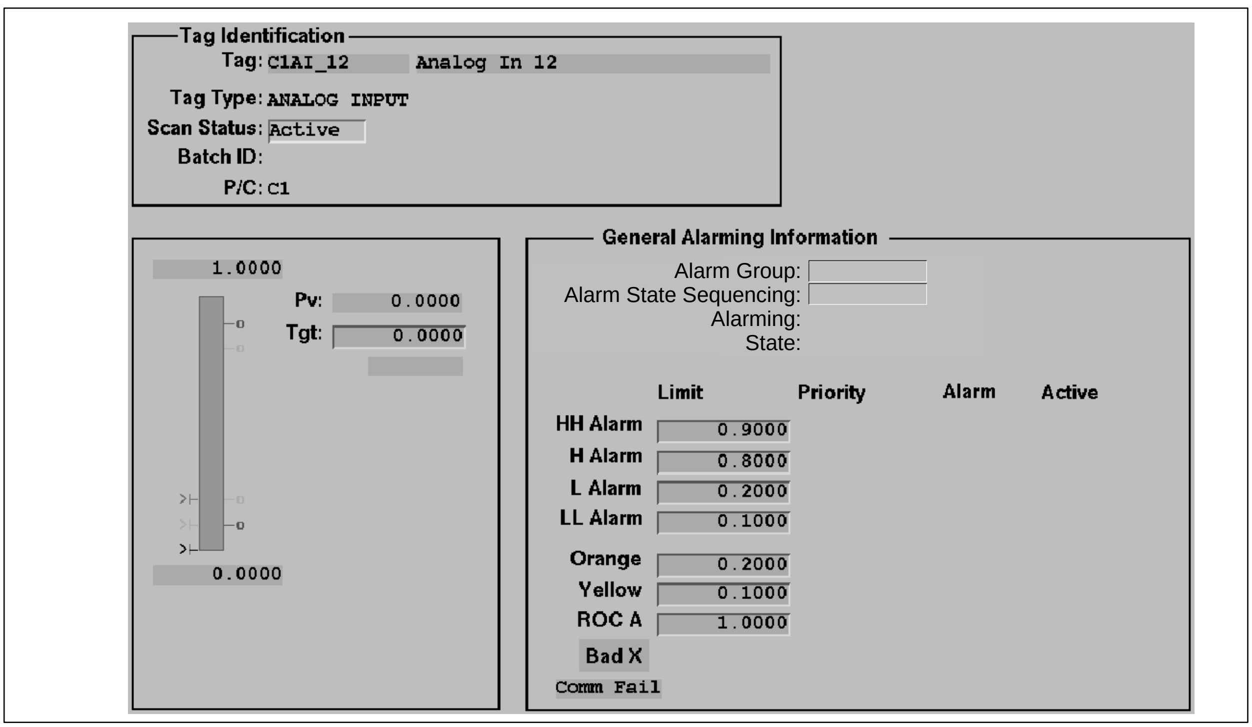


Figure 6-2 Tag Detail Display

Figure 6-3 shows three kinds of command forms that an operator sees. In Example A, the operator types a new value into the box on the command form and presses **Enter**. The new value appears automatically on the tag detail. The operator can change the value as often as necessary before closing the command form by selecting the **Dismiss** pushbutton.

Example B shows a command form with several fields. The operator can change values for more than one entry, pressing **Enter** after each new value.

In Example C, the operator changes certain variables, not by entering a new value, but by choosing among options. The operator selects the pushbutton to the right of the field on the command form that contains the variable. Then the operator selects a new option from the pop-up list that appears.

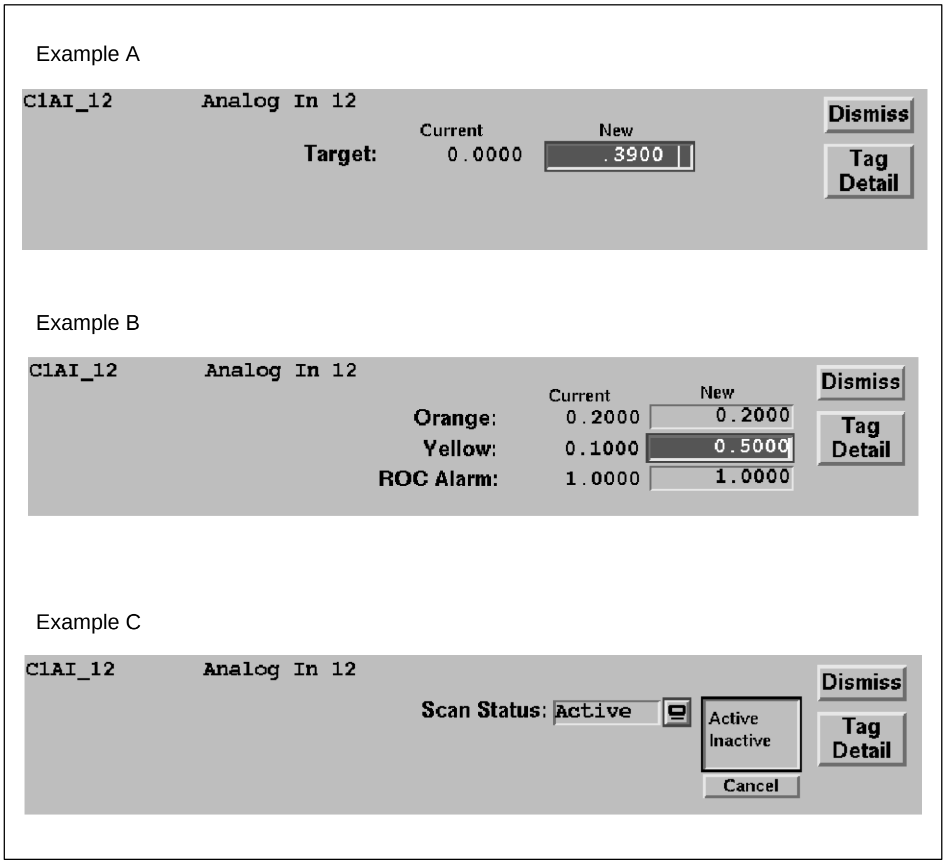


Figure 6-3 Tag Detail Command Form

Entering a Remark

In some cases, federal regulations or company policy may require operators to enter a remark each time they change a value in the controller from a command form. You can enable the OSx Remark feature to ensure operator compliance with this requirement.

When the Remark feature is enabled, and an operator changes a value on a command form and dismisses that form, a new command form appears (Figure 6-4). The Tag and User fields are read-only and are already filled. The operator simply types a comment in the Remark field, explaining why the value was changed, and clicks **OK**. If no remark is entered, the operator is prompted to enter one. The form cannot be dismissed, and no other tag detail or graphic can be viewed until a remark is entered.

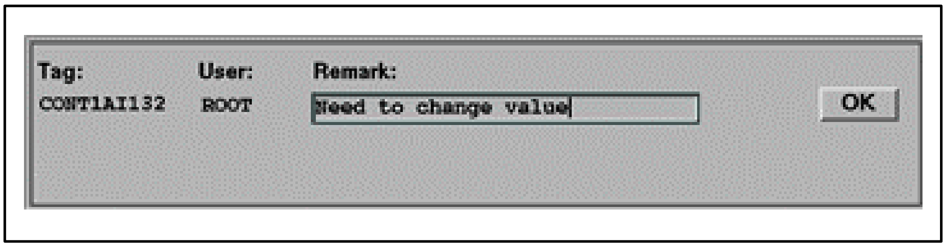


Figure 6-4 Remark Form

This remark is then listed on a separate line of the Operator Change Log, along with the associated tag and the operator's user ID. On a very busy system, the remark may be separated by some lines from the value change that it explains, but the two can be matched by tag name.

To enable the Remark feature on an OSx station, follow these steps:

1. Open an XTerm window, and log in as `tistar`.
2. At the prompt, type the following command:

   ```
   remark.sh on
   ```

   and press **Enter**. The Remark feature is now enabled.

To disable the Remark feature, repeat the steps above, but enter the following command: `remark.sh off`

Analog Input

Figure 6-5 shows a typical detail display for an analog input tag.

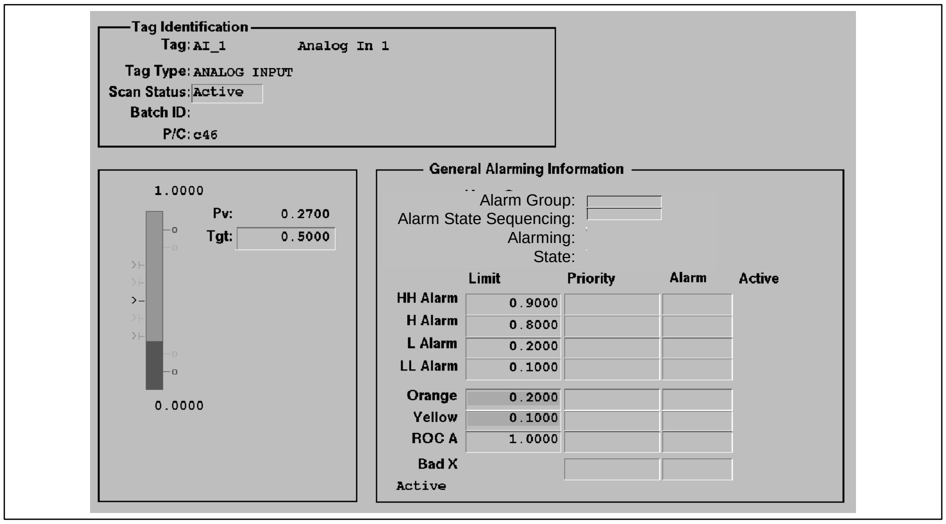


Figure 6-5 Analog Input Tag Detail

Table 6-1 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in an analog input tag detail.

Table 6-1 Security Privileges for Analog Input Tag Details

Attributes	Security Privileges				
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control
Target	✓	✓	✓	✓	✓
HH_Alarm	✓	✓			✓
H_Alarm	✓	✓			✓
L_Alarm	✓	✓			✓
LL_Alarm	✓	✓			✓
Orange_Dev	✓	✓			✓
Yellow_Dev	✓	✓			✓
ROC_Alarm	✓	✓			✓
Scan Status	✓	✓			✓
Alarm Priority	✓	✓			✓
Enable/Disable Alarms	✓	✓			✓

Analog Output

Figure 6-6 shows a typical detail display for an analog output tag.

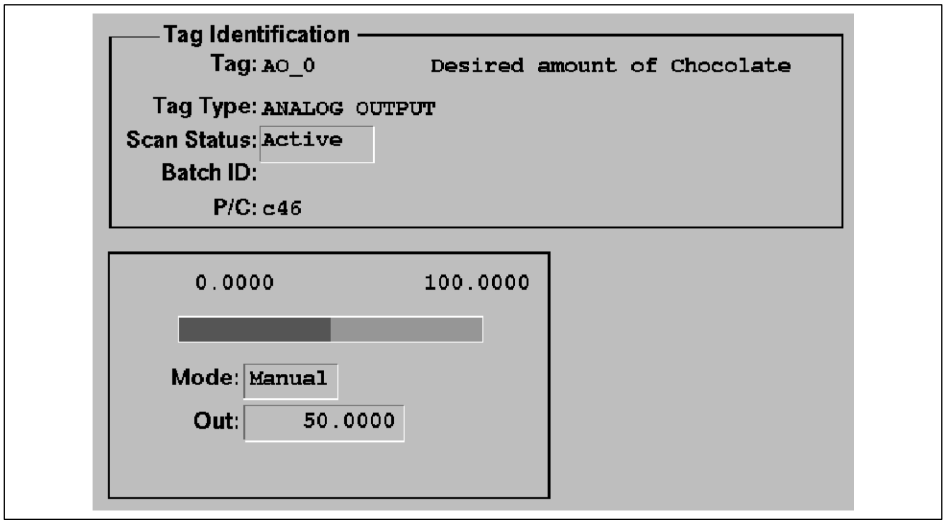


Figure 6-6 Analog Output Tag Detail

Table 6-2 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in an analog output tag detail.

Table 6-2 Security Privileges for Analog Output Tag Details

Attributes	Security Privileges				
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control
Output	✓	✓	✓	✓	✓
Mode	✓	✓	✓	✓	✓
Scan Status	✓	✓			✓

Loop

Figure 6-7 shows a typical detail display for a loop tag.

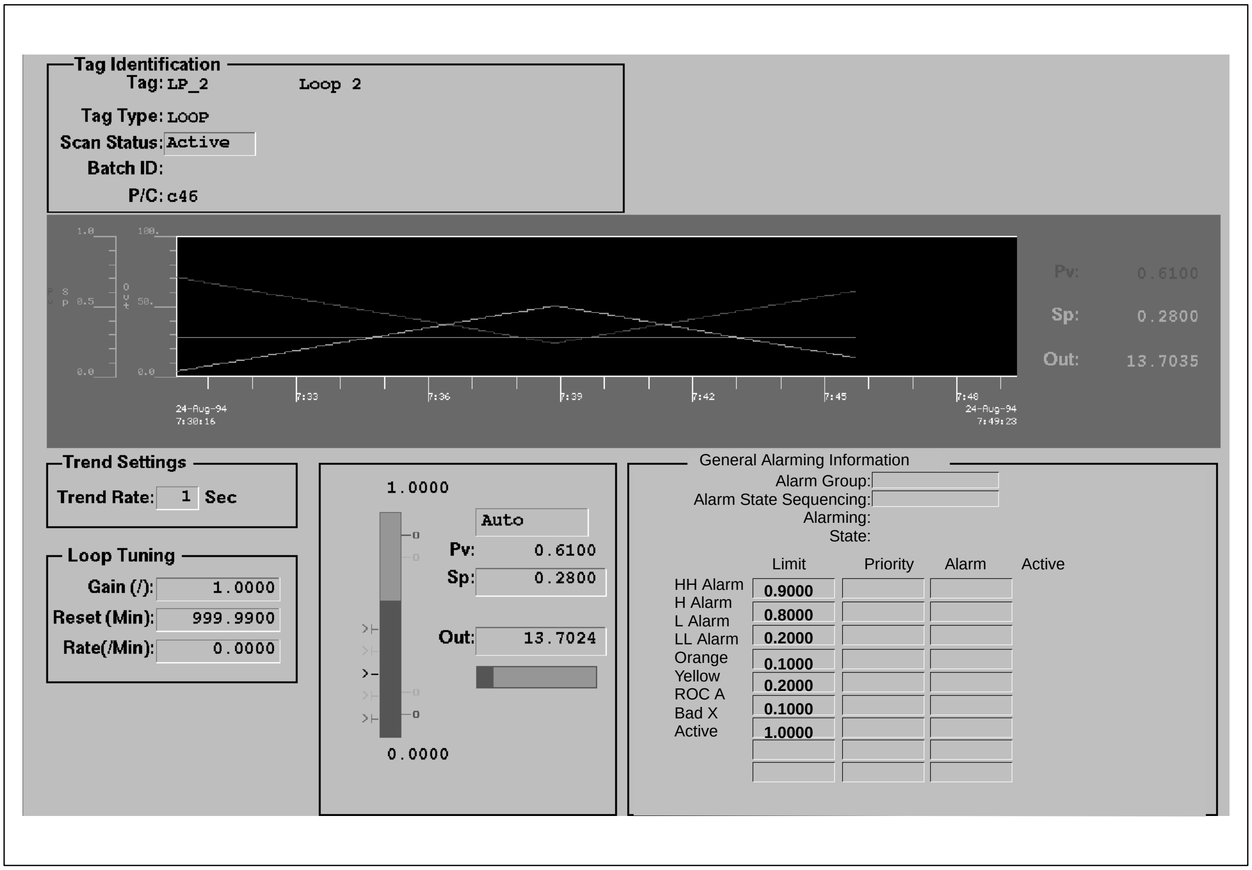


Figure 6-7 Loop Tag Detail

Table 6-3 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in a loop tag detail.

Table 6-3 Security Privileges for Loop Tag Details

Attributes	Security Privileges					
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control	Control Tuning
Setpoint	✓	✓	✓	✓	✓	
Output	✓	✓	✓	✓	✓	
Mode	✓	✓	✓	✓	✓	
Plot_Rate	✓	✓		✓	✓	
HH_Alarm	✓	✓			✓	
H_Alarm	✓	✓			✓	
L_Alarm	✓	✓			✓	
LL_Alarm	✓	✓			✓	
Orange_Dev	✓	✓			✓	
Yellow_Dev	✓	✓			✓	
ROC_Alarm	✓	✓			✓	
Gain	✓	✓				✓
Rate	✓	✓				✓
Reset	✓	✓				✓
Scan Status	✓	✓			✓	
Alarm Priority	✓	✓			✓	
Enable/Disable Alarms	✓	✓			✓	

Digital Input

Figure 6-8 shows a typical detail display for a digital input tag.

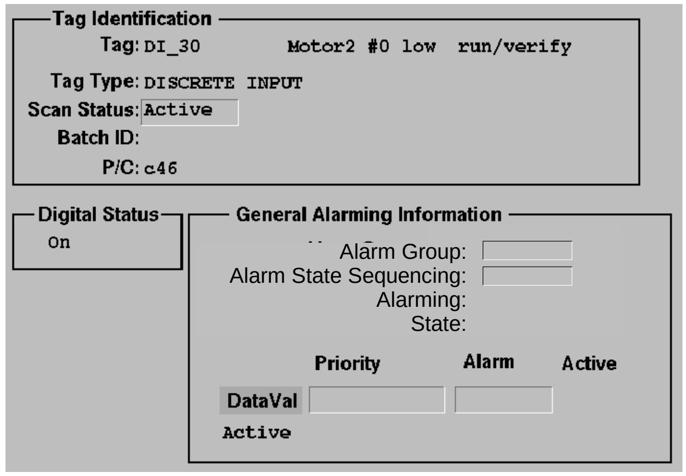


Figure 6-8 Digital Input Tag Detail

Table 6-4 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in a digital input tag detail.

Table 6-4 Security Privileges for Digital Input Tag Details

Attributes	Security Privileges				
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control
Scan Status	✓	✓			✓
Alarm Priority	✓	✓			✓
Enable/Disable Alarms	✓	✓			✓

Digital Output

Figure 6-9 shows a typical detail display for a digital output tag.

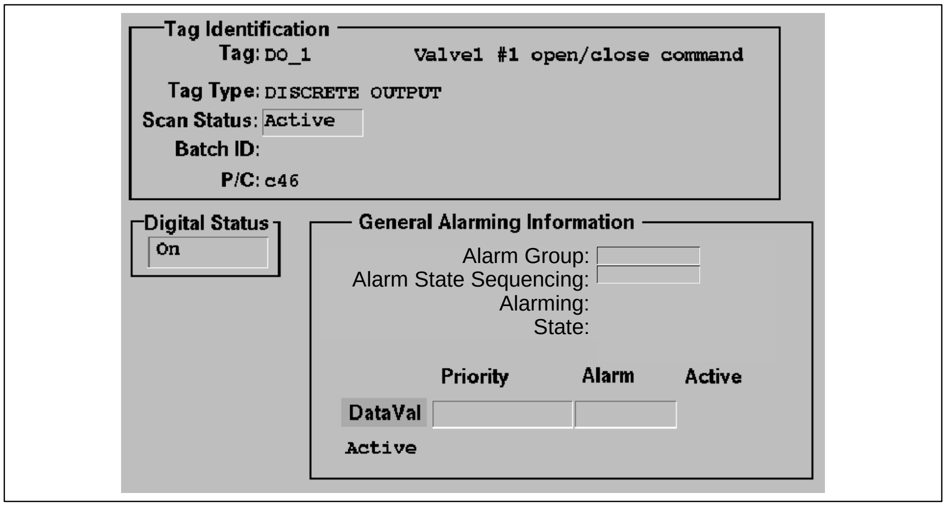


Figure 6-9 Digital Output Tag Detail

Table 6-5 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in a digital output tag detail.

Table 6-5 Security Privileges for Digital Output Tag Details

Attributes	Security Privileges				
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control
Command	✓	✓	✓	✓	✓
Scan Status	✓	✓			✓
Alarm Priority	✓	✓			✓
Enable/Disable Alarms	✓	✓			✓

10-Bit Digital Input

Figure 6-10 shows a typical detail display for a 10-bit digital input tag.

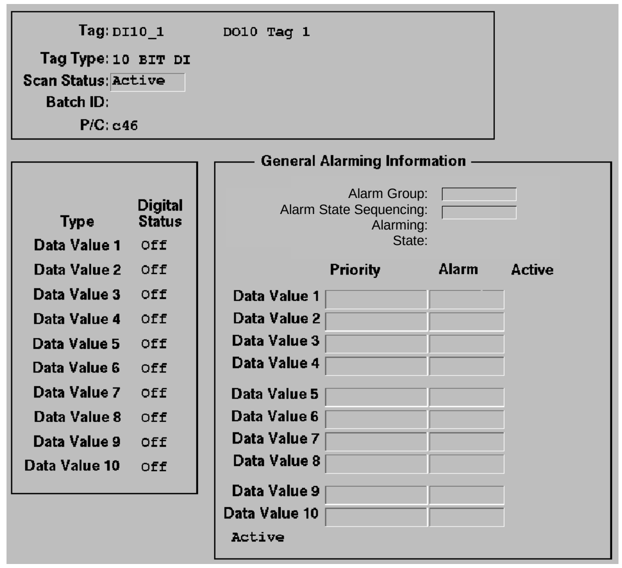


Figure 6-10 10-Bit Digital Input Tag Detail

Table 6-6 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in a 10-bit digital input tag detail.

Table 6-6 Security Privileges for 10-Bit Digital Input Tag Details

Attributes	Security Privileges				
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control
Scan Status	✓	✓			✓
Alarm Priority	✓	✓			✓
Enable/Disable Alarms	✓	✓			✓

10-Bit Digital Output

Figure 6-11 shows a typical detail display for a 10-bit digital output tag.

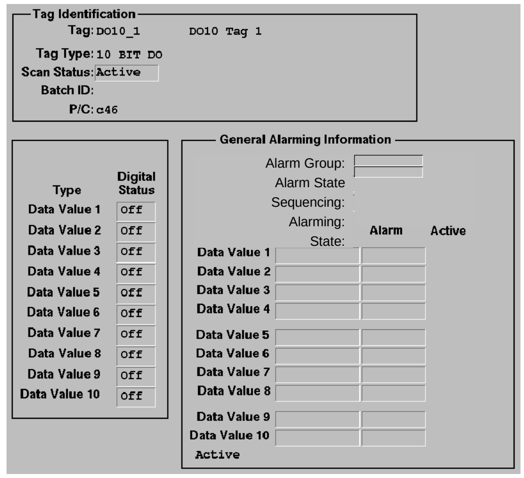


Figure 6-11 10-Bit Digital Output Tag Detail

Table 6-7 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in a 10-bit digital output tag detail.

Table 6-7 Security Privileges for 10-Bit Digital Output Tag Details

Attributes	Security Privileges				
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control
Data_Val1	✓	✓	✓	✓	✓
Data_Val2	✓	✓	✓	✓	✓
Data_Val3	✓	✓	✓	✓	✓
Data_Val4	✓	✓	✓	✓	✓
Data_Val5	✓	✓	✓	✓	✓
Data_Val6	✓	✓	✓	✓	✓
Data_Val7	✓	✓	✓	✓	✓
Data_Val8	✓	✓	✓	✓	✓
Data_Val9	✓	✓	✓	✓	✓
Data_Val10	✓	✓	✓	✓	✓
Scan Status	✓	✓			✓
Alarm Priority	✓	✓			✓
Enable/Disable Alarms	✓	✓			✓

Calculated Variable

Figure 6-12 shows a typical detail display for a calculated variable tag.

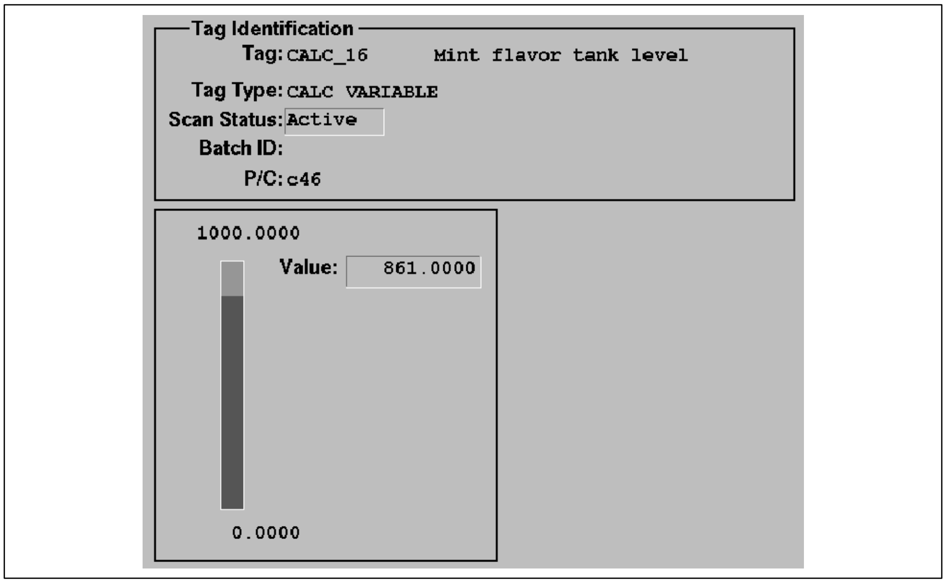


Figure 6-12 Calculated Variable Tag Detail

Table 6-8 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in a calculated variable tag detail.

Table 6-8 Security Privileges for Calculated Variable Tag Details

Attributes	Security Privileges				
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control
Value	✓	✓	✓	✓	✓
Scan Status	✓	✓	✓		✓

Integer Variable

Figure 6-13 shows a typical detail display for an integer variable tag.

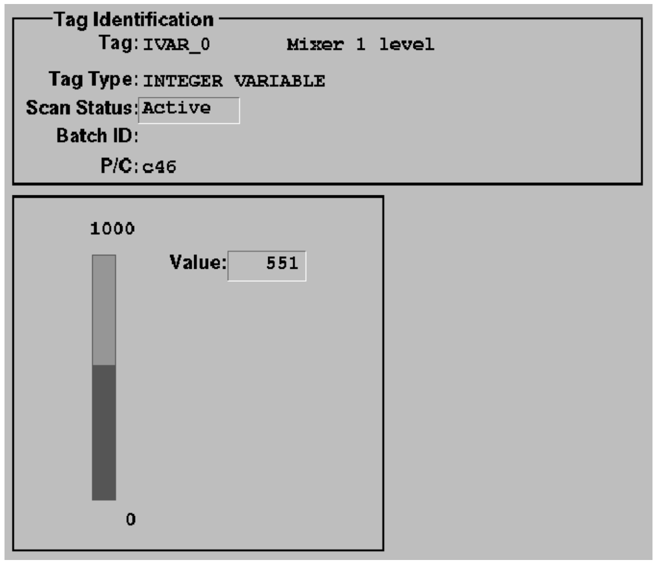


Figure 6-13 Integer Variable Tag Detail

Table 6-9 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in an integer variable tag detail.

Table 6-9 Security Privileges for Integer Variable Tag Details

Attributes	Security Privileges				
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control
Value	✓	✓	✓	✓	✓
Scan Status	✓	✓	✓		✓

Timer

Figure 6-14 shows a typical detail display for a timer tag.

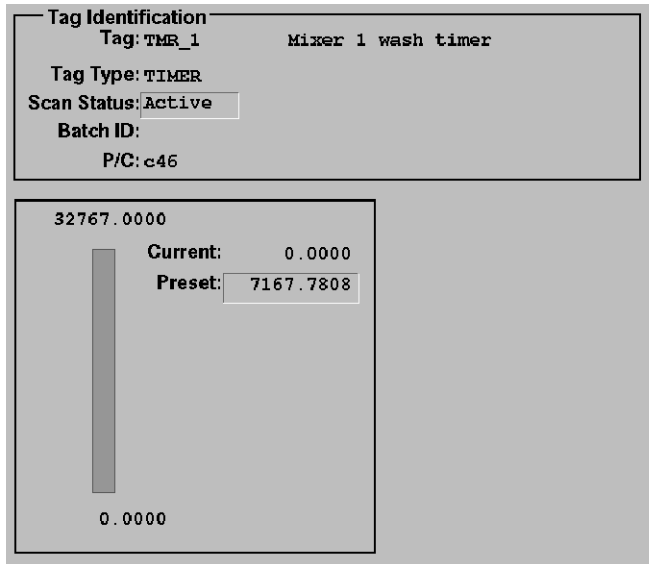


Figure 6-14 Timer Tag Detail

Table 6-10 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in a timer tag detail.

Table 6-10 Security Privileges for Timer Tag Details

Attributes	Security Privileges				
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control
Preset	✓	✓	✓	✓	✓
Scan Status	✓	✓			✓

Counter

Figure 6-15 shows a typical detail display for a counter tag.

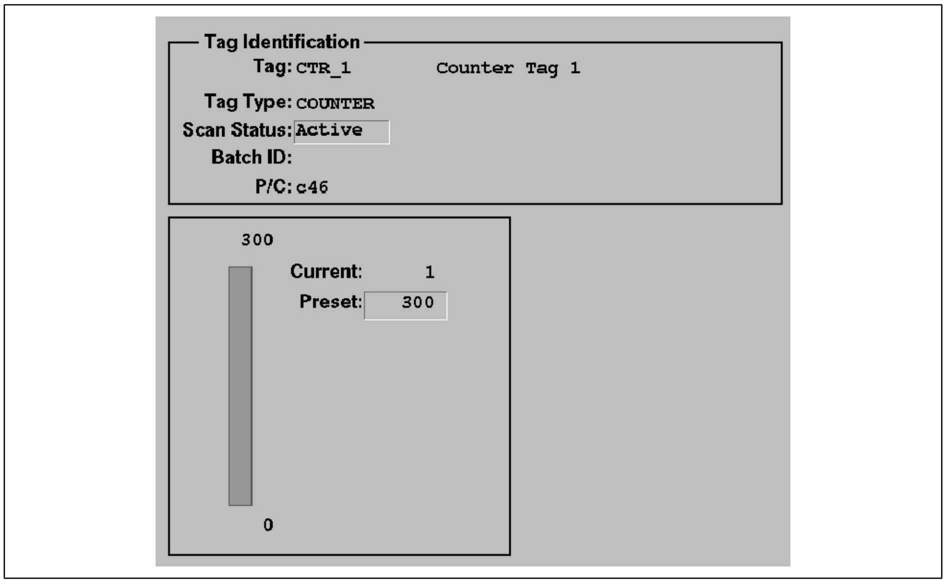


Figure 6-15 Counter Tag Detail

Table 6-11 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in a counter tag detail.

Table 6-11 Security Privileges for Counter Tag Details

Attributes	Security Privileges				
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control
Preset	✓	✓	✓	✓	✓
Scan Status	✓	✓			✓

Device

You can configure five types of devices in OSx: single-feedback valves (VLV1), dual-feedback valves (VLV2), motors (MTR1), reversible motors (RMTR), and two-speed motors (MTR2). Figure 6-16 shows a typical detail display for a reversible motor tag.

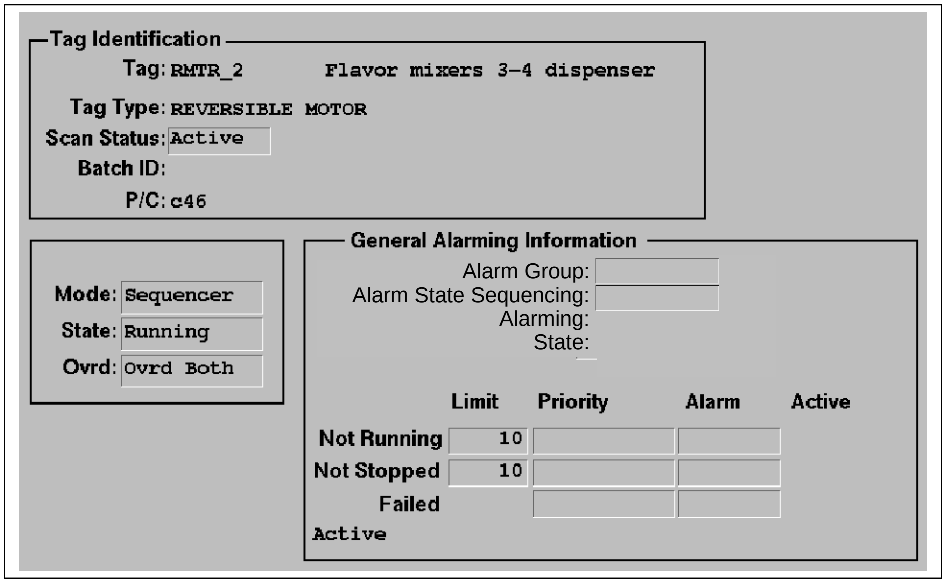


Figure 6-16 RMTR Tag Detail

Table 6-12 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in a device tag detail.

Table 6-12 Security Privileges for Device Tag Details

Attributes	Security Privileges				
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control
Mode_Cmd	✓	✓	✓	✓	✓
Setpoint	✓	✓	✓	✓	✓
Override	✓	✓		✓	✓
Timeout_1	✓	✓			✓
Timeout_2	✓	✓			✓
Scan Status	✓	✓			✓
Alarm Priority	✓	✓			✓
Enable/Disable Alarms	✓	✓			✓

System

OSx provides preconfigured system tags, identified by underscore prefixes (for example, _DATA_FILSYS). These system tags generate alarm messages when an error occurs in the system. OSx assigns each of its error codes to a particular system tag, so that you can identify the area of the system where the error has occurred. Refer to Chapter 3 for a complete list of all the system tags. Figure 6-17 shows a typical detail display for a system tag.

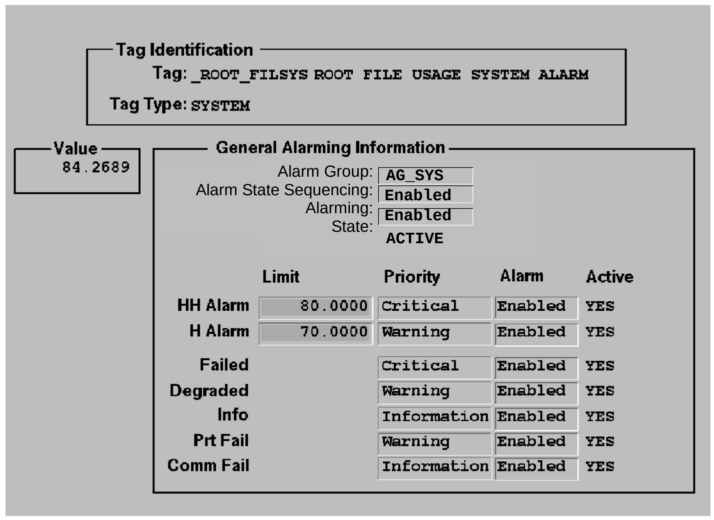


Figure 6-17 System Tag Detail

Table 6-13 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in a system tag detail.

Table 6-13 Security Privileges for System Tag Details

Attributes	Security Privileges				
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control
HH_Alarm	✓				
H_Alarm	✓				
Alarm Priority	✓	✓			✓
Enable/Disable Alarms	✓	✓			✓

Text

Figure 6-18 shows a typical detail display for a text tag.

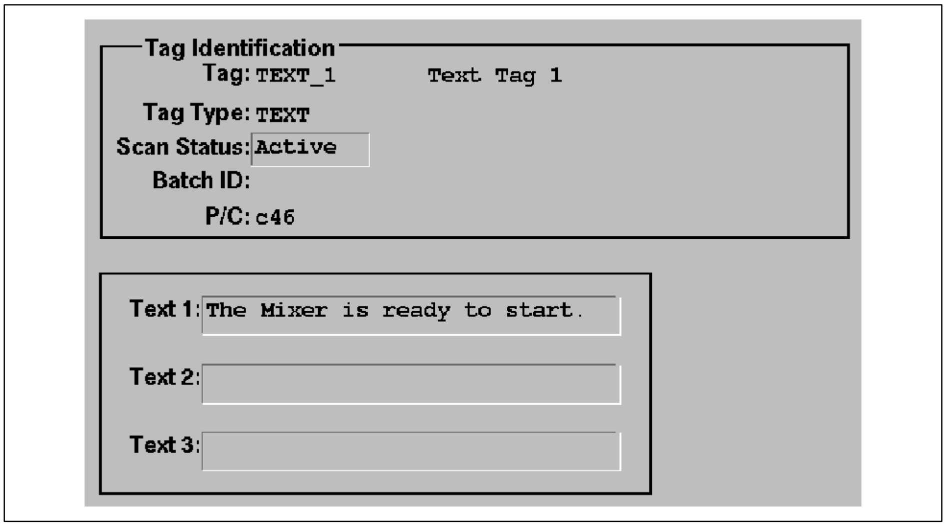


Figure 6-18 Text Tag Detail

Table 6-14 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in a text tag detail.

Table 6-14 Security Privileges for Text Tag Details

Attributes	Security Privileges				
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control
Text_1	✓	✓	✓	✓	✓
Text_2	✓	✓	✓	✓	✓
Text_3	✓	✓	✓	✓	✓
Scan status	✓	✓			✓

Unit

Figure 6-19 shows a typical detail display for a unit tag.

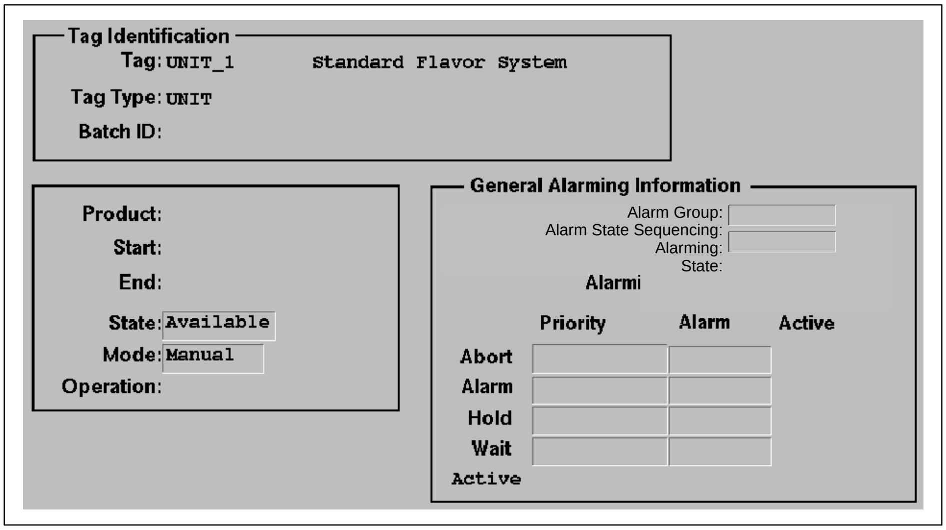


Figure 6-19 Unit Tag Detail

Table 6-15 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in a unit tag detail.

Table 6-15 Security Privileges for Unit Tag Details

Attributes	Security Privileges					
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control	Batch Control
Command	✓	✓	✓	✓	✓	✓
Mode_Cmd	✓	✓	✓	✓	✓	✓
Scan Status	✓	✓			✓	
Alarm Priority	✓	✓			✓	
Enable/Disable Alarms	✓	✓			✓	

Batch

Figure 6-20 shows a typical detail display for a batch tag.

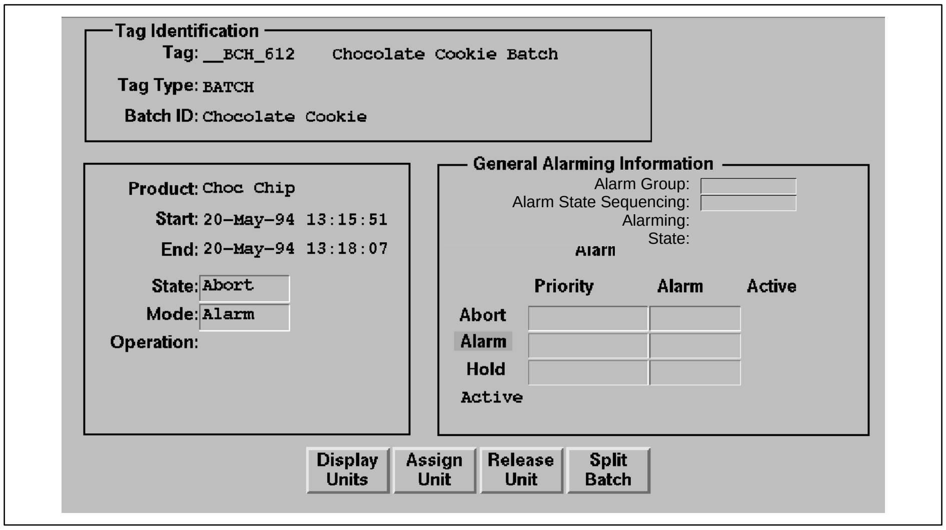


Figure 6-20 Batch Tag Detail

Table 6-16 shows the security privileges that allow an operator with the appropriate process group membership to change the values of specific attributes in a batch tag detail.

Table 6-16 Security Privileges for Batch Tag Details

Attributes	Security Privileges					
	Database Admin.	System Config.	Primary Control	Secondary Control	Tertiary Control	Batch Control
Command	✓	✓	✓	✓	✓	✓
Mode_Cmd	✓	✓	✓	✓	✓	✓
Scan Status	✓	✓			✓	
Alarm Priority	✓	✓			✓	
Enable/Disable Alarms	✓	✓			✓	

6.2 Tag Group Displays

A tag group is a collection of related tags that display information about the process in blocks called faceplates. Each tag group consists of between one and eight tags, depending on how many have been assigned to the group. OSx provides a faceplate displaying symbols and text for each tag type except the recipe area tags and the tags that represent the system nodes. Figure 6-21 shows an example of a tag group display.

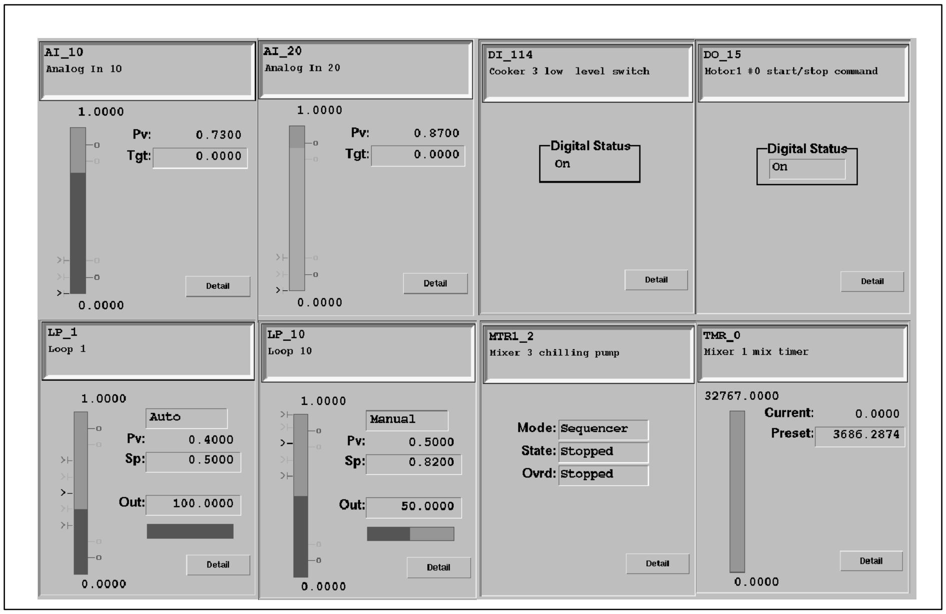


Figure 6-21 Tag Group Display

NOTE: A point on a tag group display may change to magenta, and the animation may stop, under the following circumstances:

- The OSx station loses communication with the control node.
- You deactivate the scan for a tag from a tag detail display.
- A tag contains no points that are configured for network scan.
- The controller returns invalid floating point data for tag types with floating point attributes.
- You have assigned the data to an invalid controller memory location and OSx cannot collect the data.
- You have sent the **Scan_Off** command to the OSx station.

Configuring Tag Groups

Before you create tag groups, you must first configure the tags that are to be associated with each group, as explained in Chapter 5. To configure tag groups, you must have either the Database Administration, the System Configuration, or the System Maintenance security privilege. To create a tag group, follow these steps.

1. Select **Data->Tag Group** from the menu bar (Figure 6-22).

2. Enter a name for the tag group. The space, the character combinations -> and <- and the characters ; \ , ” are invalid for tag group names. Do not use the tilde character (~) as the first character in a tag group name.

   Entering a new group name activates the select list icons. Clicking the icon displays a list of tags for you to choose from. Entering the name of an existing group displays the group description and the tags that are currently assigned to that group.

3. Enter a description for the group of up to 30 characters. The characters ; \ ” are invalid for tag group descriptions.

NOTE: OSx strips leading and trailing blank characters from tag group names and descriptions.

---

4. Enter the names of up to eight tags. Each named field corresponds to a particular faceplate position on the screen, as shown under **Position** in Figure 6-22.

You do not have to assign tags to consecutive faceplate positions. Any faceplate position may be left blank for any tag group.

When you delete a tag, OSx automatically removes the tag from its associated tag groups.

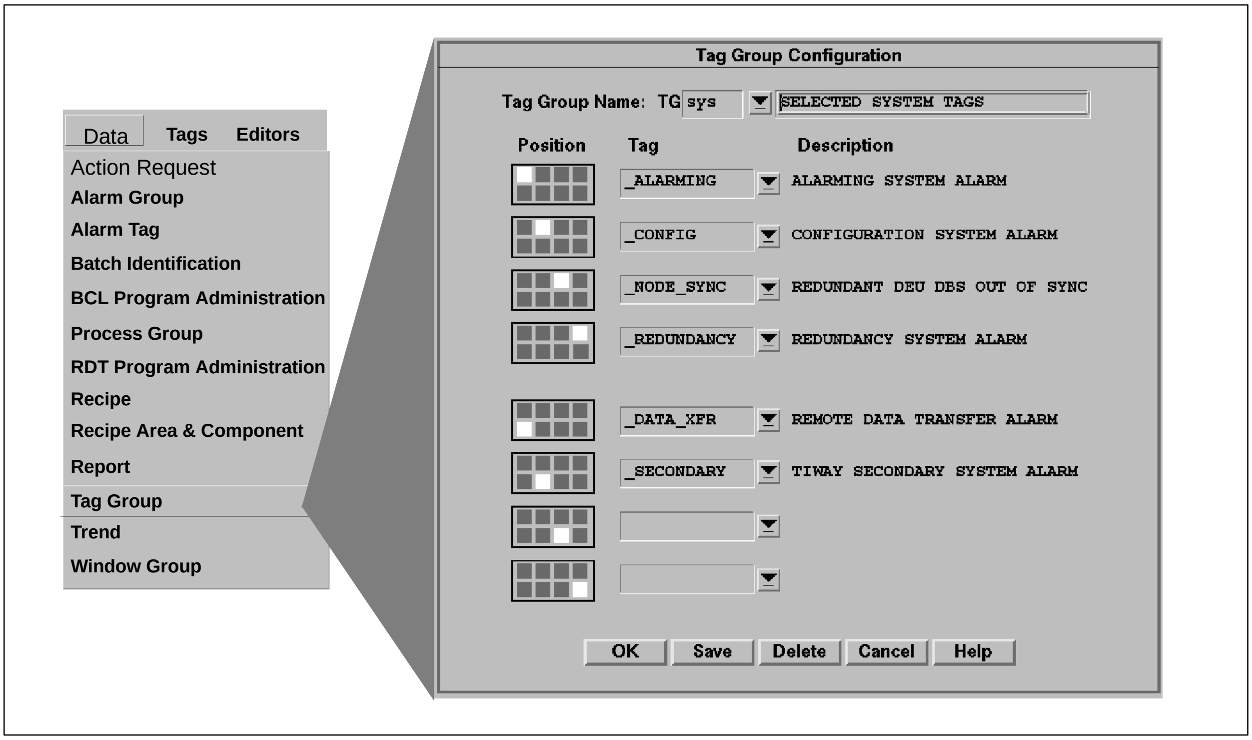


Figure 6-22 Tag Group Configuration Display

Displaying a Tag Group

A Tag Group Directory lists all configured tag groups and their corresponding descriptions. The directory lists one tag group on each line.

To view the directory of tag groups, follow these steps.

1. Set the system state to Operate. The screen displays the navigation area.
2. Select the **Directory** icon from the navigation area (Figure 6-23). A directory of display types appears.
3. Select the **Tag Group** option. The screen displays a list of tag groups.
4. Select the tag group that you want to see and press **OK**. The screen displays the tag group selected (Figure 6-24).

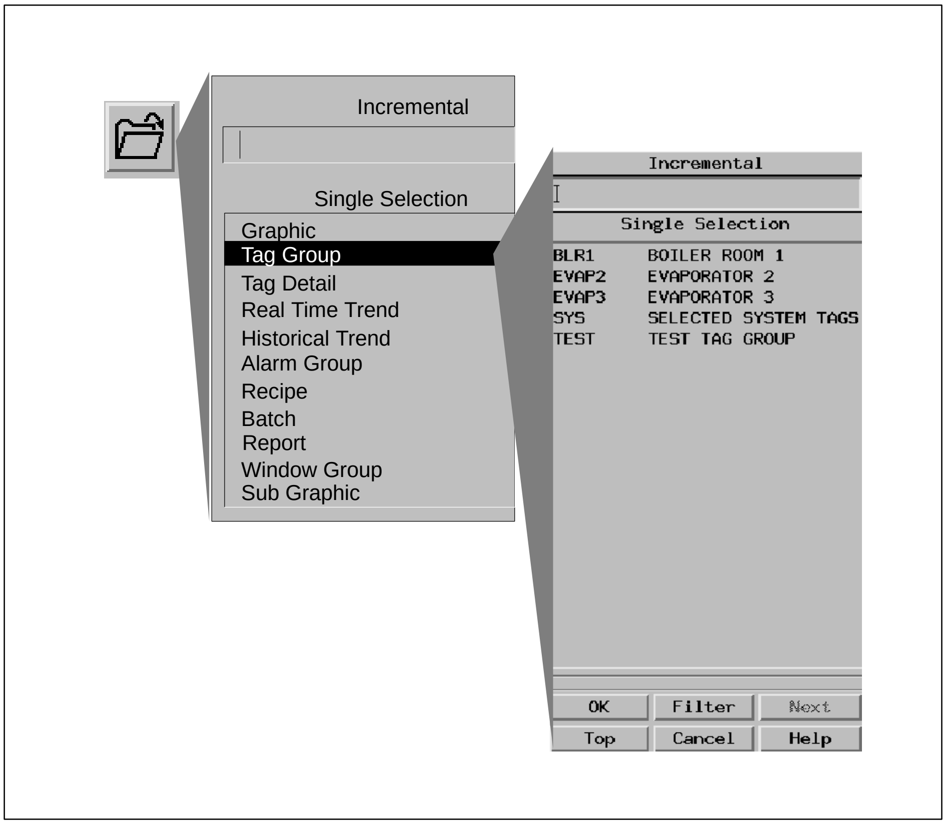


Figure 6-23 Tag Group Directory

Figure 6-24 shows an example of a tag group as it appears in the Operate state. Note that the configuration engineer has assigned only six tags to this group. You can display the tag detail for the tag by selecting a **Detail** button.

NOTE: If you display a command form for one of the tag faceplates and make a change to a field, and the change does not succeed, the value in the database does not change. However, the command form displays the value you attempted to enter. The tag faceplate continues to display the correct, unchanged value. When you redisplay the command form, it contains the original, unchanged value.

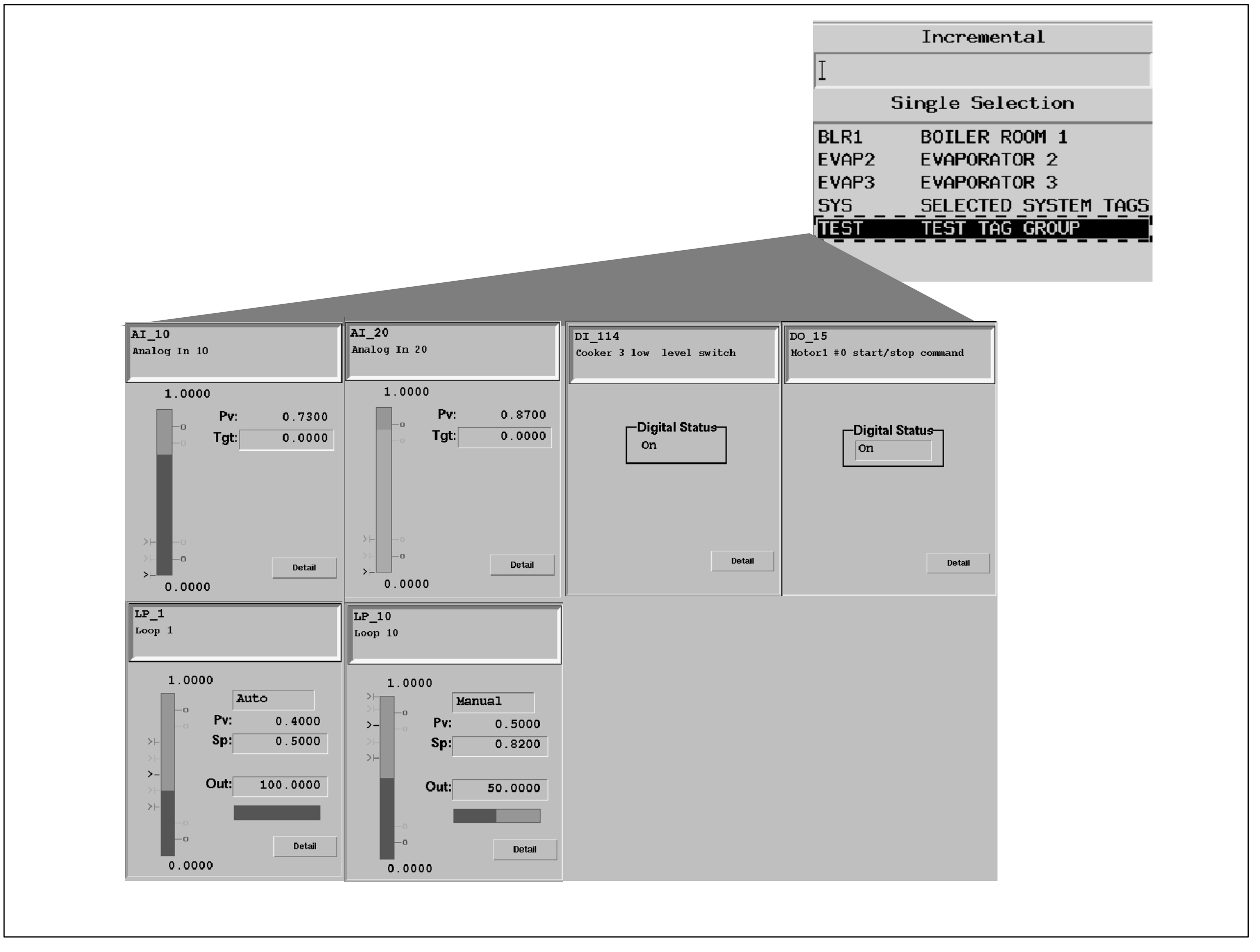


Figure 6-24 Tag Group Display

Chapter 7
Configuring System Security

7.1 User Security Configuration Overview

Security configuration involves the assignment, distribution, and maintenance of User IDs and passwords within OSx. To control system access, you assign each user a unique User ID. Only alphanumeric characters (A–Z, a–z, and 0–9) are valid for User IDs.

User IDs determine specific privileges that allow users to do certain tasks. PCS 7 OSx uses passwords to verify User IDs. For this reason, unique passwords play an important role in the operation of the SIMATIC PCS 7 OSx system. Each User ID has only one password. Only alphanumeric characters (A–Z, a–z, and 0–9) are valid for passwords. To maintain confidentiality, passwords never appear on the screen; neither when you configure them nor when users enter them.

You can add or modify users at any time. To keep a written record of all User IDs, their associated passwords, and associated security privileges, use the Security Configuration Planning Sheet in Appendix A.

OSx comes preconfigured with the name `root` as a User ID. The User ID `root` requires no password and has all security privileges.

CAUTION
Leaving the preconfigured User ID “root” with no password could jeopardize system security.   **This could allow unauthorized personnel access to the OSx system.**   **Assign a new password to the root User ID as soon as possible to preserve system security.**

OSx provides many standard security privileges. Each privilege controls access to part of the OSx system, as shown in Table 7-1.

Table 7-1 Standard Security Descriptions

Name	Level of Control
Primary Control	Access tag group (faceplate) displays to change setpoints, output, control mode, analog input, target, and timer/counter preset; command digital outputs and acknowledge alarms; commit a batch
Secondary Control	Access tag detail displays, change trend rates, override device control, and acknowledge alarms
Tertiary Control	Change alarm limits, enable/disable alarms, take alarm tags out-of-service, and acknowledge alarms
Control Tuning	Tune loops and change PID constants.
Process Cyclic Function	Define recipes for batch processes.
Real-Time Trend	Define trend groups, collection parameters.
Historical Trend	Define trend groups, collection parameters.
Report Configuration	Define report structures and schedule reports (GEN).
Operator Level Report Request [1]	Print and display control over operational reports.
System Configuration	Define system and configure user security privileges, and acknowledge alarms.
Database Administration	Configure system and acknowledge alarms (does not allow recipe configuration).
System Startup System Shutdown	Define system states and startup parameters. Assign new primary from Network Setup.
System Maintenance	Access maintenance statistics and diagnostics.
Management Level Report Request [1]	Access management reports (MGMT).
Control Application	Define unit structure, recipe templates, and so forth.
Action Request Configuration	Change Action Request configuration.
Batch Control [2]	Perform batch operations, hold, and abort.
Role Switch	Reserved for future development.
Operations	Acknowledge alarms, answer actions requests, and download recipes.
Graphical Editor	Access the graphical editor.
Re-Sync	Resynchronize the station.

1 Before you use Operator Level or Management Level Report Requests, you must configure system level user privilege and report type.
2 Refer to the *SIMATIC PCS 7 OSx Batch Programming Manual* for a complete description of the security privileges for batch control.

Table 7-2 lists the specific functions that the security privileges control. A **Y** under a particular privilege indicates that you can execute the function when your User ID is assigned the privilege. When multiple privileges are marked for a function, any one of the marked privileges allows you to execute the function.

Regardless of the security privilege assigned, you must be a member of the appropriate process group (assuming they are configured) in order to interact with tag details, alarm groups, or action requests that are associated with a specific process group.

Table 7-2 Functions Controlled by Security Privileges

System Function	Required Privilege																				
	Primary Control	Secondary Control	Tertiary Control	Control Tuning	Process Cyclic FN	Real-Time Trend	Historical Trend	Report Config.	Operator Level Report Request	System Config.	Database Admin.	System Startup/ Shutdown	System Maint.	Mgmt. Level Report Req.	Control Appl.	Action Request Config.	Batch Control [4]	Role Switch [5]	Operations	Graphical Editor	Re-Sync
Configure Trends						Y	Y			Y	Y										
Reports: Set Destination, Schedule, Copy								Y													
Preview Report	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Print Report Format	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Delete Report Format								Y		Y	Y										
Configure Tag Group										Y	Y										
Configure Recipes [1]					Y																
Control Recipe function [2] (selecting, modifying)	Y				Y					Y	Y								Y		
Acknowledge Alarm	Y	Y	Y							Y	Y								Y		
Disable/Enable Alarm Group			Y							Y	Y										
Respond to Action Request [2]	Y	Y	Y								Y								Y[3]		
Modify Tag Value [2]	Y	Y	Y							Y	Y										
Modify Loop Tune Parameter				Y						Y	Y										

1 The user can view all configuration/displays with a User ID and password.
2 The Primary, Secondary, and Tertiary privileges modify a subset of tag values. See Chapter 6 for tag-specific differences. If modifying a recipe function or an action request requires a change to a tag/attribute combination, then other privileges may be required.
3 This privilege allows you to respond to a View action request. Viewing the action request answers it.
4 Refer to the *SIMATIC PCS 7 OSx Batch Programming Manual* for a complete description of security for batch control.
5 Reserved for future development.

Table continued on next page.

Table 7-2 Functions Controlled by Security Privileges (continued)

System Function	Required Privilege																				
	Primary Control	Secondary Control	Tertiary Control	Control Tuning	Process Cyclic FN	Real-Time Trend	Historical Trend	Report Config.	Operator Level Report Request	System Config.	Database Admin.	System Startup/ Shutdown	System Maint.	Mgmt. Level Report Req.	Control Appl.	Action Request Config.	Batch Control [4]	Role Switch [5]	Operations	Graphical Editor	Re-Sync
Modify Alarm Priorities and Limits; Enable/Disable Alarms and Scan of Tags			Y							Y	Y										
Modify Value in Tag Group [1]	Y	Y	Y							Y	Y										
View/Print Report/Log File	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Delete Report/Log File								Y		Y	Y										
Download Recipe	Y				Y					Y	Y								Y		
Edit/Compile Report								Y		Y	Y										
Configure Screen Hierarchy										Y	Y										
Configure Alarm Group										Y	Y										
Configure Alarm Tag										Y	Y										
Change System State			Y							Y	Y	Y	Y								
Graphical Editor Operations										Y	Y									Y	
Configure Calendar										Y	Y										
Configure User ID										Y	Y										
Modify Tag Capacities										Y	Y										
Install Tags [2]; Access Tag Configurator										Y	Y										
Configure Initial State Transition										Y	Y	Y									
Configure Event Scan Period										Y	Y	Y									
Configure Alarm Refresh Rate										Y	Y	Y									
Configure Data Staleness Period										Y	Y	Y									
Configure Printers										Y	Y										
Screen Print Setup										Y	Y										

1 The user can view all configuration/displays with a User ID and password.
2 OSx allows you to begin working before notifying you if your security privilege is inappropriate. Any changes that you make are not saved.
4 Refer to the *SIMATIC PCS 7 OSx Batch Programming Manual* for a complete description of security for batch control.
5 Reserved for future development.

Table continued on next page.

Table 7-2 Functions Controlled by Security Privileges (continued)

System Function	Required Privilege: Primary Control	Secondary Control	Tertiary Control	Control Tuning	Process Cyclic FN	Real-Time Trend	Historical Trend	Report Config.	Operator Level Report Request	System Config.	Database Admin.	System Startup/ Shutdown	System Maint.	Mgmt. Level Report Req.	Control Appl.	Action Request Config.	Batch Control [1]	Role Switch [2]	Operations	Graphical Editor	Re-Sync
Set OSx Date and Time			Y							Y	Y		Y								
Run Diagnostics													Y								
Handle Errors from Data Archive										Y	Y										
Initialize Tape for Data Archive and Restore from Archive										Y	Y										
Configure Data Archiving										Y	Y										
Configure Action Request [3]											Y					Y					
Create/Modify Process Groups										Y	Y										
Configure Recipe Area															Y						
Set Station Role										Y	Y										
Switch Station Roles										Y	Y										
Network Setup										Y	Y	Y[4]									
Network Status	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Resynchronize Station										Y	Y										Y
Configure Window Groups										Y	Y										

1 Refer to the *SIMATIC PCS 7 OSx Batch Programming Manual* for a complete description of security for batch control.
2 Reserved for future development.
3 OSx allows you to begin working before notifying you if your security privilege is inappropriate. Any changes that you make are not saved.
4 Only the New Primary command is accessible for this security privilege.

7.2 Configuring User Security

Choose one of the following for assigning security privileges:

- Continue to use the initial security while configuring the OSx system. Then configure the remaining securities after you have completed all configuration tasks and system verification checks.
- Configure detailed security privileges for all system users before configuring the OSx system.

The SIMATIC PCS 7 OSx system can be in either the Offline or the Operate state for security configuration.

Before you can enter a User ID, you must have the System Configuration or Database Administration security privilege. The `root` User ID has these privileges by default.

CAUTION

Any user with System Configuration or Database Administration security privileges can add new users, and modify or delete existing users. The root User ID has these privileges by default.

The security of your system depends on the appropriate assignment of security privileges.

Assign the root User ID and System Configuration or Database Administration security privileges only to properly authorized personnel.

When you assign a User ID, remember these general guidelines:

- User IDs can contain up to seven characters, and passwords can contain up to eight characters.
- User IDs and passwords must consist of letters and integers only.
- OSx does not distinguish between upper and lower case letters for User IDs and passwords.
- If you change privileges for a User ID while the user is logged on, your changes do not take effect until the user logs off and logs on again. If the user is already logged on, logging on again without logging off first does not cause the new security privileges to take effect. The user must log off, and then log back on.

Configuring User Security (continued)

Assigning User IDs and Passwords

To assign a User ID and password follow these steps:

1. Select **Startup->User Security** from the menu bar (Figure 7-1).
2. Enter a User ID and a password. Only alphanumeric characters (A–Z, a–z, and 0–9) are valid for User IDs and passwords. OSx verifies that the User ID is unique. If the User ID and password that you enter already exist in the system, the screen displays the privilege levels that were assigned previously.
3. Select the privileges for this User ID. Table 7-2 on page 7-4 lists the operations that are allowed by the various privileges. By default, all privileges are set to **No**.

NOTE: The password for the default user IDs (ENGINR, GENRPT, MGTRPT, OPER, OPRRPT, ROOT, and SUPER) in this utility is the space character. Press the spacebar followed by **Enter** in the password field.

To assign a User ID to a process group, follow the procedure on page 7-10.

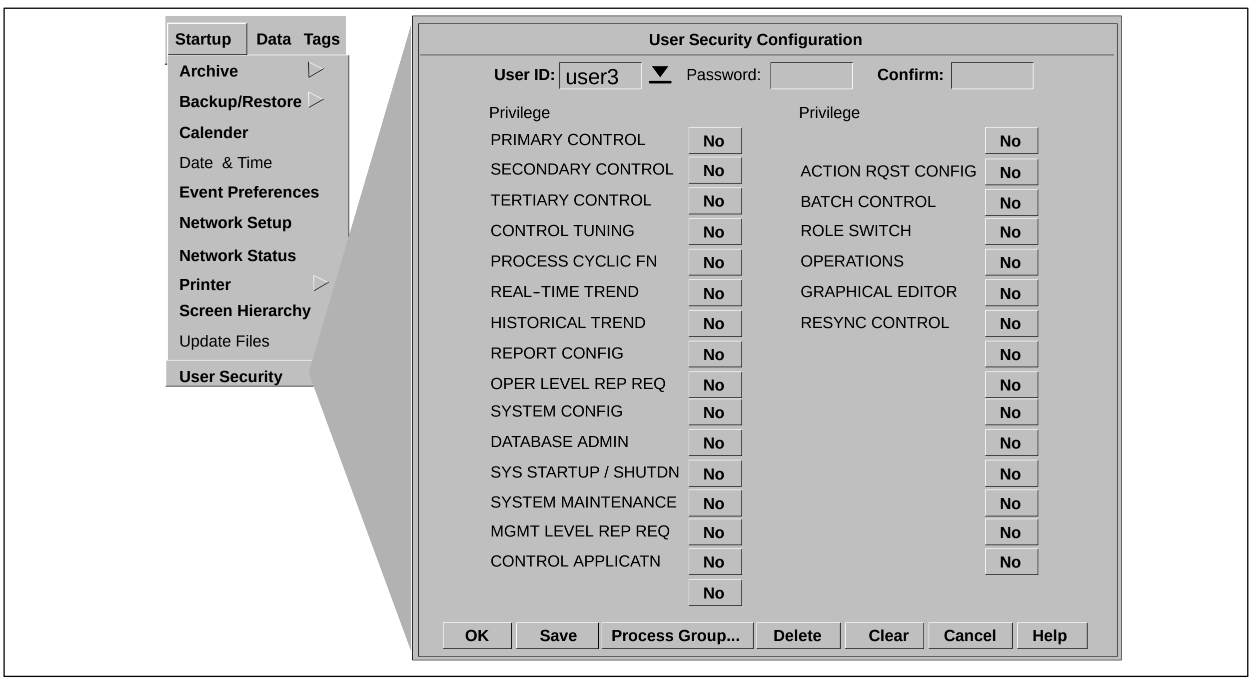


Figure 7-1 User Security Configuration Dialog Box

Modifying a Password

You can modify a user's password if you have the appropriate security privilege. Follow these steps:

1. Enter the User ID.
2. Place the cursor in the password field, and type a new password.
3. Confirm the password.

If you configure a User ID that is a blank, and you want to modify its password or security privileges in the User Security Configuration window, you cannot select the blank User ID from the User ID select list. You must enter a space in the User ID field by pressing the space bar followed by the **Enter** key to access the properties of a blank User ID.

Deleting a User ID

You can delete a User ID at any time. Follow these steps:

1. Enter the User ID.
2. Select the **Delete** pushbutton on the User Security Configuration dialog box. OSx removes the User ID from the database.

CAUTION
Leaving the preconfigured User ID "root" with no password could jeopardize system security. **This could allow unauthorized personnel access to the OSx system.** **Assign a new password to the root User ID as soon as possible to preserve system security.**

Recording User ID and Password Changes

The following changes are automatically recorded in the Operator Change Log:

- Adding a new User ID and password
- Modifying an existing user's password
- Modifying an existing user's security privileges
- Deleting a User ID

Assigning User IDs to Process Groups

A process group does not have to have a description before you assign a User ID to it. Therefore, you can assign a User ID to one or more process groups, whether or not they have descriptions, and then enter the descriptive text for the process groups at another time. Refer to Chapter 2 for detailed information about how to assign User ID membership in a process group for differential process control.

Until you disable its membership in a process group, a User ID is linked by default to all thirty-two process groups. To change membership to a process group, follow these steps.

1. Select the **Process Group** pushbutton on the User Security Configuration display. The screen displays the Process Group Assignment screen (Figure 7-2).

2. Select **No** for each process group that you do not want to associate with a User ID. The **Yes** field preserves the link between the User ID and the process group. The **No** field disables the link between the User ID and the process group.

NOTE: Always set unused process groups to **No**. Otherwise, unintended memberships may occur between users, tags, alarm groups, and action requests.

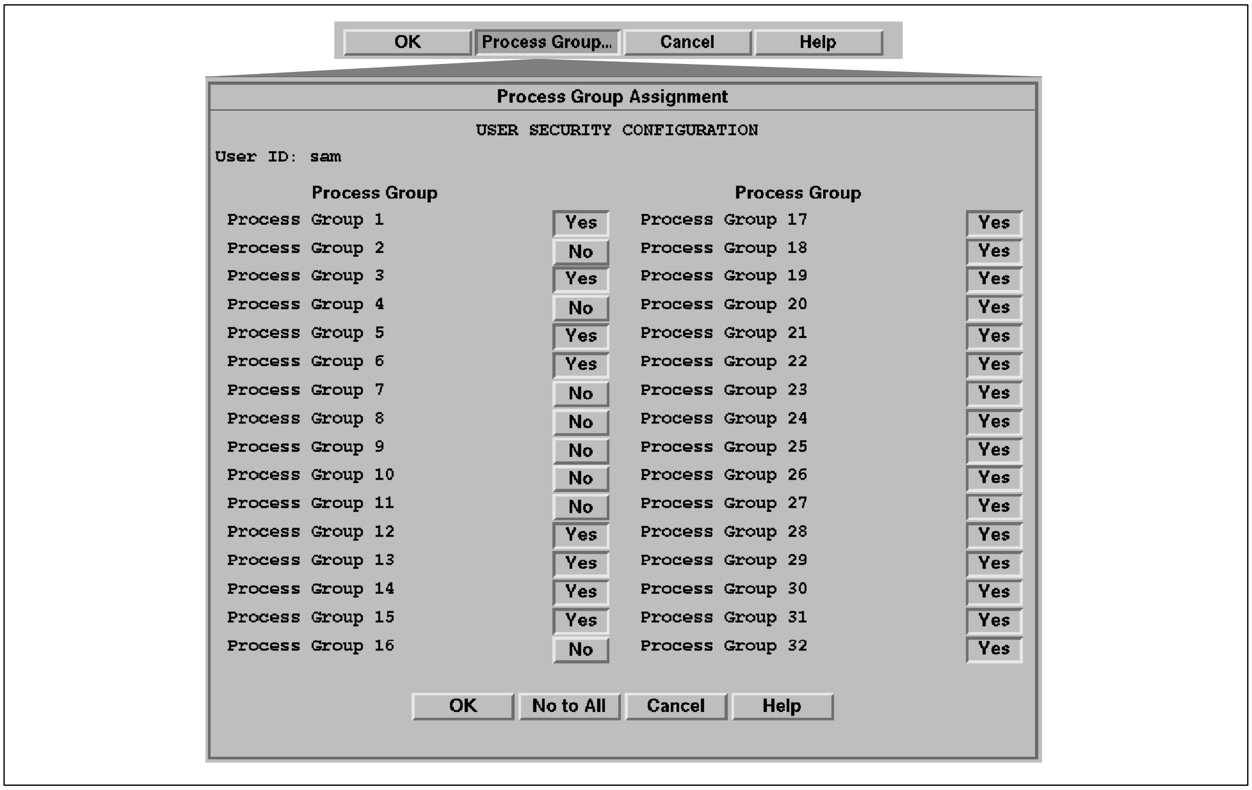


Figure 7-2 Process Group Assignment Display

Chapter 8

Configuring Trends

8.1 Configuring Trend Screens

Trend displays show the history of a controlled process by plotting the analog and digital values of tags. The SIMATIC PCS 7 OSx system enables you to configure trends that you can view in historical or real-time mode. In addition to current data, real-time trends display the history of a process for a predefined period of time. After the historical data is completely displayed on a real-time trend, OSx updates the trend display with real-time data.

Trend displays plot the analog and digital values on X and Y axes. The X axis of a trend display represents elapsed time. The Y axis represents values in engineering units. Trend groups can consist of up to eight analog and eight discrete points.

To configure trends, you must have the Database Administration, the System Configuration, the Real-Time Trend, or the Historical Trend security privilege. The system can be in either the Operate or the Offline state. Select the **Data->Trend** from the menu bar to display the Trend Configuration dialog box (Figure 8-1). Click the **General** tab, and fill in the fields described below.

Name Enter a name for the trend. You cannot fill out the rest of the form until you enter a name. Each trend must have a unique name of up to five characters, in upper or lower case. The space, the character combinations –> and <– and the characters \ , ” ; are invalid for trend names.

Description Enter a description for the trend. The trend description is optional and can have up to 30 characters. The characters \ ” ; are invalid.

Historical and Real-Time Viewing Window Enter a time span for each window. A viewing window allows you to specify the amount of time to display data on a trend. For example, if you set the viewing window to 20 hours, the screen displays data accumulated over the last 20 hours.

Default viewing periods are provided by OSx: 8 hours for historical trends and 40 minutes for real-time trends. The minimum viewing period for both real-time trends and historical trends is 1 minute. The maximum viewing period for real-time trends is 99 hours. The maximum viewing period for historical trends is 99 days.

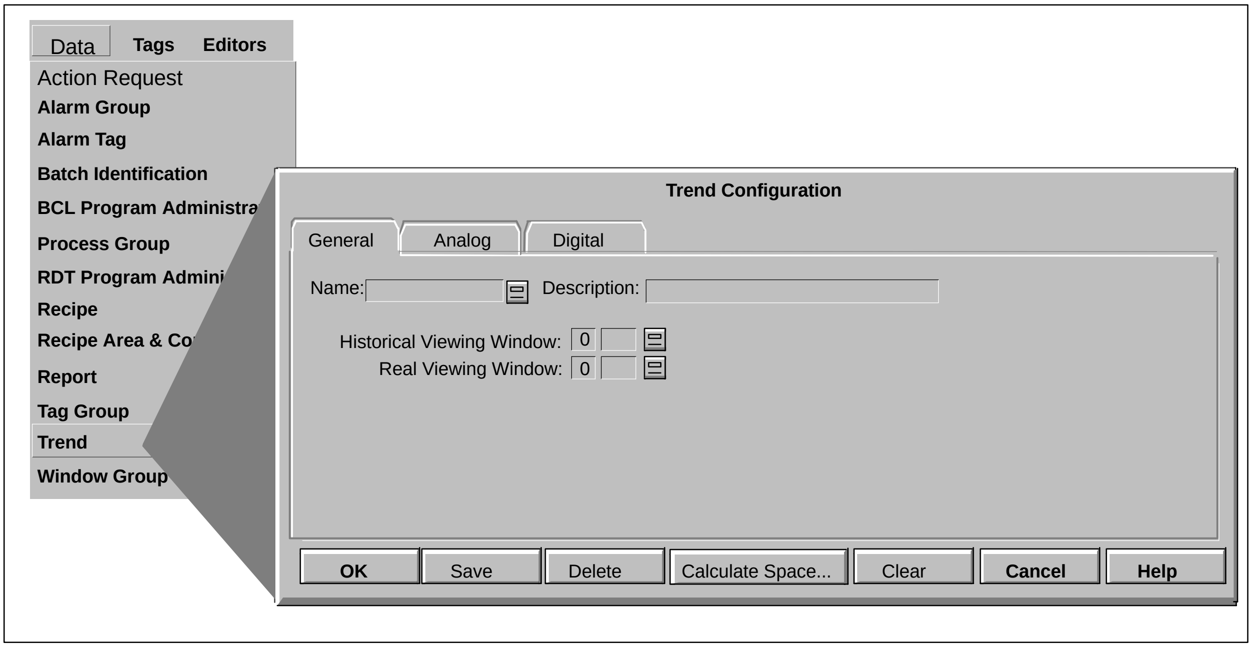


Figure 8-1 Trend Configuration Screen

Continue configuring the trend on page 8-4.

After you enter the trend characteristics, you need to specify the tags that you want to trend. You can trend between zero and eight each of analog and digital points for each group. Analog tags require a specified range, and digital tags require bit names. For information on bit names, refer to Chapter 3 on defining tags.

Trend configuration information updates automatically if there are any changes to tag information. For example, if a tag is deleted from the database, OSx automatically removes the tag from all trend groups.

To configure analog points for the trend, click the **Analog** tab and fill in the fields shown in Figure 8-2. To configure digital points for the trend, click the **Digital** tab and fill in the fields shown in Figure 8-3. The fields are described below and in the following pages.

Tag and Attribute Enter a tag name, or use the long-list display tool to the right of the field to choose from a list of tags. The default attribute name, range, rate, and duration appear. If you want to change the default values, you can type over them, or use the long-list display tool to the right of the field to choose from a list of valid entries.

If the tag/attribute pair that you enter is in another trend group, any changes that you make in the rate or duration are reflected in all the other trend groups containing that tag. A particular tag/attribute pair is only collected at one rate.

Range (analog values only) Enter a range, or Y axis, for the trend. Range specification enables you to view the point values of interest. OSx plots only the analog tag values within the specified range on the trend. It plots values outside of that range at the extreme points of the graph.

NOTE: If range values are not specified for each analog tag, OSx defaults to the high and low ranges specified during tag configuration.

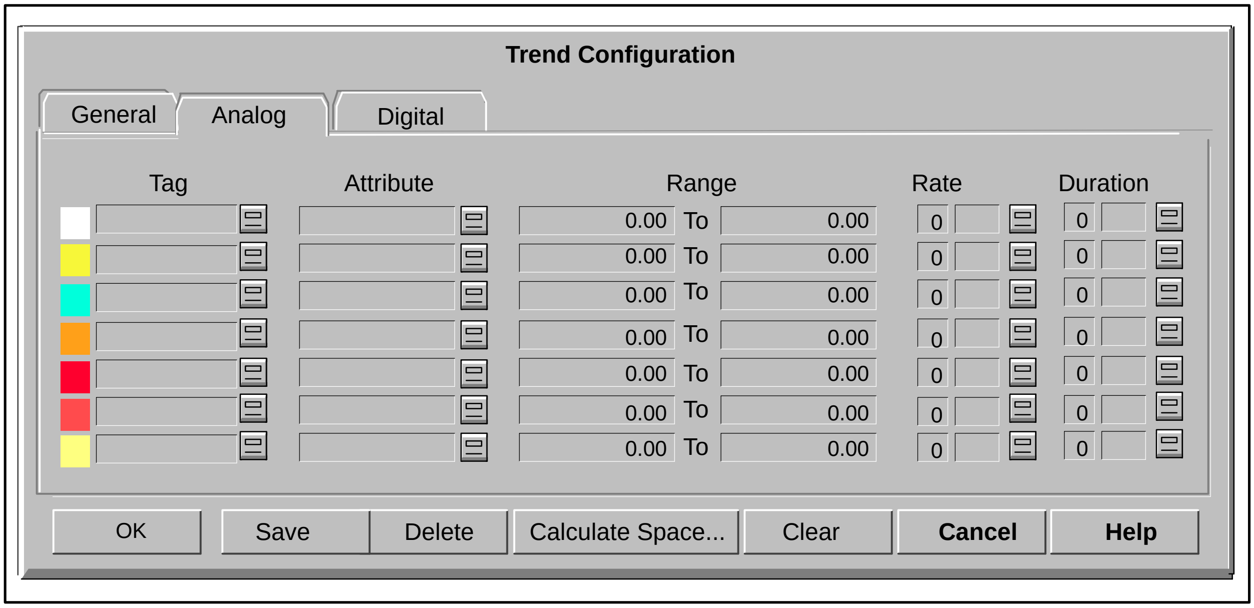


Figure 8-2 Configuring Analog Trends

Bit Name (digital values only) When you enter the name for a digital tag in the Tag field, a default bit name appears in the Bit Name field (Figure 8-3). If you want to change the default bit name, you can type over it, or use the long-list display tool to the right of the field to choose from a list of valid bit names for that tag. You can select any bit associated with a digital tag/attribute pair to trend. Table 8-1 lists the default bit names. Some tag types allow you to specify other attributes besides the Status attribute.

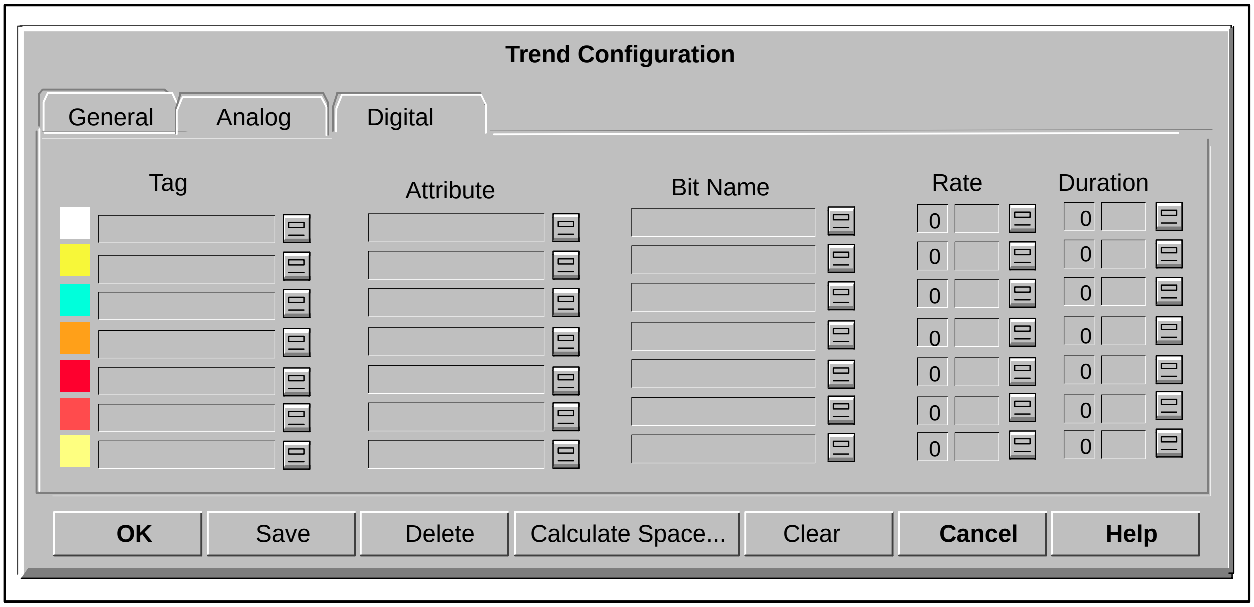


Figure 8-3 Configuring Digital Trends

Table 8-1 Default Bit Names for Digital Trends

Tag Type	Attribute	Default Bit Name	Bit Position	Tag Type	Attribute	Default Bit Name	Bit Position
AI	Status	Active	0x0001	MTR1	Status	Running	0x8000
AO	Status	Auto_Man	0x8000		Override	Ovrd_Fdbk	0x8000
AREA	Status	Err_Comp	0x0080		Mode_Cmd	Mode	0x8000
BCH	Status	Active	0x0001		Setpoint	Setpoint	0x8000
CALC	Status	Active	0x0001	MTR2	Status	Running	0x8000
CTR	Status	Active	0x0001		Override	Ovrd_Low	0x4000
DI	Status	Data_Val	0x8000		Mode_Cmd	Mode	0x8000
DI10	Status	Active	0x0001		Setpoint	Setpoint	0x8000
DO	Status	Data_Val	0x8000	RMTR	Status	Running	0x8000
	Command	Cmd_Val	0x8000		Override	Ovrd_Rev	0x4000
DO10	Status	Active	0x0001		Mode_Cmd	Mode	0x8000
	Data_Val1	Cmd_Val1	0x8000		Setpoint	Setpoint	0x8000
	Data_Val2	Cmd_Val2	0x4000	SYSTEM	Status	Failed	0x8000
	Data_Val3	Cmd_Val3	0x2000	TEXT	Status	Active	0x0001
	Data_Val4	Cmd_Val4	0x1000	TMR	Status	Active	0x0001
	Data_Val5	Cmd_Val5	0x0800	UNIT	Status	Active	0x0001
	Data_Val6	Cmd_Val6	0x0400	VLV1	Status	Opened	0x8000
	Data_Val7	Cmd_Val7	0x0200		Override	Ovrd_Fdbk	0x8000
	Data_Val8	Cmd_Val8	0x0100		Mode_Cmd	Mode	0x8000
	Data_Val9	Cmd_Val9	0x0080		Setpoint	Setpoint	0x8000
	Data_Val10	Cmd_Val10	0x0040	VLV2	Status	Opened	0x8000
IVAR	Status	Active	0x0001		Override	Ovrd_Open	0x8000
LOOP	Status	Active	0x0001		Mode_Cmd	Mode	0x8000
	Mode	Cascade	0x2000		Setpoint	Setpoint	0x8000

Rate Enter a sampling rate. For historical trends, the rate determines the period of time between data samples. For example, if you specify a rate of three minutes, a sample is taken every three minutes. If you do not specify the sampling rate, OSx uses the default rate of two minutes.

For real-time trends, the sample rate is based on a formula that takes the number of points to be displayed and the width of the viewing window (in seconds) into consideration.

OSx saves historical data on an exception basis. It scans the current process values at the rate specified, but saves only those values that have changed since the previous sample. If you do not want historical data, set the sample rate high and the duration low. This causes OSx to sample infrequently and to save it for only a short time. The minimum rate period that you can specify is one second.

To estimate whether or not historical trending is likely to fall behind or to miss collecting some data points, you can calculate the average trend rate for a particular trend configuration.

The average trend rate is expressed in points per second and is calculated by taking the inverse of the trend rate for all configured trend points and adding them together. A single trend point is represented by one line on a real-time or historical trend display. An example of how to calculate the estimated trend rate is shown below.

Example: 1000 analog and digital trend points are configured in the following ways:

```
20  points at a  1 second rate =  20 *  1/1 =  20
100 points at a  2 second rate = 100 *  1/2 =  50
500 points at a  5 second rate = 500 *  1/5 = 100
100 points at a 10 second rate = 100 * 1/10 =  10
280 points at a 20 second rate = 280 * 1/20 =  14
                                              ----
                                              194 pts/sec
```

A configured trend collection rate of 194 points per second may be difficult to maintain if system loading is heavy. The tag detail for the _HT_COLLECT system tag shows what the real average trend collection rate is. The Value attribute for this tag is a running average of how many points were collected per second for the last 30 seconds. If the trend collection rate is too high, an _HT_COLLECT alarm warns you that some trend data was not collected as configured.

You can configure an OSx report to generate the estimated required trend performance rate, as shown below.

```
Cell     Contents
----     ------------------------------
A1   :   <#readlist (ht_schedule:rate;all)
B1   :   <#calclist (1.0/a1)
C1   :   @sum (B1)
```

Duration Specify duration, the length of time that OSx saves data for a particular tag.

A trend file contains approximately 800 total sampled points. This is true regardless of the sample time of the trended point or the specified duration.

When the sample rate is low (for example, once every five seconds), OSx maintains multiple trend files to support the specified duration time. It deletes the oldest trend file when the duration time is exceeded. Regardless of when a trend file is deleted, OSx archives it if the automatic archive feature is enabled. See the *SIMATIC PCS 7 OSx System Administration Manual* for information about the archival of trend data.

If the sample rate is high (for example, once every ten minutes), or the data changes infrequently and the duration time is short, then the duration time may be exceeded before the trend file reaches its capacity. In this case, OSx does not delete the file until all trended points contained in the file have exceeded the duration time. Since the file size is fixed at about 800 total sampled points, some of the trended points are available for viewing beyond the specified duration time.

NOTE: When you use the same tag in several trend groups, and you change the rate/duration in one of those groups, the rate/duration changes on all the other trend groups associated with that tag.

If you want to trend a tag for varying rates or durations, you need to configure more than one tag for the same network point and trend each of the tags. If you do not want to save any historical data for this trend point, set the duration as one minute.

8.2 Modifying Trend Points

If you make trend configuration changes in the Operate state, follow these guidelines:

- When you modify points, always make modifications to the trend group directly. Do not delete the group and then add it back immediately. Doing so may cause the point(s) that you changed not to be collected until you transition the system to Offline and back to Operate again.
- If you must delete points and then add them back, allow a wait period to elapse before you add the points back. Use the following algorithm to determine the wait period:

  number of points deleted * the maximum trend rate for any point

  For example, if you delete 10 points and the maximum trend rate is 2 minutes, then wait 10 * 2 = 20 minutes before re-adding the points that you just deleted. This should avoid having to transition states to maintain trend collection for points deleted and then added back.

8.3 Reading Disk Space Information

Disk space can be determined in the following ways for all trend groups and for the currently displayed trend :

- Maximum (worst case) amount of disk space available for historical trend data. This serves as a checkpoint between the actual disk usage for a particular group and the total amount of disk space available for historical trend data.
- Actual amount of disk space used to store historical trend data.
- Amount of free disk space available to store historical trend data. Always refer to the **Free Space** field when configuring historical trends to ensure that disk space for that trend is available.

To calculate disk space usage, select the **Calculate Space...** pushbutton on the Trend Configuration screen (Figure 8-4).

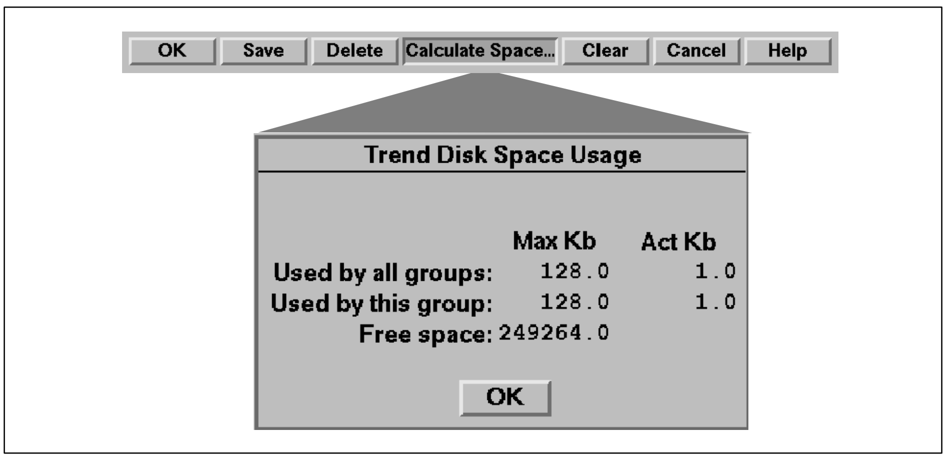


Figure 8-4 Trend Disk Space Usage Display

Storage Limits

OSx stores historical trend data in files in the **/usr/tistar/hist/trend** directory on the hard disk. In addition to the disk space limitation (determined in the trend configuration), there is also a limit on the number of files allowed.

The operating system requires an "i-node," or file slot, for each unique file in the system. Use the `df` command from an XTerm window to determine the number of i-nodes available. The displayed i-nodes for the **/usr/tistar** file system are available for trend files.

NOTE: Trending is not the only process that writes files and uses up i-nodes: reports and custom programs can also create files in the **/usr/tistar/hist** directory. The number of i-nodes available varies as the system runs.

The number of trend data points stored in each file is fixed: 819 for analog points; 1,024 for digital points. To calculate the number of files needed to hold the desired trend data, divide the number of trend samples to be stored by the number of trend samples allowed per file:

$$\text{number of samples per trended point} = \frac{\text{trend duration time (sec)}}{\text{sample rate (sec/sample)}}$$

(trend duration time is the configured time to keep data in seconds)

$$\text{number of files needed per trended point} = \frac{\text{number of samples}}{\text{samples per file}}$$

The total number of files needed equals the sum of files needed for all trended points.

Example

The following example assumes 300,000 available i-nodes and shows 25 digital points trended for 15 days at a sample rate of 10 seconds. The actual number of i-nodes would be the value displayed using the **df** command described in the previous section. This number varies according to the operating system platform and system file usage.

$$\frac{15 \text{ (days)} * 24 \text{ (hours/day)} * 3{,}600 \text{ (sec/hour)}}{10 \text{ sec/sample}} = 129{,}600 \text{ samples needed}$$

$$\frac{129{,}600 \text{ samples}}{1{,}024 \text{ samples/file}} = 127 \text{ files needed per point}$$

(fractional files count as one file)

The total files needed = 25 trended points * 127 files per point = 3,175.

If the system had 300,000 available i-nodes, then there would be 296,825 (300,000 – 3,175) after the trend date has filled up.

To find out the lowest sample rate that could be configured for 100 digital points, simply re-arrange the calculation.

$$\frac{300{,}000 \text{ files}}{100 \text{ trended points}} = 3{,}000 \text{ files allowed per trended point}$$

3,000 files * 1,024 points per file = 3,072,000 samples that can be stored.

The trade-off is now between duration versus sample rate: larger sample rates give longer durations. The following calculation gives the duration for a sample rate of 5 seconds:

duration = 3,072,000 samples * 5 sec/sample = 177.8 days

NOTE: These calculations express a worst case data usage where the value of the data point is different each time it is sampled. If the value of a data point does not change, that value is not written to file.

Monitoring Disk Space

OSx stores both report output and historical trend files on the **/usr/tistar** file system. Monitor OSx usage of disk space in the appropriate file system over the course of a day after you have configured the system. If the rate of disk space usage is too high, you can change the configuration to reduce the rate.

To monitor OSx usage of disk space, follow these steps:

1. Set the system to the Operate state, and select the Directory icon in the navigation area.
2. Select **Tag Detail** from the select list, and press **OK**.
3. Select _OSX_FILSYS to monitor the file system.

In the tag detail, the Value field shows how full the file system is as a percentage. This number varies over the course of a day, depending on system activity. It generally increases until the end of the day, when archiving transfers historical data to archive media.

To clear disk space for system operation, you can remove or truncate outdated files from the OSx screen or from an OSx terminal window. See the chapter on troubleshooting in the *SIMATIC PCS 7 OSx System Administration Manual* for procedures.

To further reduce disk space usage, reconfigure your system in one or more of the following ways:

- Trend only those points important for historical archiving and for operation analysis.
- Trend fewer points, or points that change less often.
- Decrease trend file duration.
- Decrease the rate that points are trended.
- Schedule reports less frequently.
- Direct report outputs to a printer instead of to files.
- Decrease the amount of text that a report produces.
- Decrease reports output file duration.
- Decrease log file duration.

8.4 Viewing Window

Determining Historical Viewing Window Size

When determining the historical viewing window, remember the relationship of display rate to sample rate. If the viewing window is set too large, the trend data can be missing or distorted.

Use the following equation for calculating the display rate, assuming an 924 points per display default. (You can change this value by resizing the Trend DDO. See the *SIMATIC PCS 7 OSx Graphical Editor Manual* for more information.)

$$\frac{\text{window size (time)}}{\text{924 (points per display)}}$$

The following equation calculates a viewing window of 140 minutes:

$$\frac{\text{140 minutes (8,400 seconds)}}{\text{924 points}} \qquad = \text{9 seconds per point}$$

If the sample rate for any of the trended points in the trend group is less than nine seconds, there could be several trend points to display for a single display point. In that case, the displayed trends may not be accurate.

Determine a window size that provides a display rate less than the lowest trend sample rate for that trend group. This helps to reduce the potential for distorted data.

Figure 8-5 shows how data can be distorted if the viewing window is too large: in the example, the trended point is a digital value sampled every five seconds; the historical viewing window is set to 140 minutes.

- The dots represent the trend data that was captured: one data point every five seconds.
- The dotted lines represent the times when a display point is created (for a viewing window of two hours, a display point is created every nine seconds).

The circles show the data value chosen for each display point and the bottom time line shows the resulting trend display.

- When no data point falls exactly on a display point, the display point assumes the value of the last data point seen. The result is that points A and B are missed.
- Point C is not displayed because it occurs after one display point and the next display point is covered by point D.

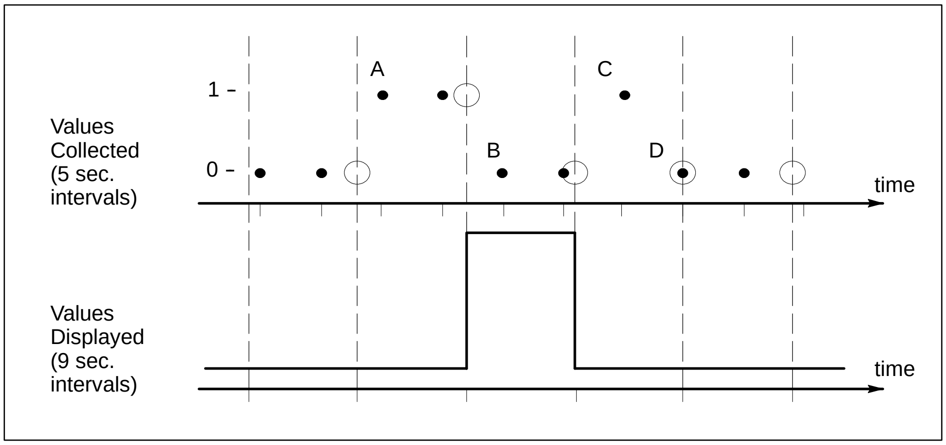


Figure 8-5 Example of a Too-Large Viewing Window

Example

The following example calculates a viewing window for a trend group with these parameters:

- Lowest trend rate in a trend group is five seconds.
- historical viewing window = sample rate * 924

  = 5 seconds * 924
  = 4,620 seconds
  = 77 minutes

For this group, a viewing window size smaller than 77 minutes gives a display in which data are correctly represented.

8.5 Displaying Trend Data

Accessing the Trend Directory

The Trend Directory lists the configured trend names and their corresponding descriptors. The directory lists one trend on each line.

To access trend displays, follow these steps:

1. Set the system state to Operate. The screen displays the navigation area on all nodes where a user is logged in.
2. Select the **Directory** icon from the navigation area. A directory of display types appears.
3. Select either the **Historical Trend** or the **Real Time Trend** option. The screen displays a list of trends.
4. Select the trend that you want to see. OSx plots the trend on the screen.

Real-Time Trend Displays

Real-time trend data is collected and displayed in strip-chart format with the oldest value on the left and the newest value on the right. See Figure 8-6. You can adjust the size of the data display in minutes or hours. When you request a real-time display, OSx first displays historical values for the duration specified for the viewing window.

OSx displays trend data as a continuous line. As new real-time data appears on the right side of the screen, the older trend data scrolls off the screen to the left. The current value for each variable is shown in numeric format to the right of the chart. If no data is available for a specific period of time, the trend line turns magenta for that period.

When you select a new display to replace the currently displayed real-time trend, all previously displayed real-time data is lost. If you return to the previous trend, OSx first displays historical data and then real-time data.

To obtain an exact value at a given time, move the cursor into the graph area, placing it at the time needed, and click the left button. A vertical line appears at the selected time and a pop-up list displays values for each tag in the plot (Figure 8-6).

Figure 8-6 Real-Time Trend Display

Historical Trend Displays

Historical trend displays are similar to real-time trend displays, but are not updated.

The historical trend display plots the data for the time range you specify. To change a trend's start and end time, select the **Change Time** target. The historical data appears compressed if the data changed rapidly during the specified time range. Historical data does not appear at all if the tag/attribute pair was not configured for trending during the specified time. A trend line is magenta if the tag/attribute pair was not scanned during the requested time period.

To obtain an exact value at a given time, move the cursor into the graph area, placing it at the time needed, and click the left button. A vertical line or dipstick appears at the selected time and a popup list displays values for each tag in the plot (Figure 8-7).

A historical trend gives an approximate viewing-window time for its data. The data is correct, but you may not get to the endpoints of the trend with the dipstick. Therefore, make sure that the time ranges are adequate for the data that you want to view.

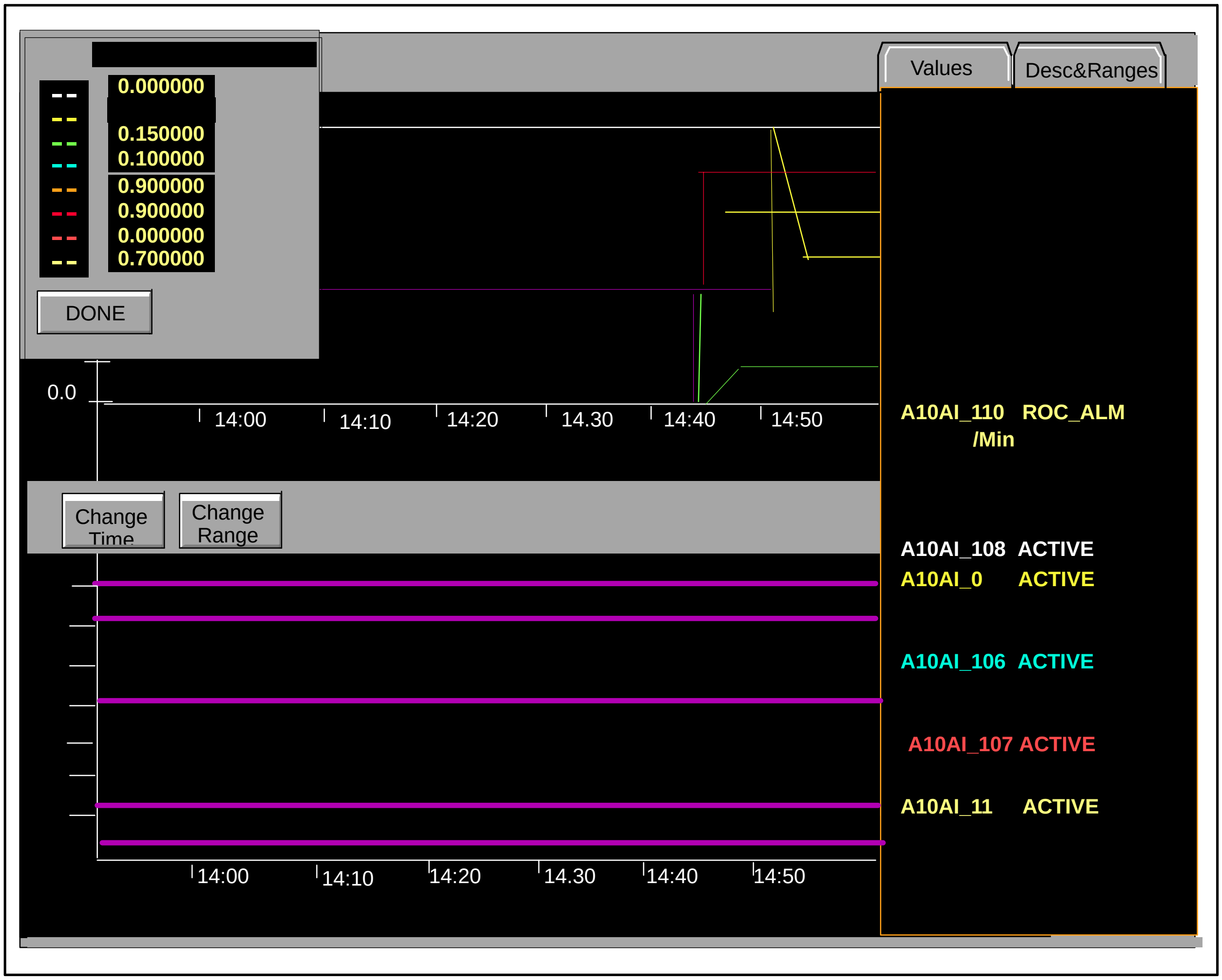


Figure 8-7 Historical Trend Display

Doubling the Size of Trend Displays

To configure trends to display at double the standard size, follow the steps below:

1. Open an XTerm window and log in as `tistar`.

2. At the command line prompt, type `ch_trends.sh 2page` and press **Enter**.

All trends will now display at double the standard size (Figure 8-8). The trend display background color is changed from black to light gray for more efficient printing, and a button is added that allows you to toggle from **Digital** to **Analog** trends.

To return to the standard display size, repeat the steps above, substituting `1page` for `2page` in the `ch_trends.sh` command.

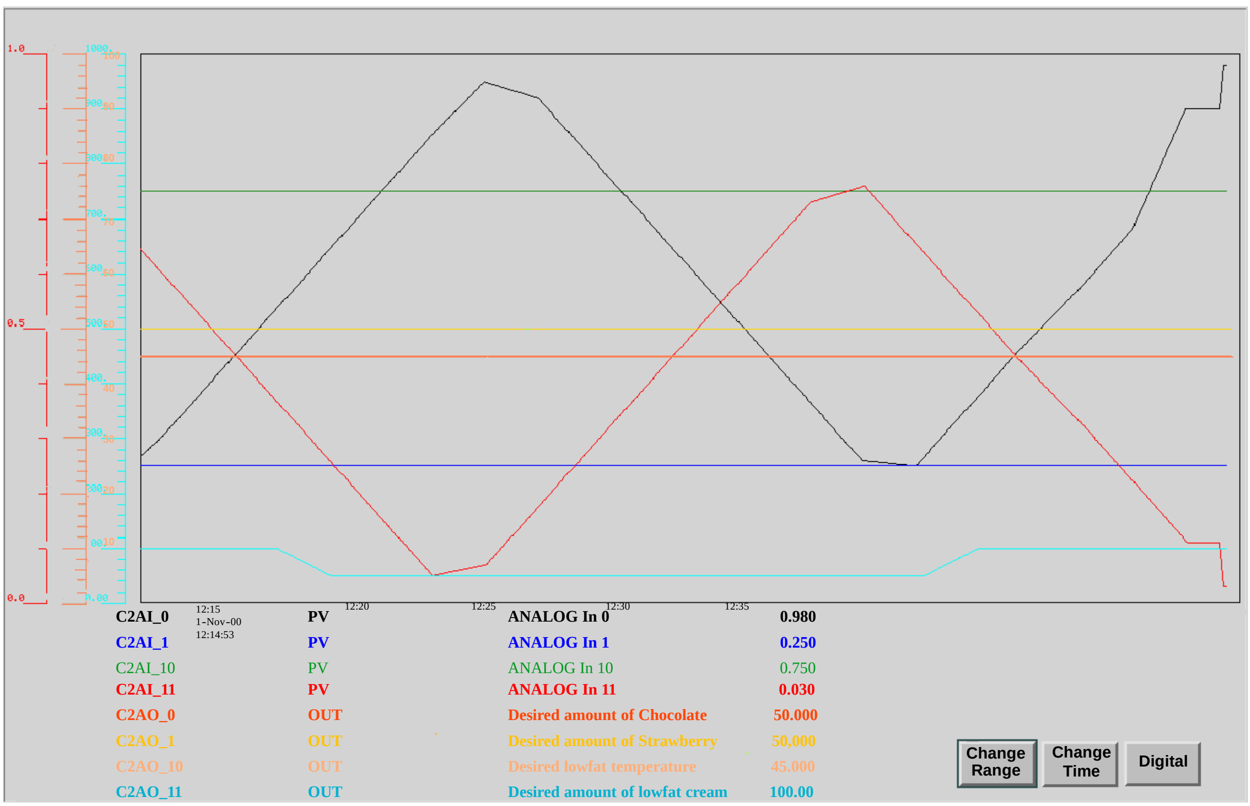


Figure 8-8 Double-Sized Trend Display

Zooming in on a Trend

To zoom in on a trend, you can drag a viewing area using the cursor. Move the cursor into the graph area, placing it at one corner of the drag area and press the middle mouse button. Drag a viewing area and click the middle mouse button again. A box containing an enlargement of the drag area appears (Figure 8-9).

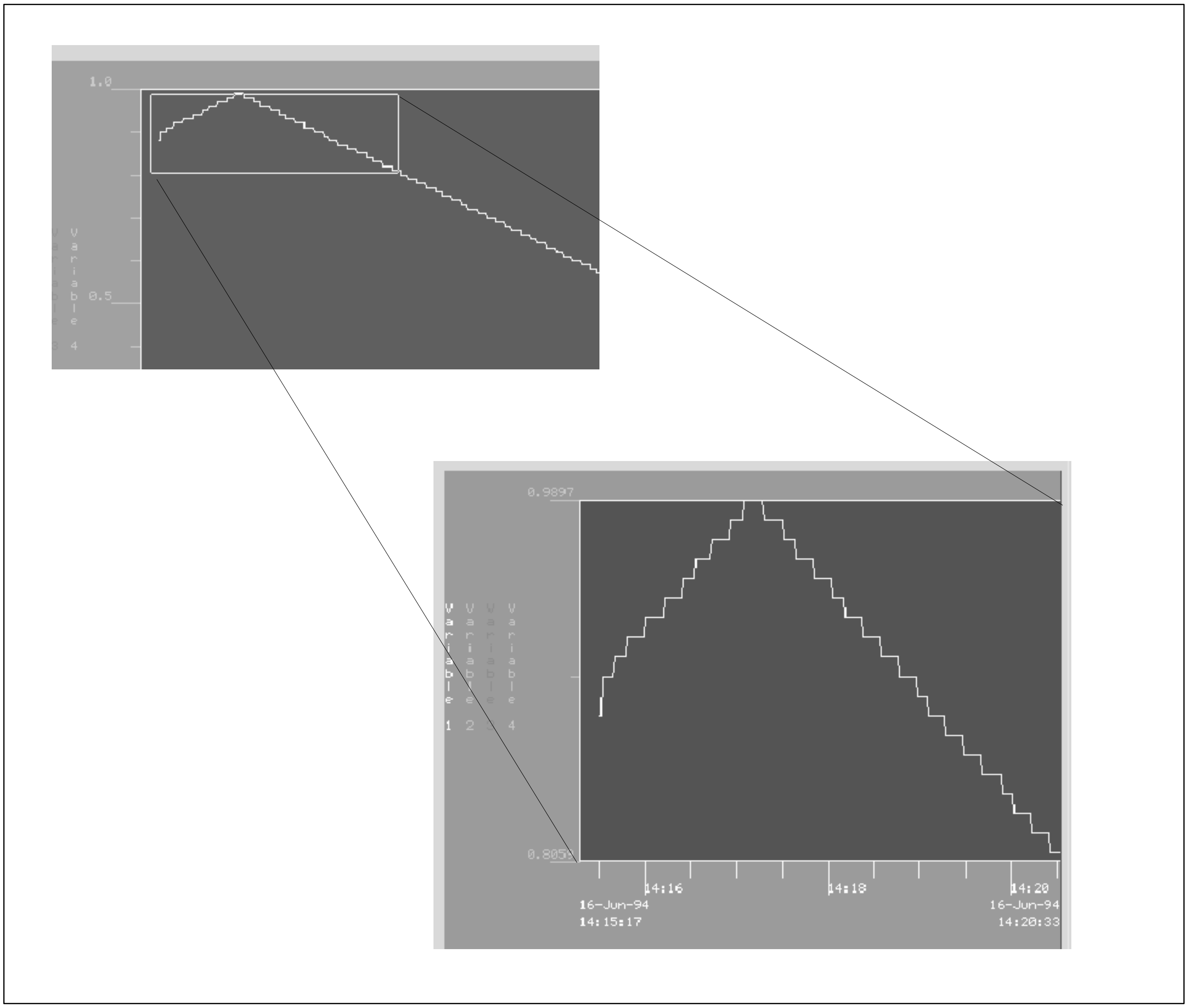


Figure 8-9 Zooming in on a Trend

Changing the Time Base or Range

Trend displays enable you to change the time base or range. The change is not permanent and only alters the time base or scale temporarily for the current display. When you select the **Change Time** or **Change Range** target, a command form appears in the lower left part of the screen (Figure 8-10). This form enables the operator to alter the values of the X axis (time) or Y axis (range). **OK** enters the new value and dismisses the command form, **Apply** enters the new value but leaves the form up, and **Cancel** dismisses the command form with no changes.

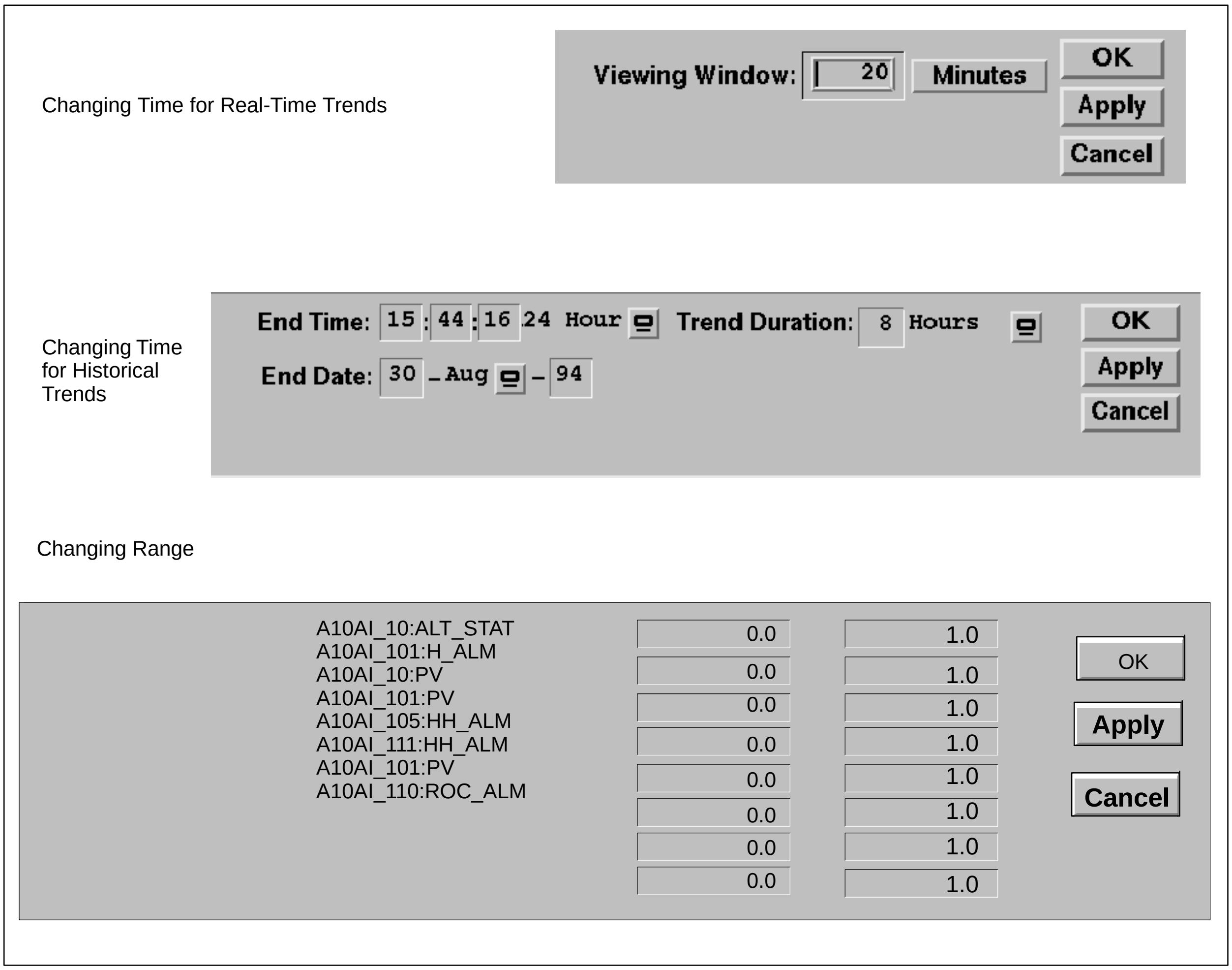


Figure 8-10 Time/Range Selection for Trends

Refreshing a Trend

If an OSx Terminal, an Xterm window, or the Diagnostics window is displayed and subsequently dismissed while displaying historical or real-time trends, portions of the trend window may not refresh. If this happens, you can refresh the trend by using any one of the following methods:

- Press **ALT-r**.
- Select **Refresh** from the root menu. (The root menu is displayed with a left-button mouse click on the root window. The root window is accessible from the bottom right gray region on the OSx screen.)
- Place the cursor in the trend window and press **ALT-F2**. This method is quicker because it redraws only the trend window. The other OSx screen areas are unaffected.

If these methods are inconvenient, a means of changing the OSx redraw operation to full screen redraw is available. Enabling full screen redraw results in a noticeably slower graphic redraw when a command form or configuration dialog is dismissed. Graphic redraw may be even slower on X Terminals or older OSx hardware.

To enable full screen graphic redraw, follow these steps:

1. Click **Control->OSx Terminal** from the main OSx menu bar.
2. Log in as tistar.
3. At the command prompt, type `refresh on` and press **Enter**.
4. At the command prompt, type `exit` and press **Enter**.

To disable full screen redraw, repeat the steps above, but enter the following command in step 3: `refresh off`

8.6 Displaying Trend Data in a Graphic

You can display one trend in a graphic display. Configure the trend as described in Section 8.1. Place one of the four following Trend DDOs as appropriate in the graphic:

- Historical analog DDO
- Historical digital DDO
- Real-time analog DDO
- Real-time digital DDO

Refer to the *SIMATIC PCS 7 OSx Graphical Editor Manual* for information about creating graphics.

When the system is in the Operate state, the operator can display the graphic and observe the trend plot. The operator can click the trend and display a box that shows exact values at the selected time (Figure 8-11).

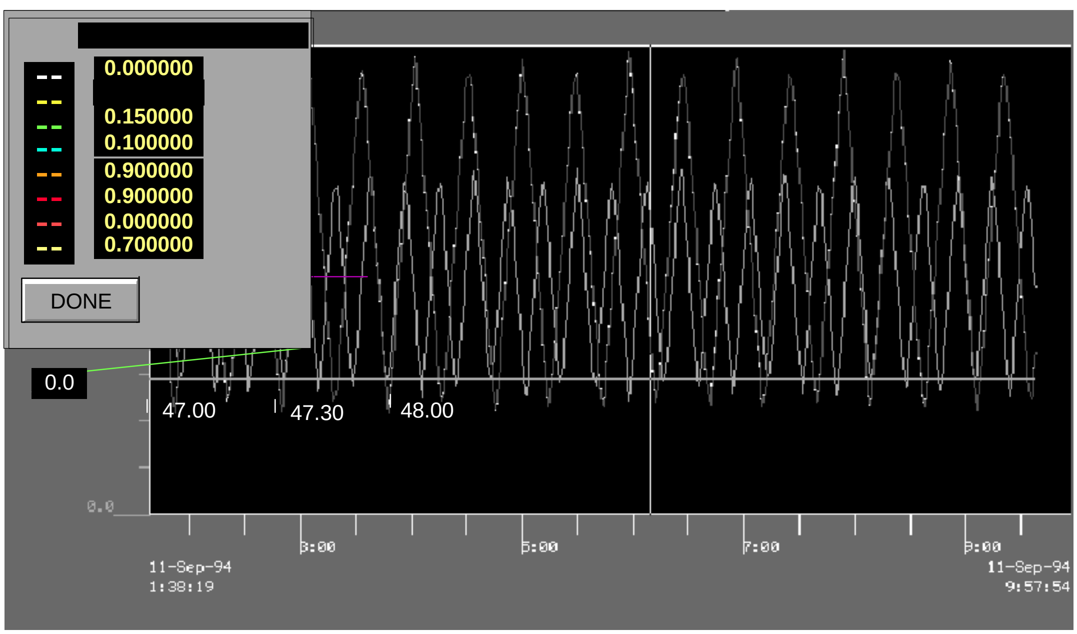


Figure 8-11 Displaying Trend Data in a Graphic

Chapter 9

Guidelines for Configuring Alarms

9.1 Alarm System Overview

When you configure a process I/O tag, you link it to a memory area of the control node (Figure 9-1). The controller provides alarms to allow you to monitor potential upset conditions in your process. These alarms are an integral part of the loop and analog alarm functions. You can also create an alarm that is based on discrete events.

When you configure an alarm system in OSx, you are creating an extension of the alarms in the controller program and have the capability of manipulating them.

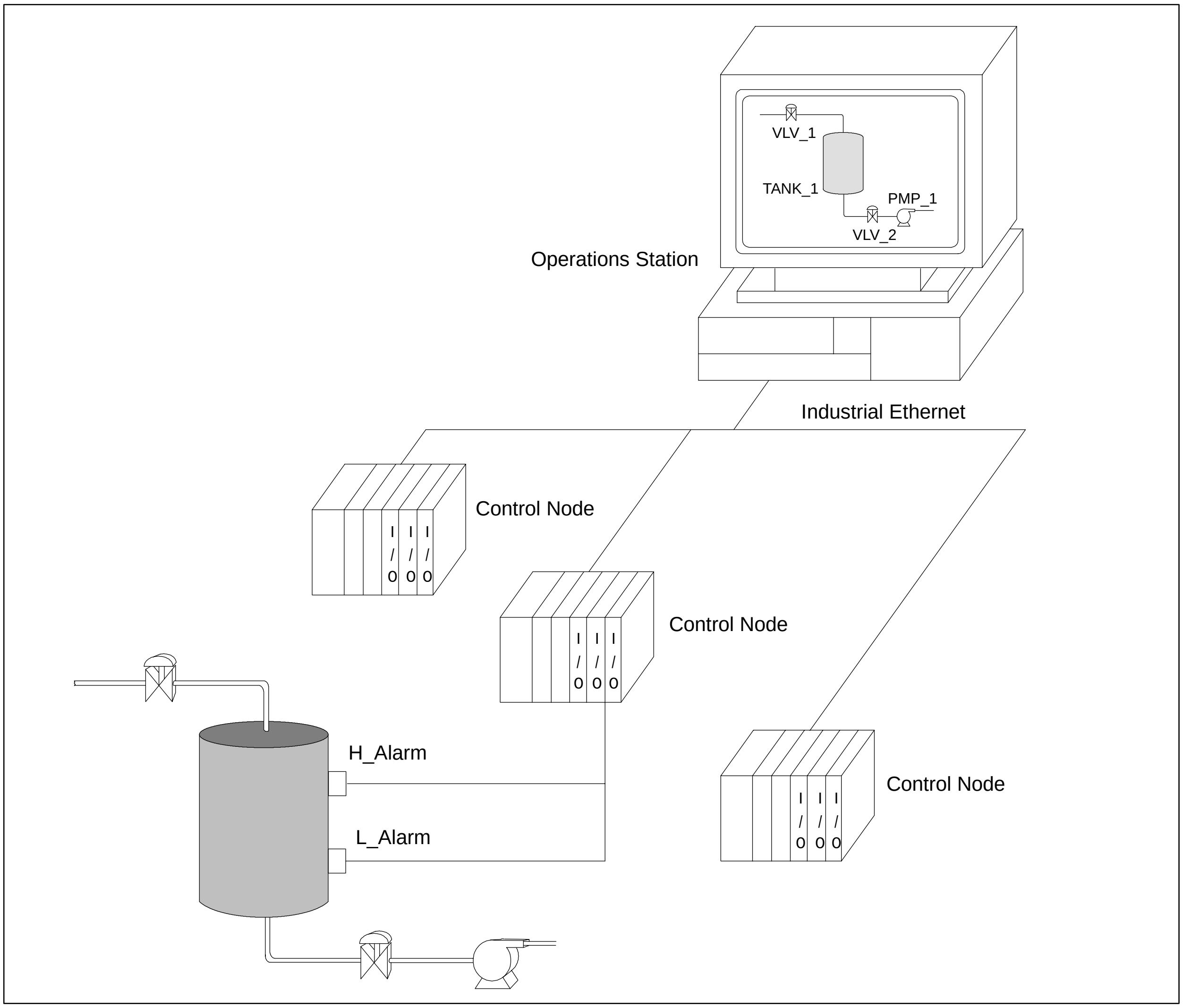


Figure 9-1 Example of Alarm System

Key Features of the Alarm System

You can use OSx alarm capabilities as you configure the system to provide operators with information about process upsets, to avoid nuisance alarms, and to devise efficient means of responding to alarm conditions.

The following features can significantly reduce the number of nuisance alarms that often overwhelm operators with alarm messages during periods of critical process upset. This chapter describes these three features:

- Alarm priorities by type of alarm
- Automatic alarm enabling and disabling by process state
- Automatic alarm suppression of related alarms

To take advantage of these features and achieve the desired results, you must have a thorough understanding of your process and the interrelated dynamics of specific areas of that process. You must also have some understanding of your OSx configuration. For this reason, you may want to begin by using the preconfigured alarm system defaults.

Alarm Priorities

SIMATIC PCS 7 OSx allows you to set alarm priorities on a sub-tag basis. For example, on analog input points you may decide to designate High High Alarms and Low Low Alarms as critical, while designating High Alarms and Low Alarms as warning for the same I/O points. Similarly, you may decide to designate High High Alarms and High Alarms as critical, Low Alarms as informational, and Low Low Alarms as warning for your plant operation.

Table 9-1 indicates the default priority level for each alarm type. Before changing the default priorities, ensure that you understand the operational details of each priority level and the differences in behavior exhibited by each.

Table 9-1 Default Alarm Priorities for Alarm Types

Tag Type	Alarm Type	Description	Default Priority
AI/LOOP	HH ALARM	High High	Critical
	H ALARM	High	Warning
	L ALARM	Low	Warning
	LL ALARM	Low Low	Critical
	ORANGE DEV	Orange deviation	Critical
	YELLOW DEV	Yellow deviation	Warning
	BAD XMTR	Bad transmitter	Information
	ROC ALARM	Rate-of-change	Warning
DI/DO	DATA VAL	Data Value	Information
VLV1	NOT OPEN	Closed	Critical
	NOT CLOSED	Open	Critical
VLV2	NOT OPEN	Closed	Critical
	NOT CLOSED	Open	Critical
	FAILED	Failed	Critical
MTR1	NOT RUNNING	Stopped	Critical
	NOT STOPPED	Running	Critical
RMTR/MTR2	NOT RUNNING	Stopped	Critical
	NOT STOPPED	Running	Critical
	FAILED	Failed	Critical

Table continued on next page.

Table 9-1 Default Alarm Priorities for Alarm Types (continued)

Tag Type	Alarm Type	Description	Default Priority
DI10/DO10	DVAL1	Data Value 1	Information
	DVAL2	Data Value 2	Information
	DVAL3	Data Value 3	Information
	DVAL4	Data Value 4	Information
	DVAL5	Data Value 5	Information
	DVAL6	Data Value 6	Information
	DVAL7	Data Value 7	Information
	DVAL8	Data Value 8	Information
	DVAL9	Data Value 9	Information
	DVAL10	Data Value 10	Information
System	PRT FAIL	Printer failure	Warning
	COM FAIL	Communication failure	Information
	H ALARM	High Alarm	Warning
	HH ALARM	High High Alarm	Critical
	INFO	Information	Information
	DEGRADED	Degraded	Warning
	FAILED	Failed	Critical
AREA*	BLK MISS	Mismatched RCBs	Warning
	REQ ABT	Request abort	Warning
	ERR ABT	Error abort	Warning
	TIMEOUT	Time out	Warning
	SEC ABT	Secondary abort	Warning
UNIT*	ABORT	Unit is holding	Information
	ALARM	Unit is in general alarm	Information
	HOLD	Unit is holding	Information
	WAIT	Unit is waiting	Information

* For more detailed information on the AREA and UNIT tags, refer to the *SIMATIC PCS 7 OSx Recipe* and *Batch Programming* manuals.

Disabling Alarms

You can configure the alarm system to enable or disable specific alarms automatically for each process state that you have defined. On operator command, you may also put an alarm group or alarm tag out of service. When you disable an alarm, the alarm shuts off for the duration of a particular process state, or until you manually enable the alarm. A message prints on the alarm log to notify the operator that the alarm has been disabled.

For example, the mixer shown in Figure 9-2 has the following process states.

Process state 1: Filling the vessel (loading liquid materials).

Process state 2: Mixing materials.

Process state 3: Discharging the vessel (draining the mixer).

During alarm configuration, you can designate that the associated alarms be enabled or disabled as the operation progresses from process state to process state.

While the mixer is filling (process state 1), the High and High High Alarms are important to prevent overfilling the vessel; you must enable these alarms. However, you could disable the Low and Low Low Alarms. In this case, the alarm system would not generate alarms as a result of low levels of materials as the mixers are being filled.

During the mixing phase (process state 2), the High High Alarm is critical to prevent backflow into the incoming pipe; you must enable this alarm. The High Alarm could be a nuisance alarm due to the turbulence that is created by the mixing process. Therefore, you might want to disable the High Alarm on the mixers during process state 2, as shown in Figure 9-2.

When the mixers are discharging (process state 3) and the level of the materials is decreasing, the Low and Low Low Alarms would be nuisance alarms, and you could disable them.

Proper application of this feature eliminates spurious alarms that tend to undermine process alarm credibility and ensures that operators receive only alarm messages that are useful in the context of the current state of the process.

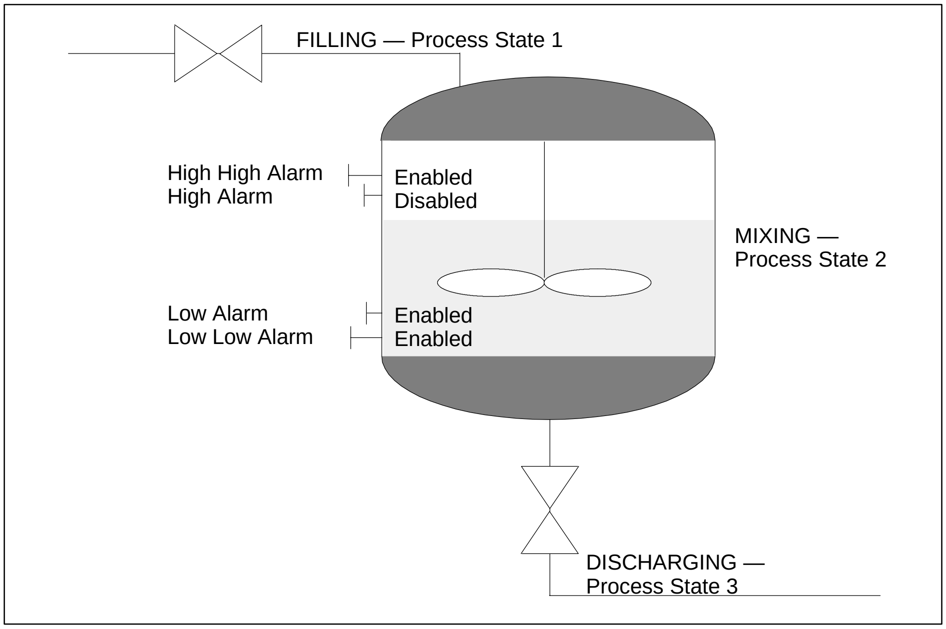


Figure 9-2 Example of Alarm Disabling by Process States

You can disable alarms in any one of the five following ways.

- From APT, you can disable alarm bits individually by marking the box in the alarm monitoring section of the CFB, especially for loop and analog input tags. The alarm bits that are typically disabled are Orange_Dev (H_Dev), Yellow_Dev (L_Dev), ROC, and Bad_Xmtr (broken transmitter), but you should inspect each tag individually to determine the bits to disable. If you use broken transmitter alarms, be sure to maintain proper calibration of the signal so that the alarm truly represents a broken transmitter. If the signal is not properly calibrated, nuisance alarms can occur.

  Keep in mind that when you disable individual alarm bits in APT, they are permanently disabled in all process states in OSx; they cannot be re-enabled in OSx for alarming.

- From the Alarm Group display, you can disable an entire alarm group. Toggle the Group Is Enabled/Disabled field (page 9-25). All the alarms in that alarm group are disabled until the group is explicitly re-enabled.

- From the tag detail for an alarm tag, you can disable a single alarm tag within an alarm group. Click the Alarming field in the General Alarming Information box on the tag detail (Figure 9-3, top). A command form appears at the bottom of the screen that allows you to disable the tag. The tag remains disabled until that field is toggled back to the value **Enabled**.

- In the Operate state, you can disable individual alarm bits within a tag from the tag detail. Toggle the value in the State column field for a particular alarm (for example, H_Alarm) to **Disabled** (Figure 9-3, bottom). When the system changes to Offline and then back to the Operate state, the alarm retains the new setting.

- In the Offline state, you can disable individual alarm bits within a tag from the Alarm Tag Configuration screen (page 9-16). When you enable or disable alarms from Alarm Tag Configuration, the changes are reflected in the tag detail, and vice versa. For example, if you disable the LL_ALARM in a tag detail, the change also shows up in the Alarm Tag Configuration screen for that tag.

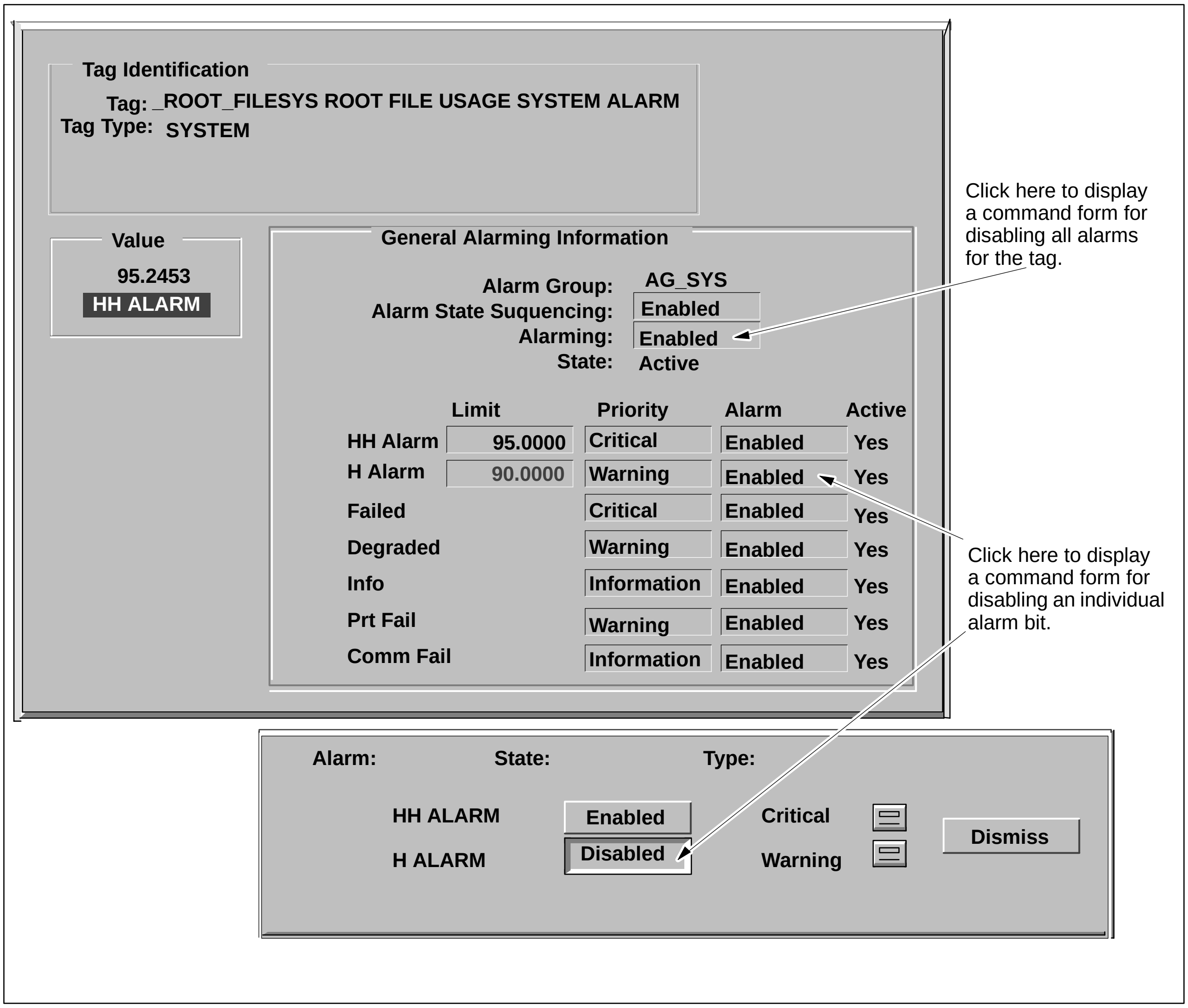


Figure 9-3 Disabling Alarms from a Tag Detail

Suppressing Alarms

Alarm suppression provides an alternate means of eliminating nuisance alarms. You can configure the alarm system to suppress alarms that, because of equipment layout, cause the system to generate other alarms. Alarm suppression causes the system to generate an alarm for only the most important cause of a fault condition. This also eliminates the situation of an operator having to deal with too many alarm messages at one time.

If you suppress an alarm for a tag, a bit in the status attribute is masked so that alarm conditions that are retrieved from the controller are ignored. You cannot use the bit that is masked in the status attribute of one tag to suppress an alarm in another tag.

WARNING

Suppressing or disabling an important alarm could result in death or serious injury to personnel and/or damage to equipment.

Make sure that an alarm is really not needed before suppressing or disabling it.

Furthermore, do not suppress or disable a Unit tag alarm when a batch program references the Unit tag status bit to determine the status of the unit. Suppressing or disabling the alarm can lead to incorrect operation of the batch program.

Figure 9-4 shows a typical example of how to use alarm suppression. Alarms are configured for the following tags: the pump, **P-601**, and the level indicator control on the large vessel, **LIC-601**. If the pump stops operating, the large vessel eventually drains out. Receipt of an alarm on the pump failure is key to returning process conditions to normal; the fact that the level in the vessel decreases is merely a side effect of the pump failure. Therefore, in certain process states, you may want to suppress the Low and Low Low Alarms for the level indicator control loop, **LIC-601**.

This scenario reduces the number of alarms to be acknowledged and allows the operator to concentrate on the primary cause of the problem.

NOTE: Do not allow alarm suppression tags to suppress each other such that they form a circular relation. That is, do not have tag 1 suppress tag 2, and have tag 2 suppress tag 3, and have tag 3 suppress tag 1.

If you have one alarm tag suppressing another, and then disable the first tag, then you have removed the suppression capability that you intended.

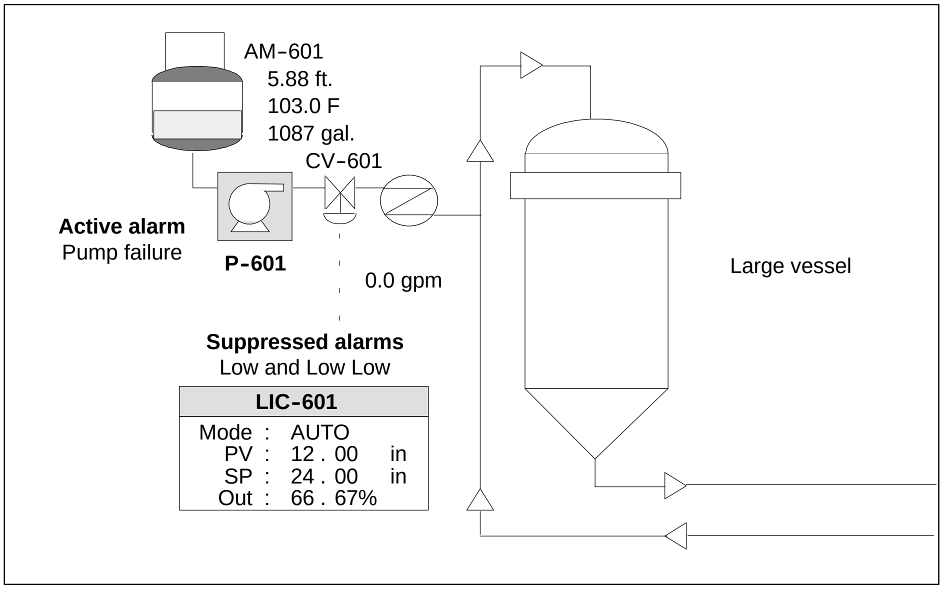


Figure 9-4 Example of Alarm Suppression

System Performance and Monitoring Device Alarms

All OSx tags have a status attribute that contains both alarm and current state information about the tag. For example, the status attribute for device tag VLV2 contains the current state of the tag (open or closed), as well as failure information.

Whenever a device changes state or fails, the controller transmits an RBE event message to all OSx stations. If you have configured the device tag for alarm monitoring, the alarm monitoring package examines each RBE event for a potential alarm condition. The alarm system must note failure conditions but ignores state transitions, such as a valve’s opening or closing.

System performance depends in part on the rate that RBE events occur. RBE events that occur at a sustained high rate can adversely affect the performance of the OSx stations. To avoid a high sustained rate of RBE event generation because of devices that change state frequently, your approach for monitoring device alarms needs to be different from the one that you use for other tag types. Consider the example below.

You want to monitor for alarm purposes the Failed bit (0x0010) of a VLV2 tag type. To do this, create a DI tag type and assign its Data Val bit (0x8000) the same controller address used for the VLV2 Failed bit. For alarm monitoring, use the DI tag instead of the VLV2 tag. No additional RBE events occur when the valve opens or closes, but the system reports a device failure if the VLV2 Failed bit becomes true.

9.2 Components of the Alarm System

Tag with Required Alarm(s)

Operators monitor plant operations by viewing process graphic displays. Process I/O tags that are configured to animated and non-animated symbols on graphics displays provide information on the status of operations. You can configure alarms for previously configured AI, DI, DI10, DO, DO10, AREA, DEVICE, LOOP, and UNIT tags. By default, all system tags are preconfigured for alarms. The status attribute of these system tags provides alarms to the OSx station. See Table 9-5 for a complete list of system alarms and the conditions that cause them.

NOTE: Because the system tags configured in OSx are crucial to system operation, you cannot delete a tag beginning with "_" from the system alarm group. If you add a tag that begins with "_" to the system alarm group, you cannot delete it. However, you can delete all other tags that you add to the system alarm group and all "_" tags in other groups.

Group State Tag

The alarm system refers to the steps in a process or batch as states. A group state tag is a tag that keeps track of the state that the process or batch is in. The group state tag can be any type of tag that provides an integer value. The alarm system uses the state number provided by the group state tag to disable and enable the appropriate alarms for that process state. If your application is a batch process that requires you to enable and disable alarms by process states (see page 9-6), you must program a group state tag.

You can configure alarm groups with just one process state for a constant alarming scenario, or with many states (up to 32) for a multiple state situation with different alarming combinations in each process state. The alarm system uses the integer value of the state number to enable/disable alarms in that process state according to your alarming plan.

Components of the Alarm System (continued)

Alarm Groups

During alarm configuration, you classify tags according to device or machine association or according to other common characteristics. You then assign them to alarm groups. The Unit Overview in Figure 9-5 shows an example of how you can divide a production unit into alarm groups. The alarm tags for the two mixers and their associated pumps, valves, and pipes are in the first alarm group. The reactor and associated pump, valves, and pipes are in the second alarm group. The evaporator and associated pump, valves, and pipes are in the third alarm group.

When an alarm occurs within the SIMATIC PCS 7 OSx system, the alarm group associated with that alarm becomes active, and the alarm system sends notification of the alarm condition to the OSx station. Alarm groups allow an operator to display and acknowledge alarms collectively instead of tag by tag.

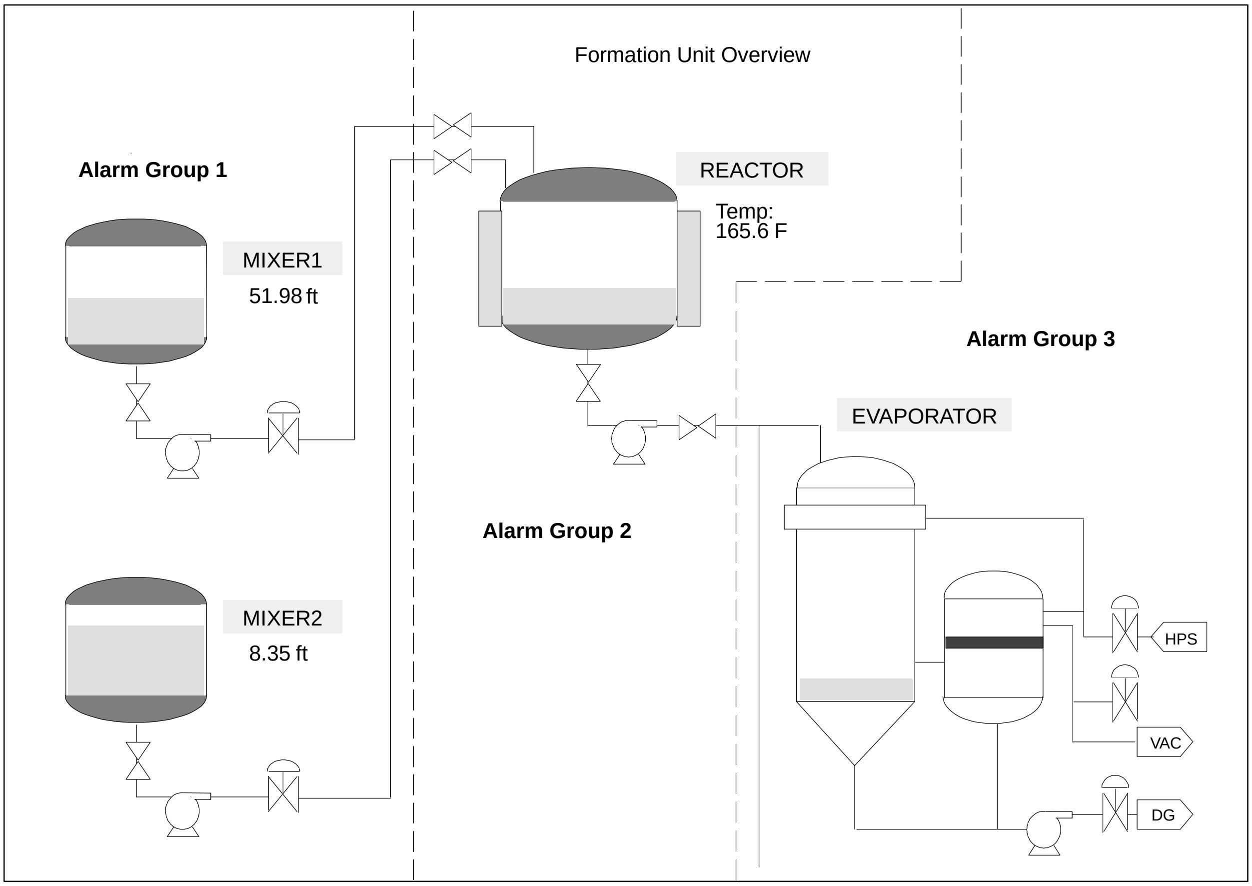


Figure 9-5 Unit Overview with Alarm Groups Identified

Alarm Tags

The status attribute of the process I/O tag that has been converted to an alarm tag provides the extension of the alarms from the controller through the tag to the alarm system. You can enable/disable the alarms that are provided in the status attribute of the alarm tag to fit the process. You can also suppress alarms based on an outside discrete event such as an emergency stop switch.

You must convert the process I/O tags with the required alarms into alarm tags in order to use them in the OSx alarm system. This is done by entering alarm data in the Alarm Tag Configuration display (Figure 9-6). Do not configure a tag for alarming unless its status attribute is networked (being scanned).

If your process requires a single (constant) alarming situation, you only need to enter alarm data for one process state. This is the default state ACTIVE that enables all alarms. The individual alarms of the alarm tag can be enabled/disabled or suppressed as needed.

If your process requires that you monitor process states in the controller program, you need to enter data for all the process states, specifying the alarms that are enabled and disabled.

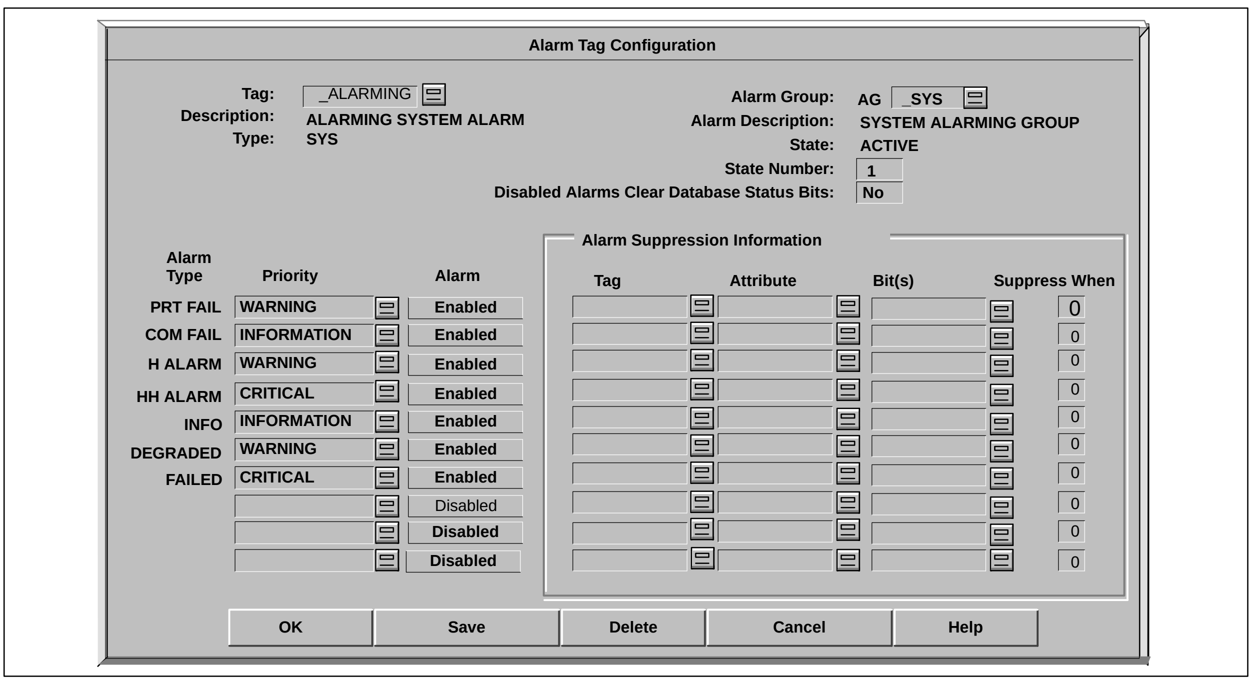
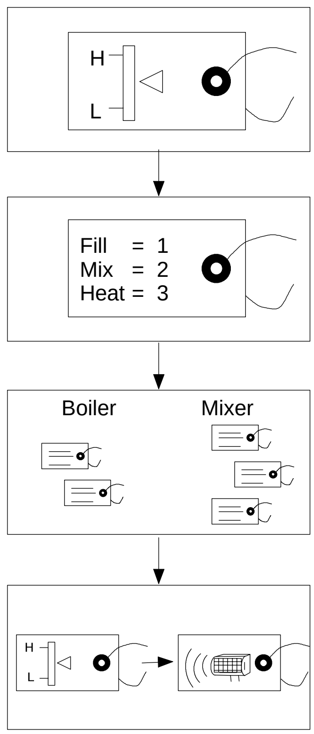


Figure 9-6 Alarm Tag Configuration Display

Where to Begin...

Configuration of the alarm system consists of four steps. For an orderly and efficient configuration, follow the steps shown in Figure 9-7 in the order suggested.



Create Process I/O Tags with the Required Alarms (Chapter 5)

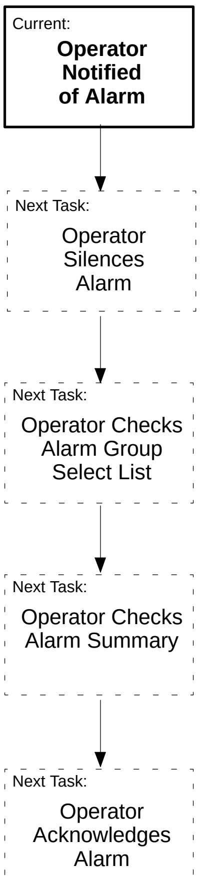
Configure Group State Tag (Section 10.2)
Optional. Only needed when monitoring process states.

Build the Alarm Group (Section 10.3)

Convert Tags to Alarm Tags (Section 10.4)

Figure 9-7 Order of Alarm Configuration

Alarm Notification and Acknowledgement



When a process upset occurs while OSx is in the Operate state, the operator receives both a visual and audible indication of the alarm at the OSx station. A dual triangle symbol, an alarm icon, appears in the lower right part of the screen, as shown in Figure 9-8. At the same time, an audible alarm sounds.

The color coding of the alarm icon and the tonal quality of the audible alarm indicate alarm priority. Table 9-2 shows the default alarm priorities and the abbreviations for each. Within OSx, critical is the highest priority alarm, and maintenance is the lowest priority alarm.

Table 9-2 Alarm Priority Indicators

Alarm Priority	Color Code	Alarm Tone
Critical (C)	Black on Red	Rapid; high-pitched
Warning (W)	Black on Yellow	Slow; low-pitched
Information (I)	White on Blue	None
Maintenance (M)	Printed Only	None

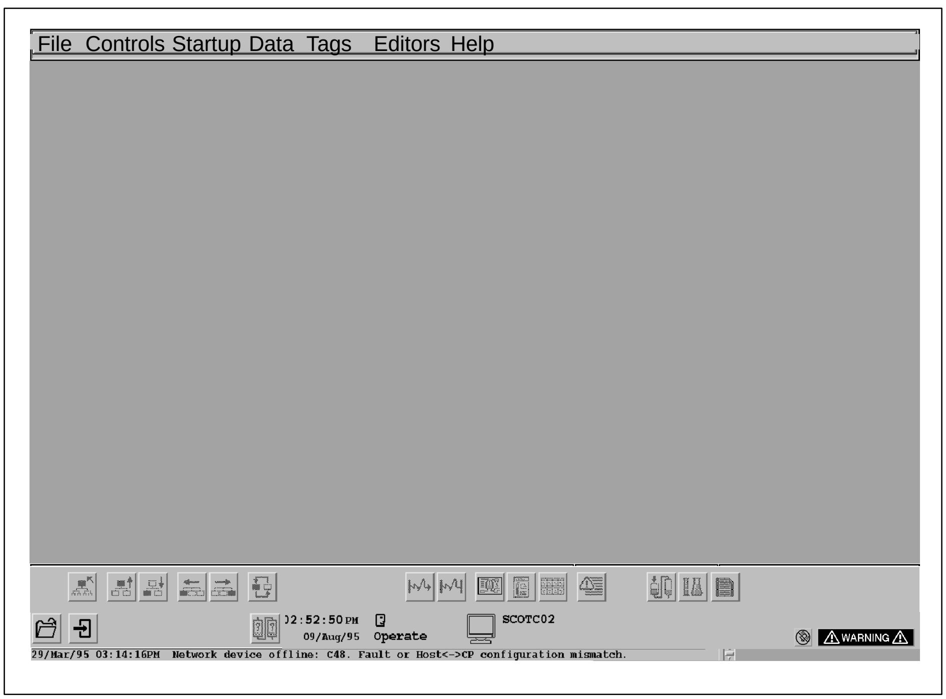
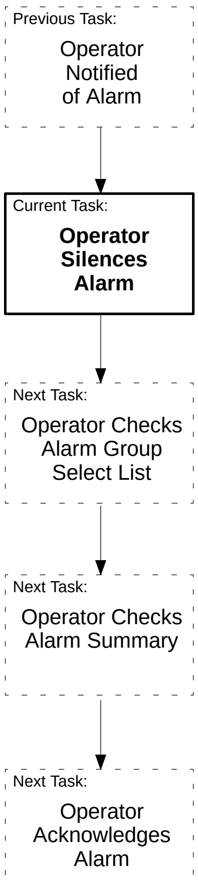


Figure 9-8 Location of Alarm Icon



In order to acknowledge an alarm, the operator must have one of the following security privileges: Database Administration, System Configuration, Primary Control, Secondary Control, Tertiary Control, or Operations.

The first step in the acknowledgment process is to stop the sound of the audible alarm. To silence the alarm, the operator presses **Alarm Silence** on the operator keyboard, or clicks on the silence icon (Figure 9-9). The audible alarm remains silent until the next alarm occurs.

After silencing the alarm, the operator needs to check the list of active alarms. This is described in the next section.

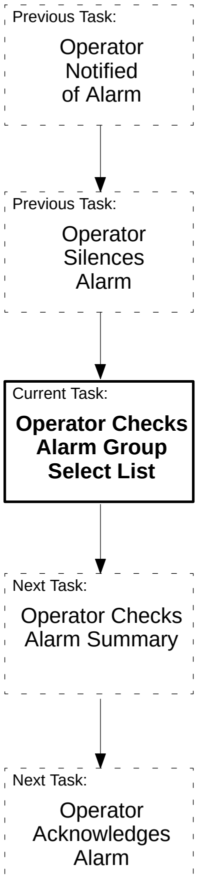
Figure 9-9 Alarm Silence Icon

WARNING

When some windows appear, they force all mouse and keyboard control to the window until the operator makes a selection from the window.

This means that the operator cannot respond to alarms that occur while the window is open, and consequently there is potential risk of death or serious injury to personnel, and/or damage to equipment.

Make sure that the operator knows that he must make a selection from the window before responding to an alarm.



The alarm system ranks all active alarm groups and displays the names of all alarm groups with active alarms in the Alarm Group Select list. The names in the Alarm Group Select list have the same color coding as the alarm icon.

To check the list of alarms, the operator moves the cursor to the flashing alarm icon on the screen and clicks on it. This causes the Alarm Summary display to appear (Figure 9-10).

The Alarm Group Select list on the left of the Alarm Summary display lists the alarm groups in order of importance from top to bottom. Alarm group importance is based on the following factors.

- The number of in-alarm conditions in an alarm group
- The number of unacknowledged alarms in a group
- The priority levels of the alarms in a group

For example, a group with two unacknowledged critical alarms is more important than a group with two unacknowledged warning alarms. When the operator clicks on the alarm icon, the Alarm Summary automatically displays the highest priority alarm group. To view alarms from another group, the operator clicks on that group in the Alarm Group Select list.

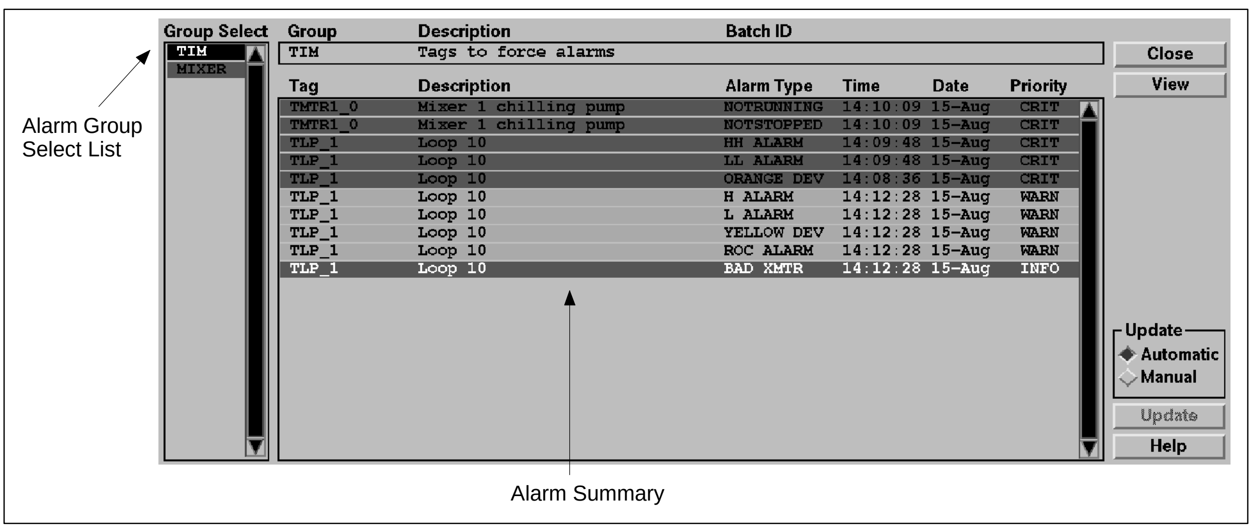
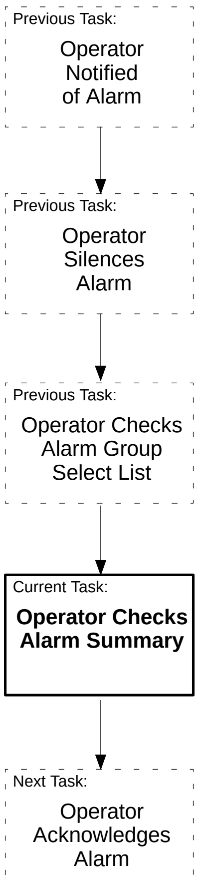


Figure 9-10 Alarm Summary Display



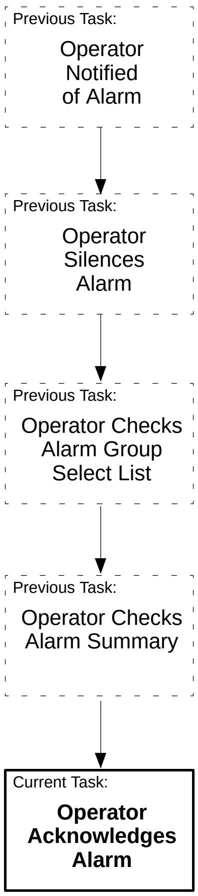
If you want to view an associated display, such as a graphic or tag detail, for a particular alarm group, you can configure that display in the Alarm Group Configuration dialog box (Section 10.3). If you have configured an associated display, the operator can view this display by selecting the **View** pushbutton.

The Alarm Summary lists all the active alarms in an alarm group, including critical, warning, and information alarms. The list provides the following information: the time that the alarm occurred, the name of the alarm tag, the tag description, the alarm type, and the alarm priority. If there is a batch name associated with the alarm group state tag, that name appears in the Batch ID field. This field has information only if there is an alarm group state tag.

The **Automatic** update mode updates displayed information automatically at preset intervals. The default is 10 seconds. You can change the default in the Event Preferences dialog box (select **Startup->Event Preferences** on the menu bar). In the **Manual** update mode, the operator must press the **Update** button for the screen to display new alarms and to remove obsolete ones.

Active alarms for the group are listed by priority and by time of occurrence within priority. For example, all critical alarms appear before warning alarms; among critical alarms, the most recent alarm is listed first. Each alarm on the Alarm Summary is highlighted in the same color coding as the alarm icon until you acknowledge the alarm.

The next section describes alarm acknowledgment procedures.



Acknowledging Alarms Priority levels determine the method an operator must use to acknowledge alarms from the Alarm Summary. Table 9-3 indicates the acknowledgment procedure required for each alarm priority level.

Table 9-3 Alarm Acknowledgment by Priority

Priority	Acknowledgment
Critical	One-by-one
Warning	All at once
Information	None — listed in the Alarm Summary
Maintenance	None — recorded in Alarm Log

The alarm information and alarm icon remain on the screen until the following actions are taken.

1. The operator corrects the alarm condition.
2. The operator acknowledges the alarm on the Alarm Summary.

   To acknowledge alarms from the Alarm Summary, the operator can either click the alarm summary message line or press **Alarm Ack** on the operator keyboard.

The alarm icon (Figure 9-11) at the bottom of the screen continues to blink until all critical and warning alarms are acknowledged. The icon leaves the screen only when all critical, warning, and information alarms clear the system and process conditions return to normal.

Figure 9-11 Alarm Icon

Acknowledging Non-Networked Alarm Tags If a tag configured for alarming has a non-networked status attribute, such as a recipe area tag, and the tag goes into alarm, the alarm unacknowledged bit of the status word remains set (=1) under the following conditions:

- The operator does not acknowledge the alarm, and
- The system transitions from the Operate state to the Offline state and then back to the Operate state.

The alarm unacknowledged bit is the 0x2 bit of the tag's status attribute. OSx sets this bit to one when the alarm has not been acknowledged. For networked status attributes, OSx sets this bit to zero if the alarm is not acknowledged before the system state changes from Operate to Offline and then back to Operate. See the chapter on defining recipe areas in the *SIMATIC PCS 7 OSx Recipe Manual*.

Alarm Group Display

The Alarm Group display lists alphabetically all tags within the corresponding alarm group. For each tag within the group, the highest priority, most recent alarm is listed. The Alarm Group display allows the operator to enable or disable the entire alarm group by toggling the Group is Enabled/Disabled field.

To view the directory of alarm groups, the operator must follow these steps.

1. Select **Directory** from the navigation area. A directory of display types appears.
2. Select the **Alarm Group** option. The screen displays a list of alarm groups (Figure 9-12).
3. Select the alarm group. The screen displays the alarm group you selected (Figure 9-13).

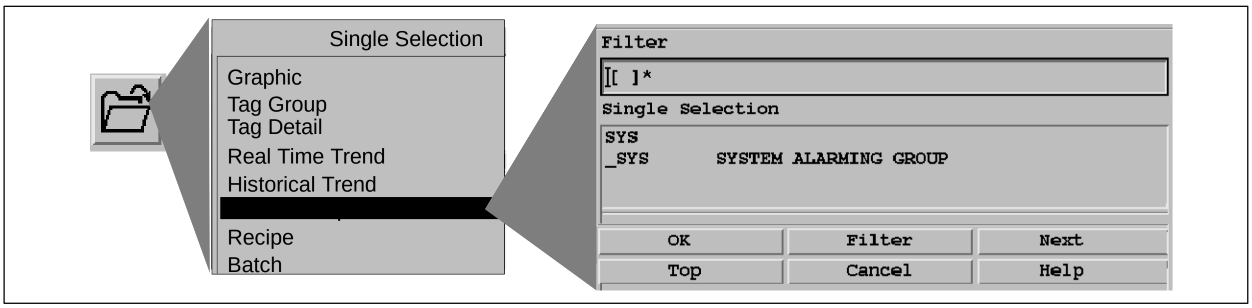


Figure 9-12 Selecting the Alarm Group Display

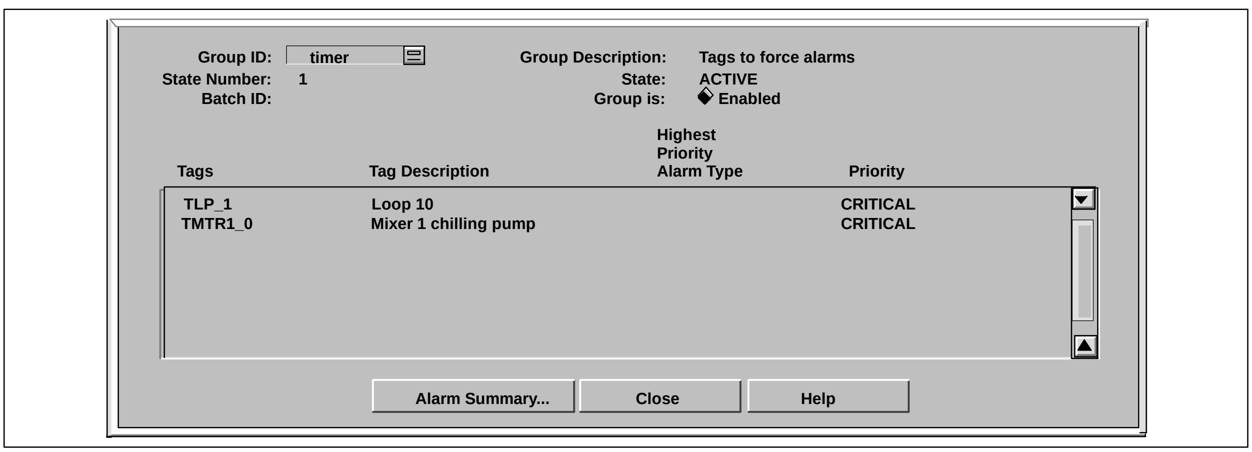


Figure 9-13 Example Alarm Group Display

The fields of the Alarm Group display (Figure 9-13) are described below.

Group ID Unique name for the alarm group. Use the short-list display tool in this field to display another alarm group from this dialog box.

Group Description Description of the characteristics or function of the alarm group.

State The current process state number.

State Description A name that describes the current process state.

Batch ID The name of the batch associated with the alarm group.

Group is Enabled/Disabled Toggle this field to enable or disable the entire alarm group.

Tags Names of tags that are assigned to this alarm group.

Tag Description Descriptions of tags that are assigned to this alarm group.

Highest Priority Alarm Type Highest priority alarm type configured for each tag within the group.

Priority Highest unacknowledged alarm priority for the tag.

Alarm Summary... Accesses the Alarm Summary display.

Alarm Mini-Windows

You can attach an alarm group to a graphic with the mini-windows feature. When you enable mini-windows from the Event Preferences screen, all active alarms in that alarm group are displayed in a list at the lower left of the graphic. Refer to Chapter 15 to configure mini-windows for an alarm group.

How the Alarm System Works (continued)

Alarm Log

The Alarm Log lists all alarms, regardless of priority. The Alarm Log is part of a standard OSx report called the System Log Report. When any alarm occurs, OSx records alarm data in the Alarm Log and sends the Alarm Log to a printer if you assigned one during printer configuration.

OSx also stores the Alarm Log with the Operator Change Log and system messages in a file called RPLOG.DDMMM-n (DD = day; MMM = month; n = number). This file is listed in the Reports Directory.

To view a report, the operator must follow these steps.

1. Select the **Directory** pushbutton from the navigation area. A directory of display types appears.
2. Select the **Report** option. The screen displays a list of reports (Figure 9-14).
3. Select the report. The screen displays the report you selected.

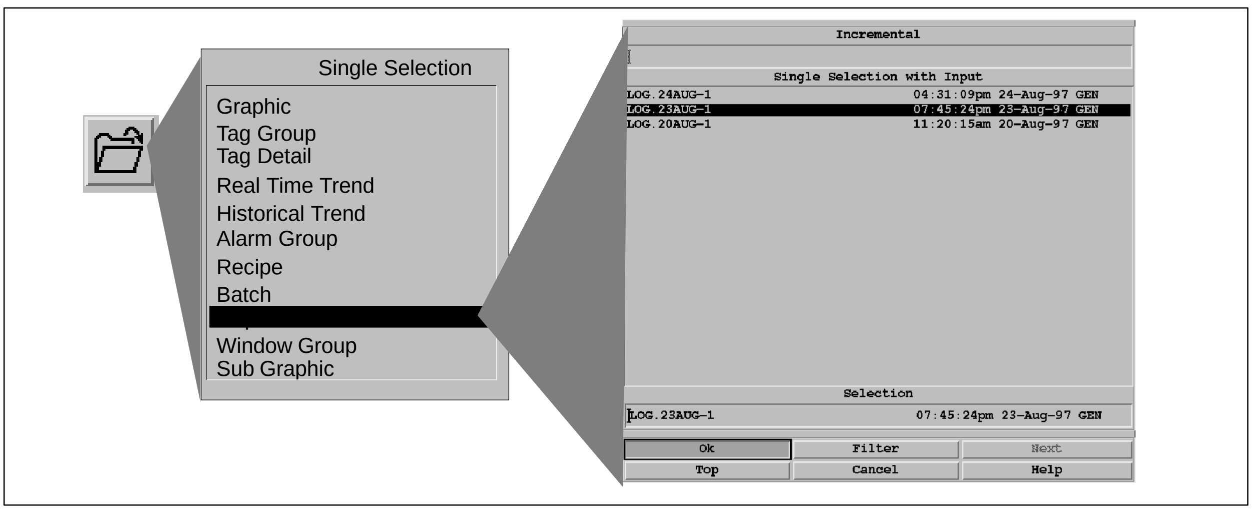


Figure 9-14 Report Directory Display

Table 9-4 lists the eight alarm log entry types and the four alarm priorities.

Table 9-4 Alarm Log Entry Types and Alarm Priorities

Log Entry Types		Alarm Priorities
IN = In Alarm	ST = Alarm Tag Enable/Disable	CRIT = Critical
OUT = Out of Alarm	SEQ = Alarm State Sequencing	WARN = Warning
ACK = Alarm Acknowledged	OOS = Out of Service	INFO = Information
PRI = User Priority Change	SUP = Alarms Suppressed	MAIN = Maintenance

Alarms and Station Synchronization

When the database is locked during OSx station synchronization, you need to be aware of the following points concerning alarms.

- The alarm icon and the silence icon appear on all stations except the primary and the station that is synchronizing.
- For new alarms, the alarm sounds on all stations except the primary and the station that is synchronizing.
- While the database is locked, you can silence alarms at any station except the primary and the station that is synchronizing.
- Alarms that occurred before the database was locked can be acknowledged from any station except the primary and the station that is synchronizing.
- You cannot acknowledge an alarm that occurs while the database is locked until OSx unlocks the database.

When the database is unlocked following OSx station synchronization, these events occur:

- The alarm icon and the silence icon appear on the primary for alarms that occurred during synchronization.
- The alarm sounds on the primary for alarms that occurred during synchronization.
- If an alarm tag has multiple alarms (for example, a DI10 having alarms 1–10), and some of the alarms are acknowledged while some are unacknowledged, an OSx station synchronizing in the Operate state will show all of the alarms for that tag as unacknowledged.

System Alarms

Table 9-5 provides a list of system alarms and describes the conditions that cause them. Note that a system alarm has the same name as the system tag with which it is associated. Alarms for system tags are annunciated only on the station where the alarm condition occurred.

Table 9-5 System Alarms

Alarm/System Tag Name	Alarm Conditions
_ACTREQ	Issued by the online alarm handling processes, for example, when errors occur on process initialization.
_ALARMING	Issued by the online alarm handling processes, for example, when the alarm icon form is removed by a non-alarming process.
_ARCHIVING	Issued by the data archiving processes, for example, because of auto archive failure, or disk full or nearly full.
_BATCH	Issued by the batch management process, for example, when commands cannot be sent to batches or units, or a BCL program abnormally exits.
_CHANGE_LOG	Currently not used.
_CONFIG	Currently not used.
_DATA_BASE	Issued by the database subsystem, for example, when backup database is restored after a failed tag capacity change or database corruption.
_DATA_XFR	Issued by the remote data transfer processes, for example, for local or remote database errors.
_DIAGNOSTICS	Currently not used.
_FO_CIRCUIT	Issued by the redundancy subsystem when a failover circuit is not connected.
_GENERAL	Issued to indicate problems of a general nature, for example, problems with handling keyboard input, or loss of communications with graphical editor.
_H1_COMM	Issued by the communications subsystem, for example, if controller offline, controller not responding, network connection down.
_HT_COLLECT	Issued by historical trend collection processes, for example, for collection falling behind schedule.
_HT_DISPLAY	Currently not used.
_HT_UPDATE	Issued by the historical trend manual update process, for example, when unable to open input file, invalid data, unconfigured trend points, disk is nearly full.
_LIBRARIAN	Issued by the librarian process, for example, for rcp failure, directory removal failure.

Table continued on next page.

Table 9-5 System Alarms (continued)

Alarm/System Tag Name	Alarm Conditions
_NODE_SYNC	Issued by the redundancy subsystem when the local station is no longer synchronized with the primary station (that is, its database and/or files are different).
_OPERATOR	Issued to indicate problems of an operator interface nature, for example, process is unable to attach to a keyboard key.
_OSX_FILSYS	Issued when the OSx file system **/usr/tistar** is nearly full. There are “high” and “high high” alarms.
_PAGE_SERVER	Currently not used.
_RECIPE	Currently not used.
_REDUNDANCY	Issued by the redundancy subsystem, for example, for error shadowing a database change, error shadowing a message, communications errors.
_REDUND_SYNC	Currently not used.
_REPORTS	Issued by the report compiler and report executables, for example, for arithmetic errors, error writing to networked points, or too many report outputs.
_ROOT_FILSYS	Issued when the root file system (/) is nearly full. There are “high” and “high high” alarms.
_RT_DISPLAY	Currently not used.
_SCHEDULE	Issued by the process execution scheduler, for example, for out of memory, or process already running.
_SECONDARY	Currently not used.
_SPOOLER	Issued by the printing subsystem, for example, for disk full, printer failure, or too many messages at once.
_SYS_CONTROL	Issued by the system controlling process, for example, for too many OSx processes, abnormal process termination, loss of 24 volt power source, or inability to communicate with other nodes or stations.
_TAG_DETAIL	Currently not used.
_TAG_GROUP	Currently not used.
_TAPE_DRIVE	Currently not used.
_TIME_SYNC	Currently not used.
_UNIX	Issued when there are problems interacting with the operating system, such as power loss.
_VERIFY	Currently not used.

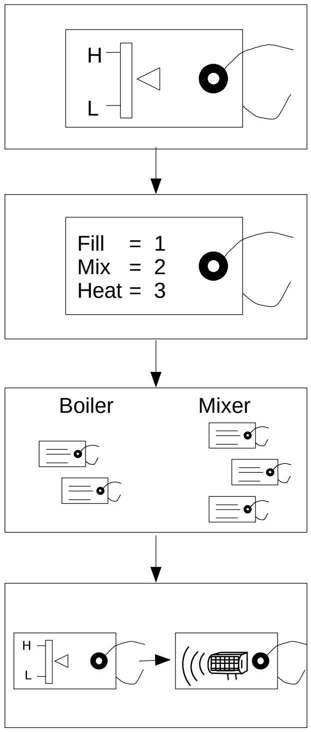
Chapter 10

Configuring the Alarm System

10.1 Planning Alarm System Configuration

The OSx alarm system is a comprehensive mechanism for monitoring potential upset conditions in your process. If you approach alarm configuration as a task that consists of four basic steps, you will find the procedure to be straightforward and simple.

Chapter 10 describes the four steps of alarm configuration in detail, and Figure 10-1 summarizes the process graphically. Review Chapter 9, which describes the theory of alarm configuration, and then when you begin to configure alarms, follow the steps in Figure 10-1 in the recommended sequence.



Create Process I/O Tags with the Required Alarms (Chapter 5)

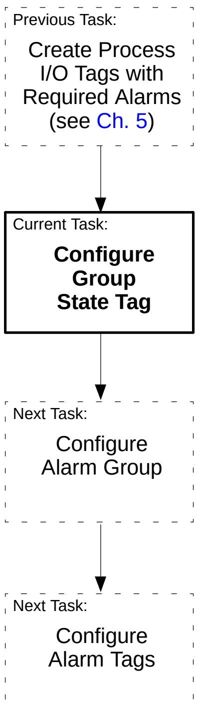
Configure Group State Tag (Section 10.2)
Optional. Only needed when monitoring process states.

Build the Alarm Group (Section 10.3)

Convert Tags to Alarm Tags (Section 10.4)

Figure 10-1 Order of Alarm Configuration

10.2 Configuring the Group State Tag (Optional)



In order to move the current state number from the controller program to the alarm system, you must configure a group state tag. Configure this tag following the same procedures as described in Chapter 5. Section 9.2 describes the uses of the alarm group state tag.

The group state tag can be any tag that is capable of providing an integer value for the process state number. You can use such tag types as the integer variable (IVAR) or counter (CTR). You must configure the group state tag before you configure the alarm group and the alarm tag.

You cannot delete a tag that is an alarm group state tag. This generates an error. Before you can delete a tag that is an alarm group state tag, you must first remove the tag from the alarm group in the Alarm Group Configuration dialog box. Then you can delete the tag from the system.

Table 10-1 shows an example of a configuration of a group state tag. The spreadsheet is 15 cells wide. The first eight cells are shown in the top half of the table; the last seven cells are shown in the bottom half.

Table 10-1 Example of Group State Tag Configuration

Columns 1-8							
Record	ControlNode	TagType	Tag	Description	ProcessGroup	ManualSet	Parent
T	Mixer	IVAR	Step.no	Mix_State	0x00000002		
A							
A							
A							
A							
A							

Columns 9-15						
Attribute	Memory	Location	Upload	Twenty%	Autolog	InitValue
STATUS	V	1	N	N	N	0
UNITS						Step
L_RANGE						1
H_RANGE						32
VALUE	V	2	N	N	N	1

10.3 Configuring the Alarm Group

Previous Task:
Create Process I/O Tags with Required Alarms

Previous Task:
Configure Group State Tag

Current Task:
Configure Alarm Group

Next Task:
Configure Alarm Tags

To configure an alarm group, you must have either the Database Administration or the System Configuration security privilege. You can add new alarm groups, and modify or delete existing ones while the system is in the Offline state. When the system is in the Operate state, you can add new alarm groups, and you can view, but not modify or delete, existing alarm groups. When you are configuring alarm groups, you cannot change system state.

Select **Data->Alarm Group** from the menu bar (Figure 10-2). Then fill in the fields described below. Configure process I/O tags and the group state tag (if needed) before you configure alarm groups.

Group ID Enter the alarm group ID, a five-character string used to identify associated alarm tags. If you previously configured the specified ID, all information pertaining to that alarm group appears on the screen. The space, the character combinations -> and <- and the characters ; \ , ” are invalid for the alarm group ID.

Group Description Enter the alarm group description, a 30-character string used to describe the function or characteristics of a group. The characters ; \ ” are invalid for alarm group descriptions.

Associated Display Enter the name of a display to be associated with the alarm group. Each alarm group has an associated alarm display. This is the display that appears when you select **View** from the alarm summary. The default associated display for each group is the alarm summary. To change the default, specify any of the displays configured within your system. If you choose a display from the select list, the system automatically precedes the display name with one of the two-character prefixes (Table 10-2). If you enter the display name manually, you must include the appropriate two-character prefix.

Table 10-2 Prefixes for Named Displays

Display Type	Prefix	Display Type	Prefix	Display Type	Prefix	Display Type	Prefix
Detail	TD	Historical Trend	HT	Real-Time Trend	RT	Alarm Summary	AS
Recipe Area	BT	Graphic	GR	Tag Group	TG	Alarm Group	AG

Group State Tag and **Attribute** Enter a tag name and attribute. The alarm group state tag is a tag point in the controller that indicates the current process state. The alarm group state tag must be an integer data type, so, for example, you can use an IVAR tag. For an IVAR tag, the alarm group state attribute is VALUE.

Default State Select one of the alarm states as a default, or use the default state of 1. The system assumes the default state under these conditions:

- You have not configured a group state tag, or
- The system reads a non-configured state number from the network (refer to the Alarm Group State Description on page 10-8).

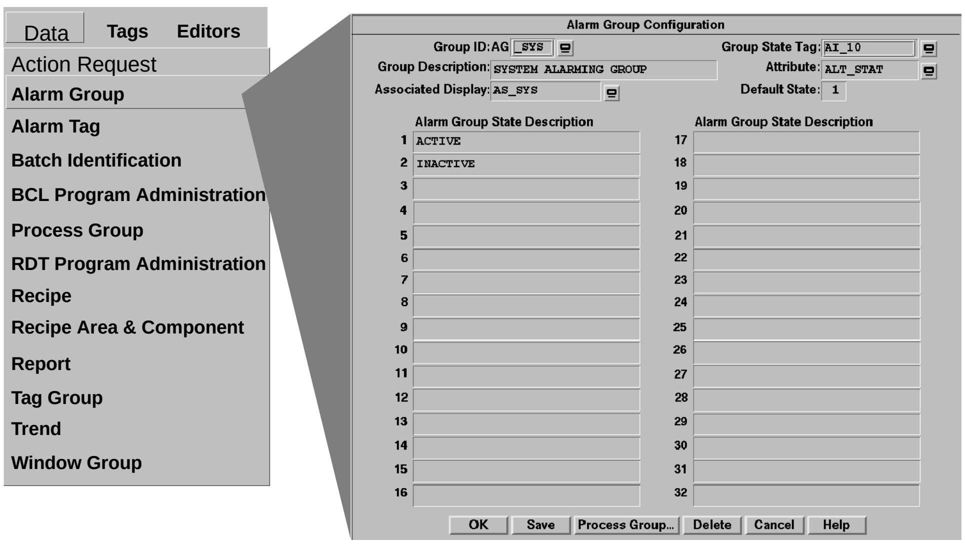


Figure 10-2 Alarm Group Configuration Display

Alarm Group State Description If you have process states, you can define them in these fields. The descriptions do not have to be entered in an order that reflects the sequence of your process, and the descriptions do not have to be unique. The characters ; \ ” are invalid for descriptions. The first two fields have predefined values of ACTIVE and INACTIVE that are OSx keywords. You can redefine these fields, but the INACTIVE keyword has a special effect in field 2 that does not apply when the term is used in any other field.

In the Alarm Group Configuration screen, the default value for field 2 under Alarm Group State Description is INACTIVE (Figure 10-2). When you configure an alarm tag in this group, as described on page 10-13, you can set the Group State Number to 2 to assign the initial value of Disabled to all alarms in the tag. However, the INACTIVE keyword only has this effect when used in field 2. Entering the word INACTIVE for any other field does not set the default for alarms to Disabled when that field is used for the Group State Number during alarm tag configuration. Therefore, if you want to be able to automatically set all alarms to disabled during alarm tag configuration, use field 2 with the INACTIVE keyword.

Enter the appropriate field number from your Alarm Group State Descriptions in the Default State field. If you are configuring for constant alarming requirements that do not change with process states, then you do not need a list of process states. In this case, use the state ACTIVE, which the system places automatically in state 1, with a default state of 1. This causes all alarms for tags in this alarm group to default to Enabled in alarm tag configuration.

You can define up to 850 different alarm groups, and each can have up to 32 process states, depending on your process requirements. The system supports a maximum of 3400 process states formed from any combination of alarm groups and process states. For example, your process can have 106 alarm groups if each group has 32 process states (for a total of 3392 process states); or 850 alarm groups if each group has four process states (for a total of 3400 process states).

If you need to assign the Alarm Group to a process group, refer to the procedure described on page 10-10.

Alarm Mini-Windows

You can attach an alarm group to a graphic with the mini-windows feature. When you enable mini-windows from the Event Preferences screen, all active alarms in that alarm group are displayed in a list at the lower left of the graphic. Refer to Chapter 15 to configure mini-windows for an alarm group.

Deleting Alarm Groups

To delete an alarm group, you must first set the system to the Offline state. Select the **Delete** pushbutton on the Alarm Group Configuration display.

Assigning Alarm Groups to Process Groups

A process group does not have to have a description before you assign an alarm group to it. Therefore, you can assign an alarm group to one or more process groups, whether or not they have descriptions, and then enter the descriptive text for the process groups at another time. Refer to Chapter 2 for detailed information about how alarm group membership in a process group can be assigned for differential monitoring of alarm situations.

You can assign new alarm groups to process groups or modify the process group assignments for existing alarm groups while the system is in the Offline state. When the system is in the Operate state, you can assign new alarm groups to process groups, and you can view, but not modify, process group assignments for existing alarm groups.

Until you disable its membership in a process group, an alarm group is linked by default to all 32 process groups. To change membership to a process group, follow these steps.

1. Select the **Process Group** pushbutton on the Alarm Group Configuration display (Figure 10-3).
2. Select **Yes** or **No** for each process group that an alarm group is to be associated. The **Yes** field links the alarm group being configured to the associated process group. The **No** field disables the link between the alarm group and the process group.

NOTE: For proper operation, ensure that the alarm group state tag is associated with the same process group(s) as the alarm group.

Always set unused process groups to **No**. Otherwise, unintended memberships may occur between users, tags, alarm groups, and action requests.

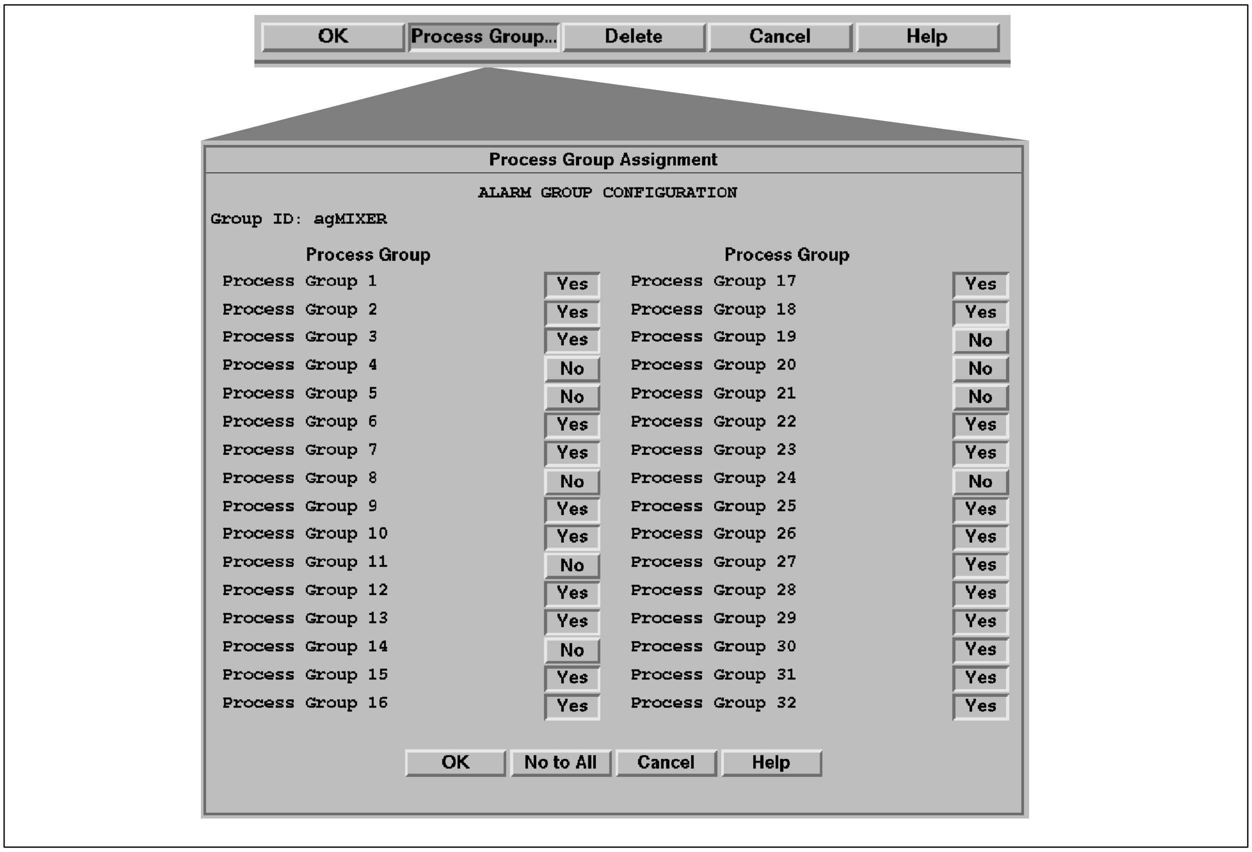
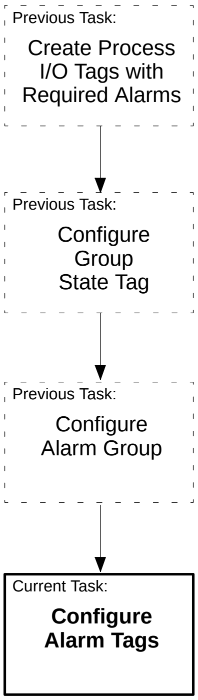


Figure 10-3 Process Group Assignment Display

10.4 Converting Tags to Alarm Tags



To convert tags to alarm tags, you must have either the Database Administration or the System Configuration security privilege. You must have already configured process I/O tags and at least one alarm group. You can add new alarm tags or modify existing ones while the system is in the Offline state. When the system is in the Operate state, you can add new alarm tags, and you can view, but not modify, existing alarm tags. When you are configuring alarm tags, you cannot change system state.

Select **Data->Alarm Tag** from the menu bar to display the Alarm Tag Configuration dialog box (Figure 10-4). Then fill in the fields described below.

Tag Enter the name of a tag that has already been configured; it must be one of the following tag types: AI, AREA, BCH, DI, DI10, DO, DO10, DEVICE, LOOP, UNIT and system tags. The system tags are preconfigured system tags resident in the database. Do not configure a tag for alarming unless its status attribute is networked (being scanned).

NOTE: Because the system tags configured in OSx are crucial to system operation, you cannot delete a tag beginning with "_" from the system alarm group. If you add a tag that begins with "_" to the system alarm group, you cannot delete it. However, you can delete all other tags that you add to the system alarm group and all "_" tags in other groups.

Description This is a text string that describes the tag. This field fills automatically when you enter the tag name and is the same description that you entered when you configured the tag.

Tag Type This is the type of tag, such as AI, LOOP, or DI. This field is filled automatically when you enter the tag name.

Alarm Group Enter the name of a previously configured alarm group.

Description This is the description that you entered for the alarm group when you configured the alarm group. This field fills automatically when you enter the group name.

Group State Description This is the description of the current process state. This field fills automatically with the default state description when you enter the alarm group name in the alarm tag configuration display. The group state description appears as you entered it when you configured the alarm group initially. If you need to view another process state description, enter a new process state number in the group state number field.

Group State Number This is the default process state number that you entered in the Default State field when you configured the alarm group. This field fills automatically when you enter the alarm group name. Change this number when you want to enter alarm tag information for a different process state.

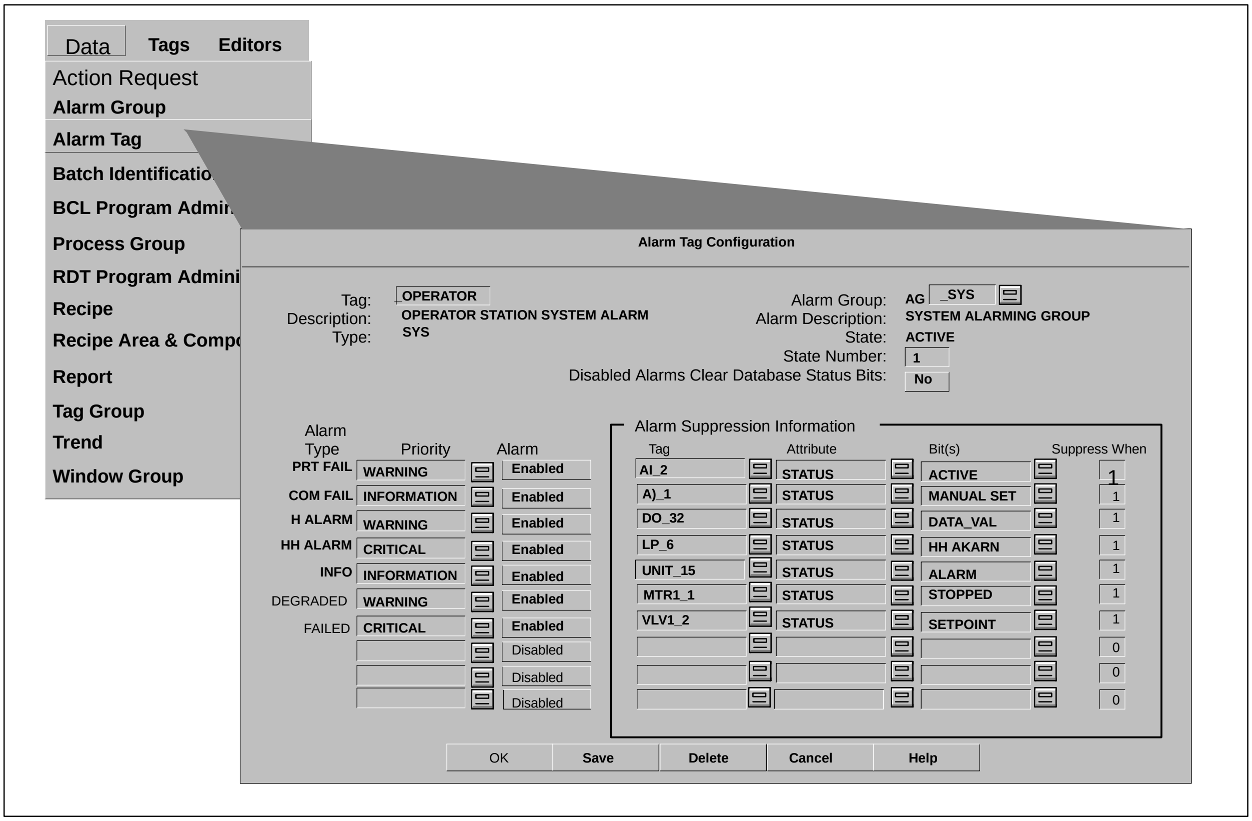


Figure 10-4 Alarm Tag Configuration Display

Converting Tags to Alarm Tags (continued)

Disabled Alarms Clear Database Status Bits Determine whether you want to clear the status bits in the database for all disabled or suppressed alarms for a particular tag, and then select **Yes** or **No.**

Tag status bits that are not disabled or suppressed are not affected by this selection. Make your selection based on the following.

No The disabled/suppressed alarm bits are kept in the database and are available for operations such as screen animation, trending, reports, and autologging.

Yes The disabled/suppressed alarm bits are not kept in the database and are not available for operations such as screen animation, trending, reports, autologging.

NOTE: Select **No** (the default option) to ensure that the operator still has mechanisms (visual representation, reports, logs) for following the status of disabled or suppressed alarm bits. If you select **Yes**, **Clear Database Status Bits**, the operator receives no information whatsoever about the condition of disabled or suppressed alarm bits.

Alarm Type When you enter the tag name, the various types of alarms for the tag type being configured appear automatically. For example, if you are configuring an alarm for a loop tag, the following alarm types appear: BAD XMTR, ROC ALARM, ORANGE DEV, YELLOW DEV, LL ALARM, L ALARM, H ALARM, and HH ALARM.

Priority For each alarm type associated with the tag, the system lists the default priority level: **Critical**, **Warning**, **Information**, or **Maintenance**. These fields fill automatically when you enter the tag name. You can change the priority if necessary.

Enable/Disable Alarm For each process state that you configured, you can enable or disable alarm types, for example, HH ALARM.

Entering Alarm Suppression Information

For each tag configured for alarming, you can configure alarm suppression information, to eliminate nuisance alarms. You can suppress one alarm for each suppression tag specified, and you can use a suppression tag and attribute more than one time. To suppress an alarm, enter the suppression tag information on the line that corresponds to the alarm type to be suppressed. Figure 10-5 shows the Alarm Tag Configuration display with the Alarm Suppression Information fields highlighted. Fill in the fields as described in the following pages.

Before saving the alarm tag configuration, fill in alarm suppression information for each alarm you want to suppress. Alarm suppression, in contrast to alarm disabling, is linked to individual tags, not to specific alarm group states. This means that if you have defined suppression for any alarm, the suppression applies globally to all alarm group states.

> **⚠ WARNING**
>
> **Suppressing or disabling the alarm can lead to incorrect operation of the batch program.**
>
> **Incorrect operation could result in death or serious injury to personnel and/or damage to equipment.**
>
> **Make sure that an alarm is really not needed before suppressing or disabling it. Do not suppress or disable a Unit tag alarm when a batch program references the Unit tag status bit to determine the status of the unit.**

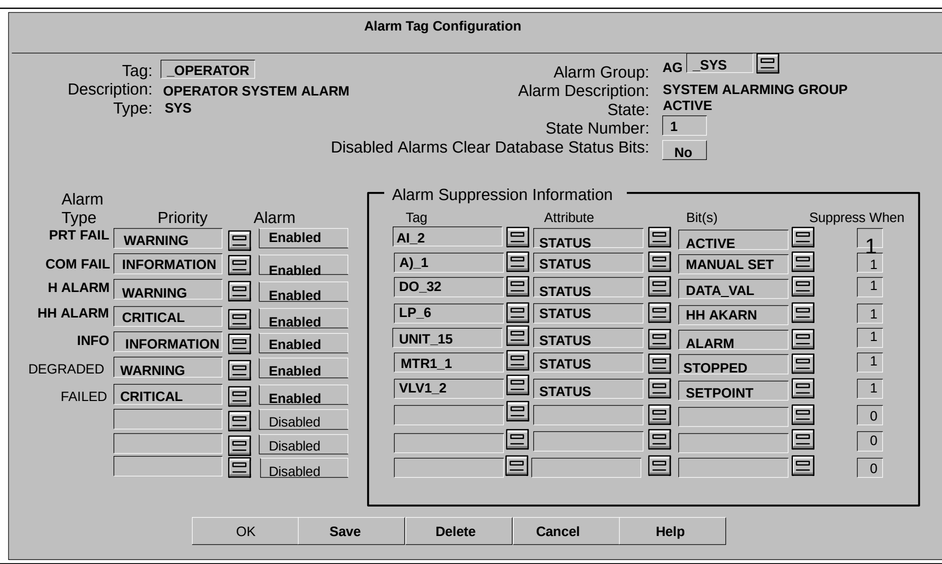


Figure 10-5 Alarm Tag Suppression Fields

Entering Alarm Suppression Information (continued)

Tag Name and Attribute Enter the name of the suppression tag and its associated attribute exactly as they were entered during the tag configuration. The default attribute is status. If an attribute other than status is required, enter the name of that attribute instead.

NOTE: Floating point attributes cannot be used for alarm suppression. Use only signed or unsigned 16-bit integer attributes.

Bit(s) Enter the bit name or value. This field identifies the name of the suppression bit or the corresponding hexadecimal constant or integer constant for the suppression bit. Enter the bit name, or the hexadecimal constant (prefixed by 0x), or the integer constant. Refer to the example in Figure 10-6. Chapter 3 on defining tags, shows the bit names and hexadecimal values for all the tag types. The Error Dev bit of the LOOP status attribute is not available for alarm suppression.

If you enter an integer value in the Bit column, the system converts the value to hexadecimal after you save the alarm configuration and recall it. If the resultant hexadecimal value has an associated bit name, the system converts the hexadecimal value to the bit name after you save the alarm configuration and recall it.

NOTE: Do not allow alarm suppression tags to suppress each other so that they form a circular relation. That is, do not have tag 1 suppress tag 2, and have tag 2 suppress tag 3, and have tag 3 suppress tag 1.

If you have one alarm tag suppressing another, and then disable the first tag, then you have removed the suppression capability that you intended.

Suppress When Select bit status. You can suppress an alarm when a bit of the suppression tag is set either to 1 or 0. The default value is one.

For example, if you set the state of the suppression bit to 1, the system suppresses the corresponding alarm when the the suppression bit is 1. Similarly, if you set the state of the suppression bit to 0, the system suppresses the corresponding alarm when the suppression bit is 0.

Requirement	For Loop1, suppress the loop orange deviation alarm when the bad transmitter alarm is true.
Data Entry for Orange Deviation Alarm	Enter these values in following fields: In the **Tag** field, select `Loop1` In the **Attribute** field, enter `Status` In the **Bit(s)** field, enter `Bad Xmtr.` You can enter `32` or `0x20` instead. Refer to the bit map below.

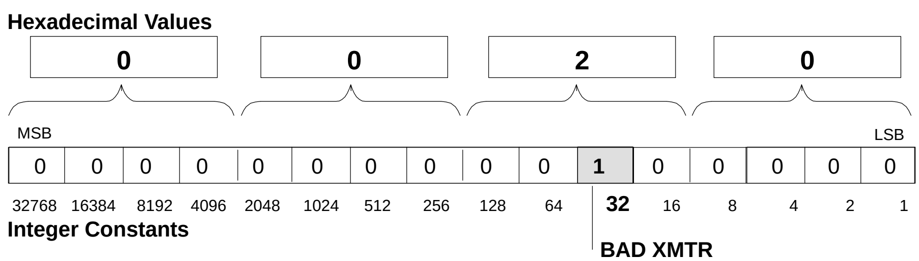
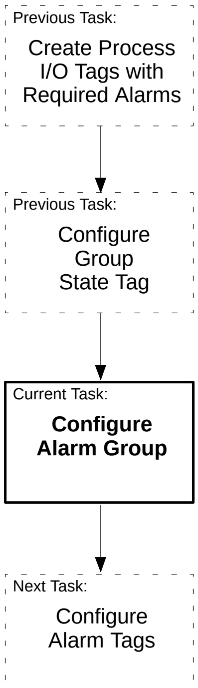


Figure 10-6 Sample Values for Alarm Suppression Bits

10.5 Example 1: Alarm Configuration for One Process State



This section gives an example of an alarm configuration for a process situation that has consistent alarming requirements. That is, the alarm requirements do not change as the state of the process changes. An alarm suppression capability that is based on a discrete event is provided.

First choose a tag, for example, a loop tag. You want to monitor these alarms: High High, High, Low, and Low Low. In this example, the alarming requirements are consistent (do not change from one process state to another), so a Group State Tag is not necessary. Therefore, you can begin your alarm configuration by configuring the alarm group.

To configure the alarm group, refer to Figure 10-7 and follow these steps.

1. Enter the group identification in the Group ID field.
   This identifies the group by name. The AG prefix is already shown preceding the field.

2. Enter the group description in the Description field.

3. Enter the name of a display to be associated with the alarm group in the Associated Display field.

4. Leave the Group State Tag and Attribute fields blank. In this example, there are no process states.

5. Leave the Default State value unchanged at 1.

6. Leave the Alarm Group State Description values unchanged. Since the process does not have multiple states in this example, a list of process states is unnecessary.
   OSx puts the default entry ACTIVE, which enables all alarms, in state 1. This value is used in the example.

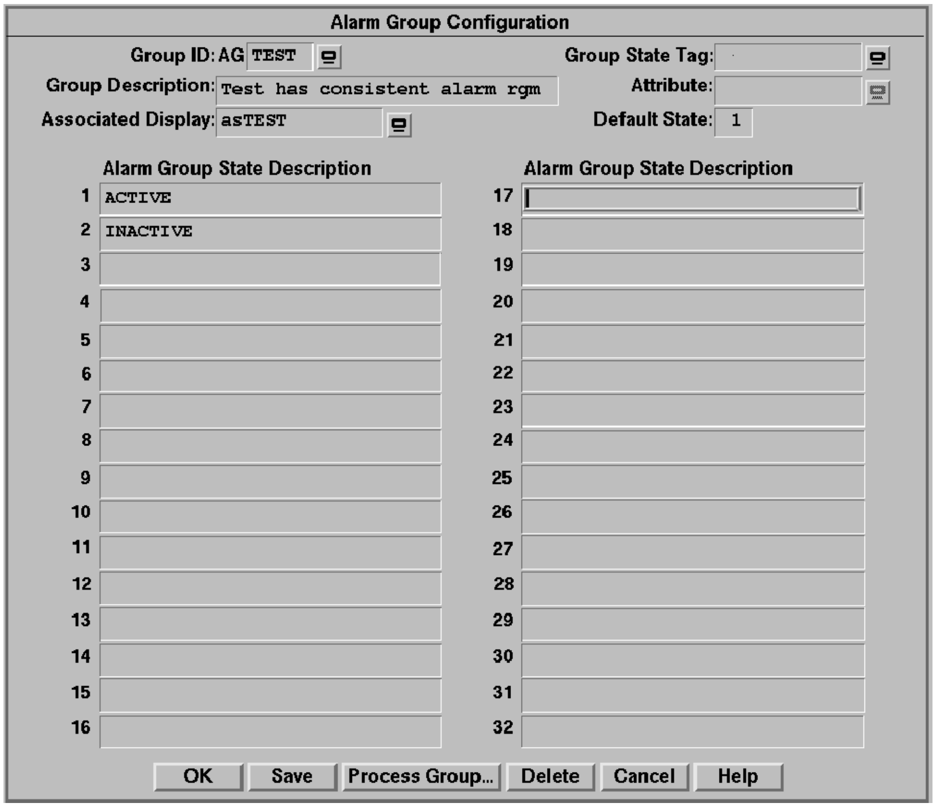
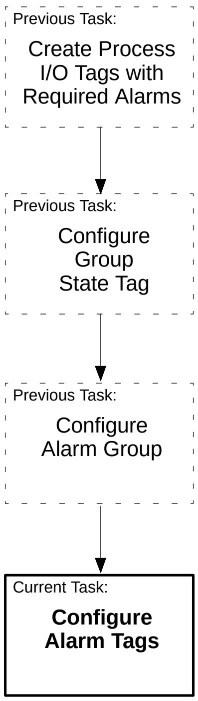


Figure 10-7 Example 1: Alarm Group Configuration

Example 1: Alarm Configuration for One Process State (continued)



To configure the alarm tag, refer to Figure 10-8 and follow these steps.

1. Enter the loop tag name in the Tag field.
   The tag type and description with all the tag alarms appear automatically.

2. Enter the Group ID that you assigned during alarm group configuration (page 10-18) in the Alarm Group field. The AG prefix appears preceding the field.
   The alarm group description appears automatically, along with the group state description and number (ACTIVE and 1), respectively.

3. For each of the four alarms that you want to monitor, enter the tag (DO_1), attribute (status), and bit (data_val) in the respective Alarm Suppression fields. Enter the value 1 in the Suppress When field for each alarm. This suppresses all four of the absolute alarms when the controller discrete input X17, which is associated with DO_1, turns on.

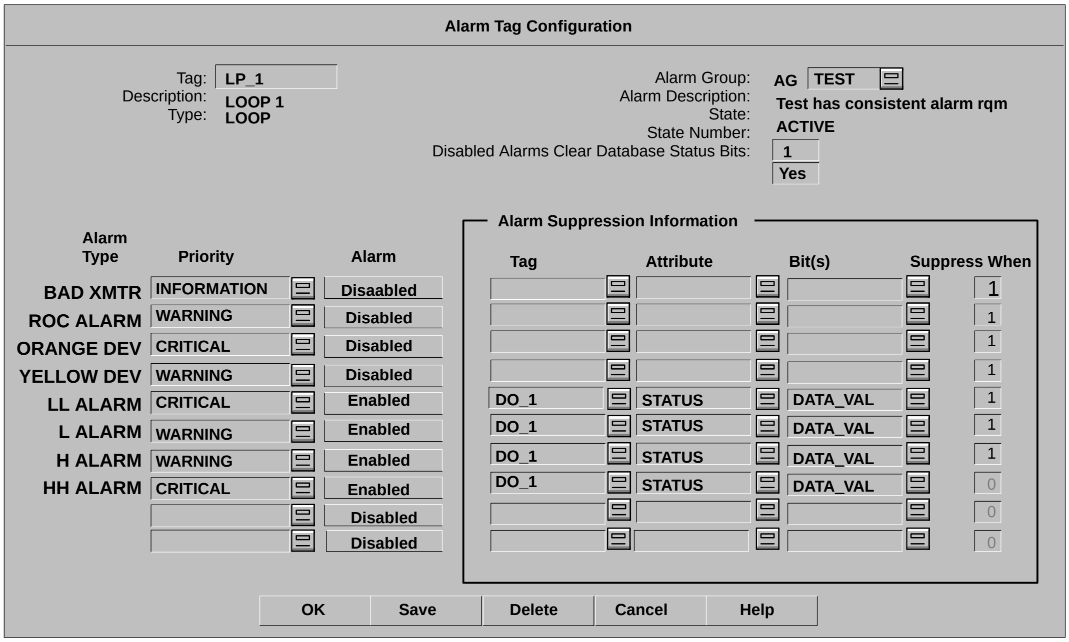


Figure 10-8 Example 1: Alarm Tag Configuration

Example 1: Alarm Configuration for One Process State (continued)

Test of the Alarm System for Example 1

Once you have completed the configuration of the alarm system, place the system in the Operate state and display the tag detail of the loop tag used in the example. Manipulate the process variable (pv) by changing the output in loop manual mode or the setpoint in loop auto mode. Forcing the rise and fall of the pv within the loop's range triggers the absolute alarms (with values assigned originally in the controller program). You can also change alarm values, provided the loop tag is configured to allow this. You can suppress, or inhibit, triggered alarms. You can cause alarms never to be triggered if you suppress before triggering has occurred. Figure 10-9 and Figure 10-10 show the relationship of the physical hardware components to the OSx station display.

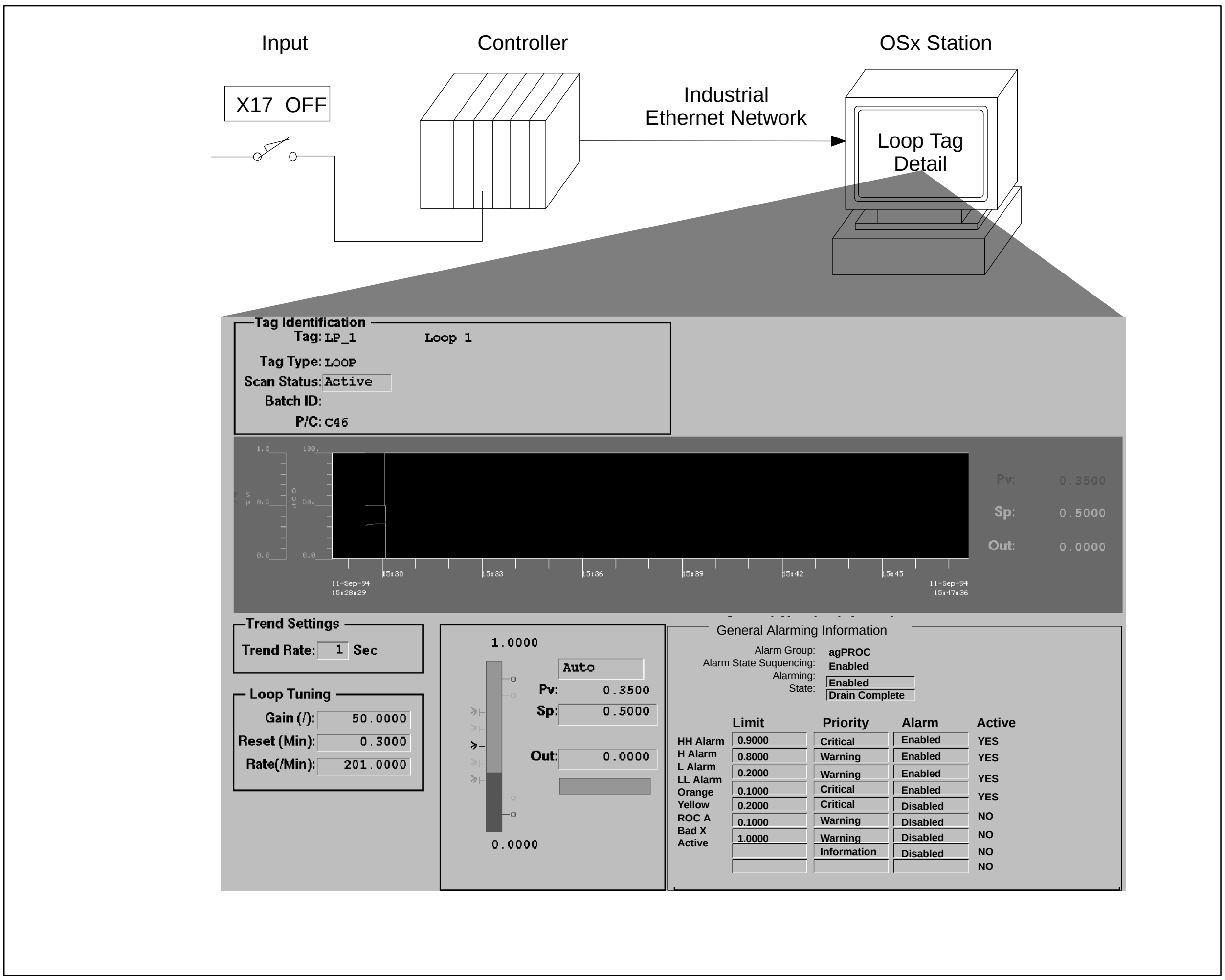


Figure 10-9 Display with Input Off

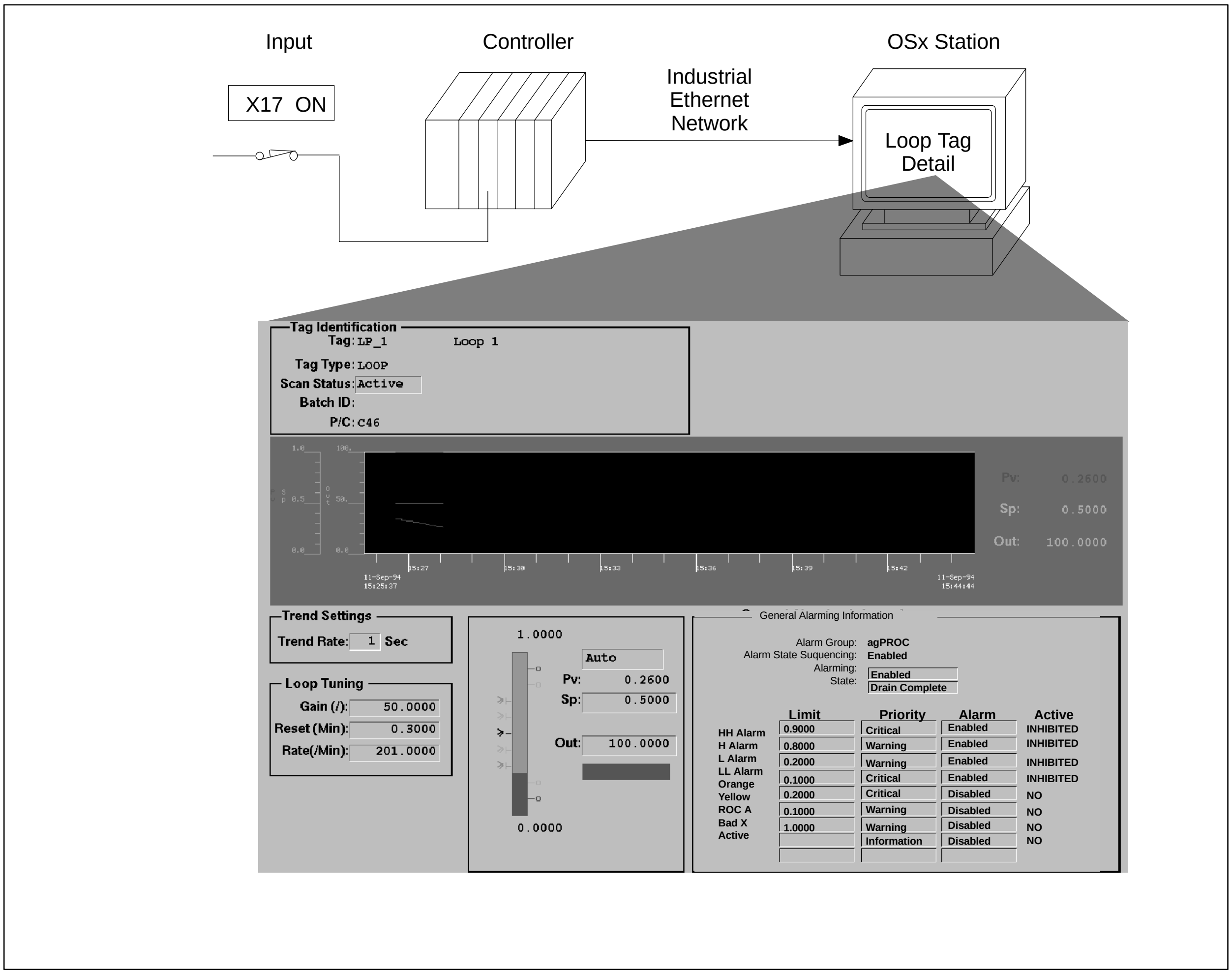


Figure 10-10 Display with Input On

10.6 Example 2: Alarm Configuration for Multiple Process States

This section gives an example of an alarm configuration for a process simulation that has varying alarm requirements. That is, the alarm requirements change as the state of the process changes. The process simulation consists of four states, each using different combinations of enabled and disabled alarms. An alarm-suppression capability that is based on a discrete event is provided.

This simulation is based on an APT Sequential Function Chart. There are four programmed states: fill, mix, heat, and drain (Figure 10-11). The code contained in the individual steps is shown in Figure 10-12.

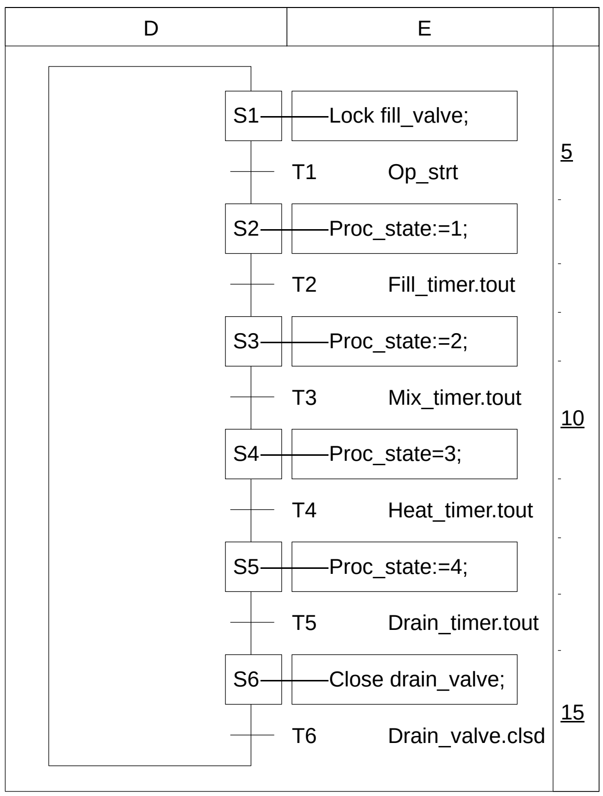


Figure 10-11 Example 2: APT Sequential Function Chart

```
Cell : E-5     Label: S1
...........      .........
   Lock fill_valve;
   Lock mixer;
   Lock heater_valve;
   Lock drain_valve;
Cell : E-6     Label: T1
...........      .........
   Op_strt
Cell : E-7     Label: S2
...........      .........
   Proc_state:= 1;
   Open fill_valve;
   Delay fill_timer; {Start fill_timer}
Cell : E-8     Label: T2
...........      .........
   Fill_timer.tout
Cell : E-9     Label: S3
...........      .........
   Proc_state:= 2
   Close fill_valve;
   Start mixer;
   Delay mix_timer;
Cell : E-10    Label: T3
...........      .........
   Mix_timer.tout
Cell : E-11    Label: S4
...........      .........
   Proc_state:= 3;
   Stop mixer;
   Open heater_valve;
   Delay heat_timer;
Cell : E-12    Label: T4
...........      .........
   Heat_timer.tout
Cell : E-13    Label: S5
...........      .........
   Proc_state:= 4;
   Close heater_valve;
   Open drain_valve;
   Delay drain_timer;
Cell : E-14    Label: T5
...........      .........
   Drain_timer.tout
Cell : E-15    Label: S6
...........      .........
   Close drain_valve;
Cell : E-16    Label: T6
...........      .........
   Drain_valve.clsd;
```

Figure 10-12   Example 2: APT Sequential Function Chart Step Code

Current Task:
Create Process I/O Tags with Required Alarms

Next Task:
Configure Group State Tag

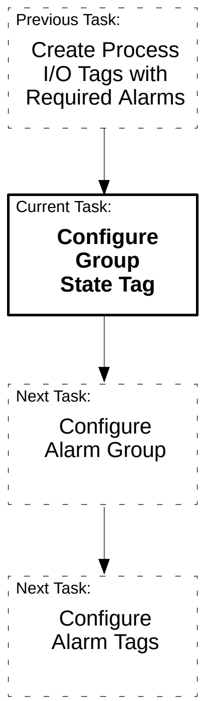
Next Task:
Configure Alarm Group

Next Task:
Configure Alarm Tags

For this example, choose a loop tag (Step 1 in the alarm configuration procedure). It has these alarms that need to be monitored: High High, High, Low, and Low Low. Table 10-3 shows the configuration for the I/O tag.

Table 10-3 Example 2: I/O Tag Configuration

Attribute	Memory	Locations	Upload	20% Offset	Init Value
CHANGE					1
STATUS	LSTATUS1	1	N	N	0
MODE	LMODE1	1	N	N	0
L_RANGE	LPVL1	1	Y	N	0
H_RANGE	LPVH1	1	Y	N	1000
PV	LPV1	1	N	N	47
SP	LSP1	1	N	N	0
ROC_ALARM	LRCA1	1	Y	N	1
L_DEV	LYDA1	1	Y	N	0.1
H_DEV	LODA1	1	Y	N	0.2
LL_ALARM	LLLA1	1	Y	N	25
L_ALARM	LLA1	1	Y	N	100
H_ALARM	LHA1	1	Y	N	800
HH_ALARM	LHHA1	1	Y	N	900
RATE	LTD1	1	Y	N	0
RESET	LTI1	1	Y	N	999.99
GAIN	LKC1	1	Y	N	1
OUT	LMN1	1	N	N	0

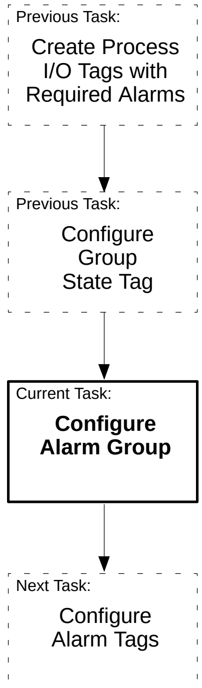


In order to move the value of the current step or state from the controller memory into the Alarm System, you must configure a group state tag. You can use any tag that is capable of providing an integer value for the process state, such as integer variable (IVAR) or counter (CTR). Table 10-4 shows the configuration for the group state tag, which is an IVAR.

Table 10-4 Example 2: Group State Tag Configuration

Attribute	Memory	Locations	Upload	Auto Log	Init Value
L_RANGE	V100	1	Y	N	1
H_RANGE	V101	1	Y	N	4
STATUS	V102	1	N	N	0
VALUE	V103	1	Y	N	1
UNITS					Step

Example 2: Alarm Configuration for Multiple Process States (continued)



After you have created the group state tag, you can configure the alarm group. Figure 10-13 shows alarm group AGPROC, which lists the four states that were programmed in the APT Sequential Function Chart (Figure 10-12). The group state tag STEPNO indicates the current process state number and corresponds to the variable Proc_state used in the APT program. The default alarm group state is State 4 (Drain).

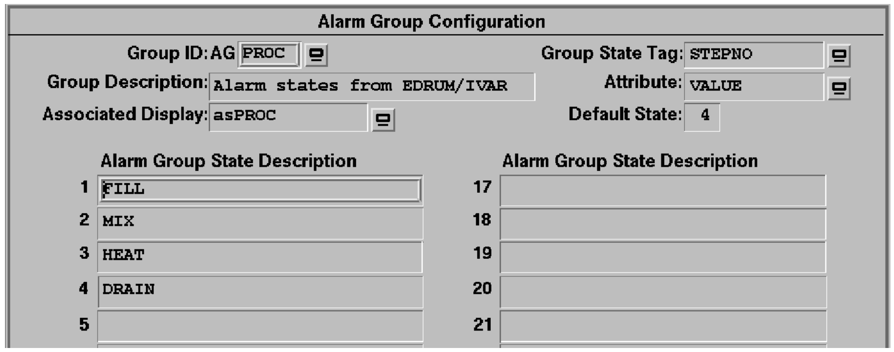
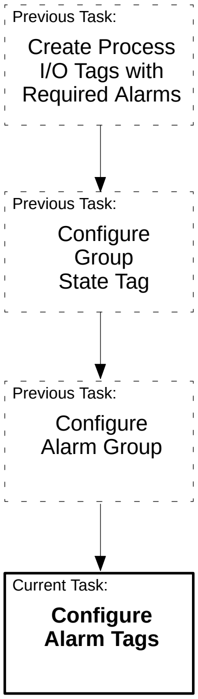


Figure 10-13 Alarm Group AGPROC



Now you can assign tags to the alarm group AGPROC. You do this in the Alarm Tag Configuration display (Figure 10-14). Here, you assign each tag that you select for alarming to the alarm group AGPROC. You can enable or disable the alarms that are provided in the status attribute of the alarm tag to fit the process.

Since this batch process has multiple states, you must fill out configuration pages for each state. You can also program alarm suppression information here (Figure 10-15).

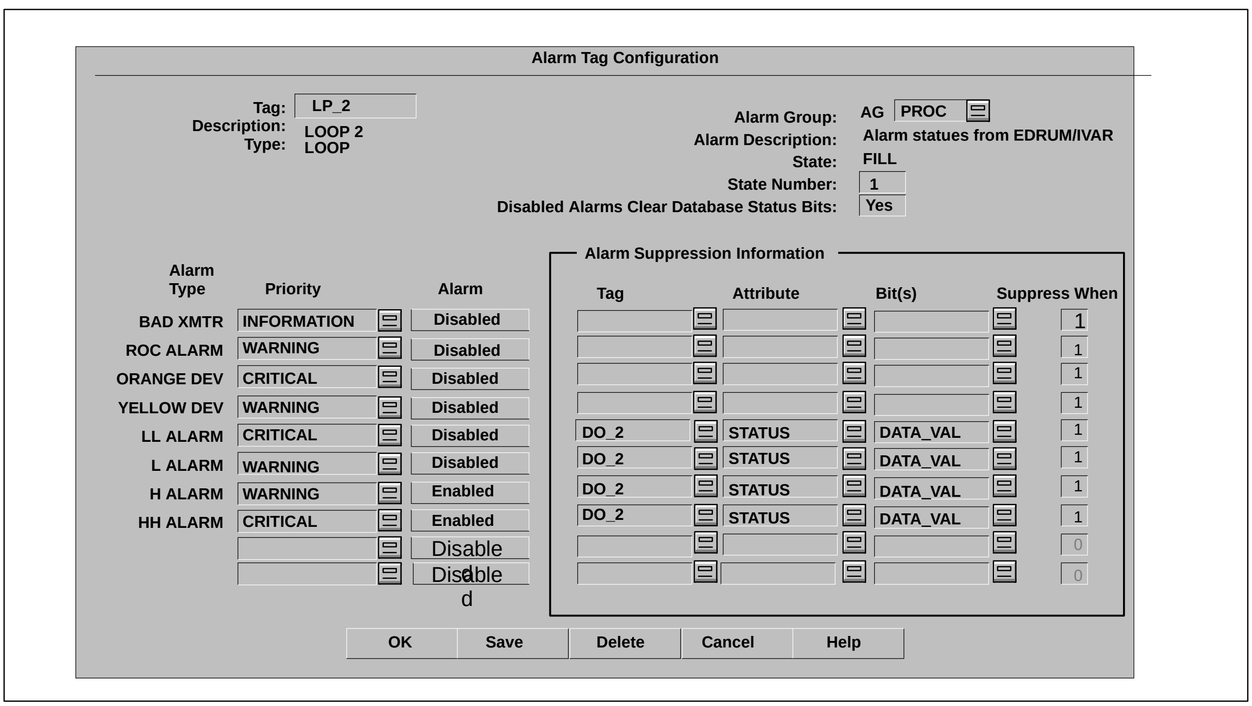


Figure 10-14 Alarm Tag for AGPROC

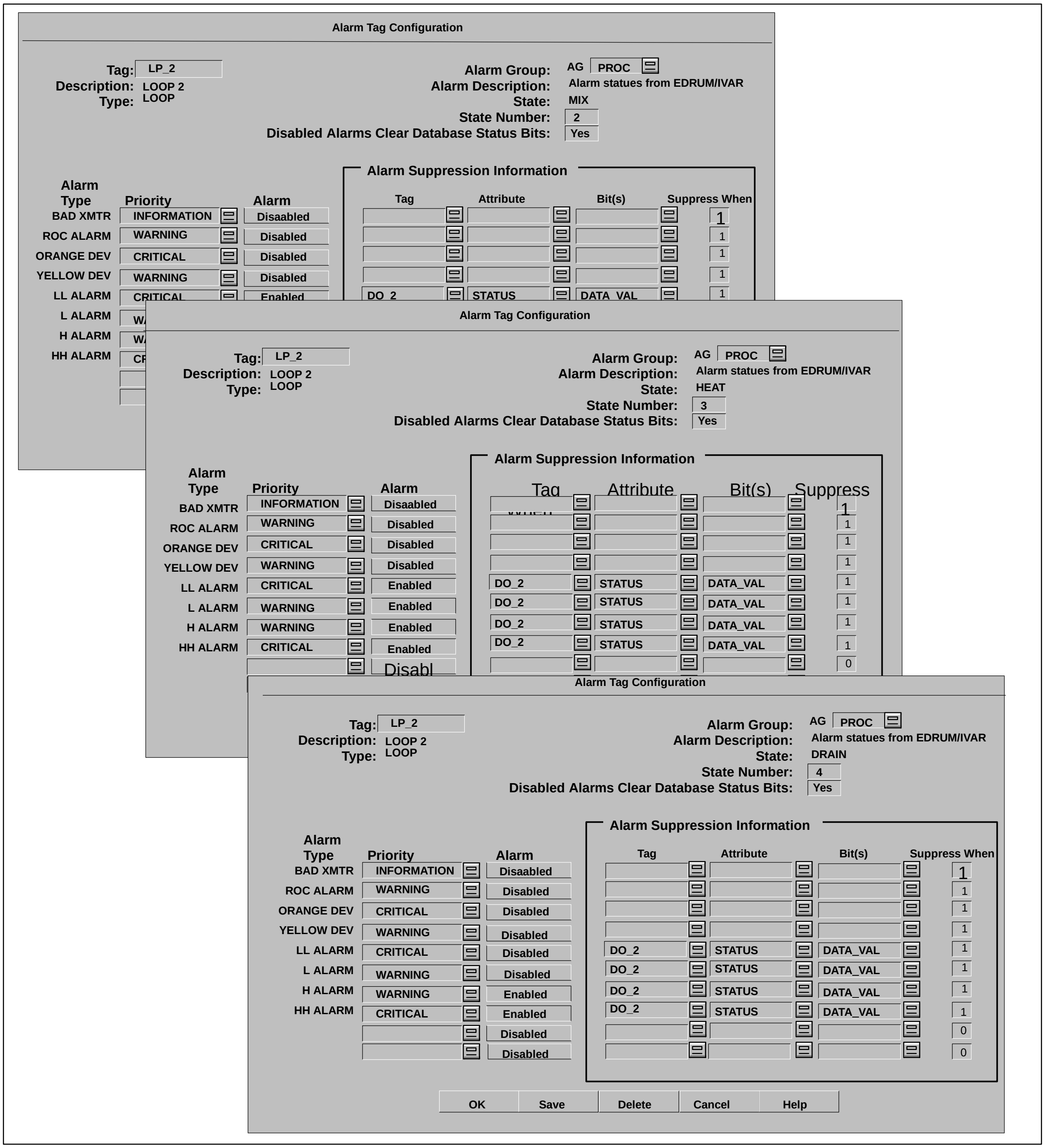


Figure 10-15 Alarm Tags for AGPROC Multiple States

Test of the Alarm System for Example 2

Once you have completed the configuration of the alarm system, place the system in the Operate state and display the tag detail of the loop tag used in the example (Figure 10-16). Manipulate the process variable (pv) by changing the output in loop manual mode or the setpoint in loop auto mode. Change alarm values. Notice, in the loop tag detail, that as the APT Sequential Function Chart runs through its steps (or states), the absolute alarms are enabled and disabled in the combination that you configured for that particular state in the alarm tag configuration.

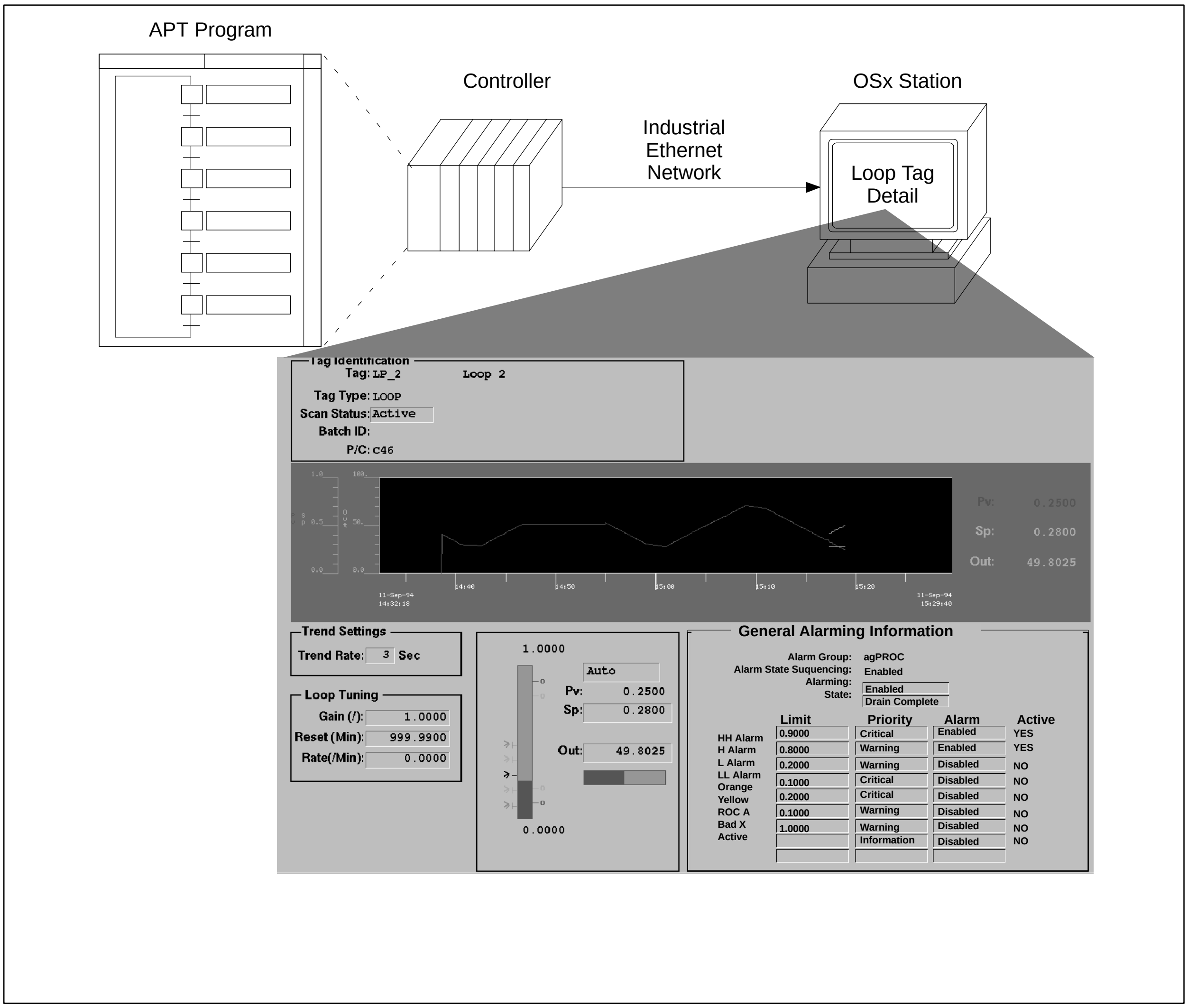


Figure 10-16 Example 2: Loop Tag Detail

Display the tag detail for the IVAR tag, and see the steps change there (Figure 10-17).

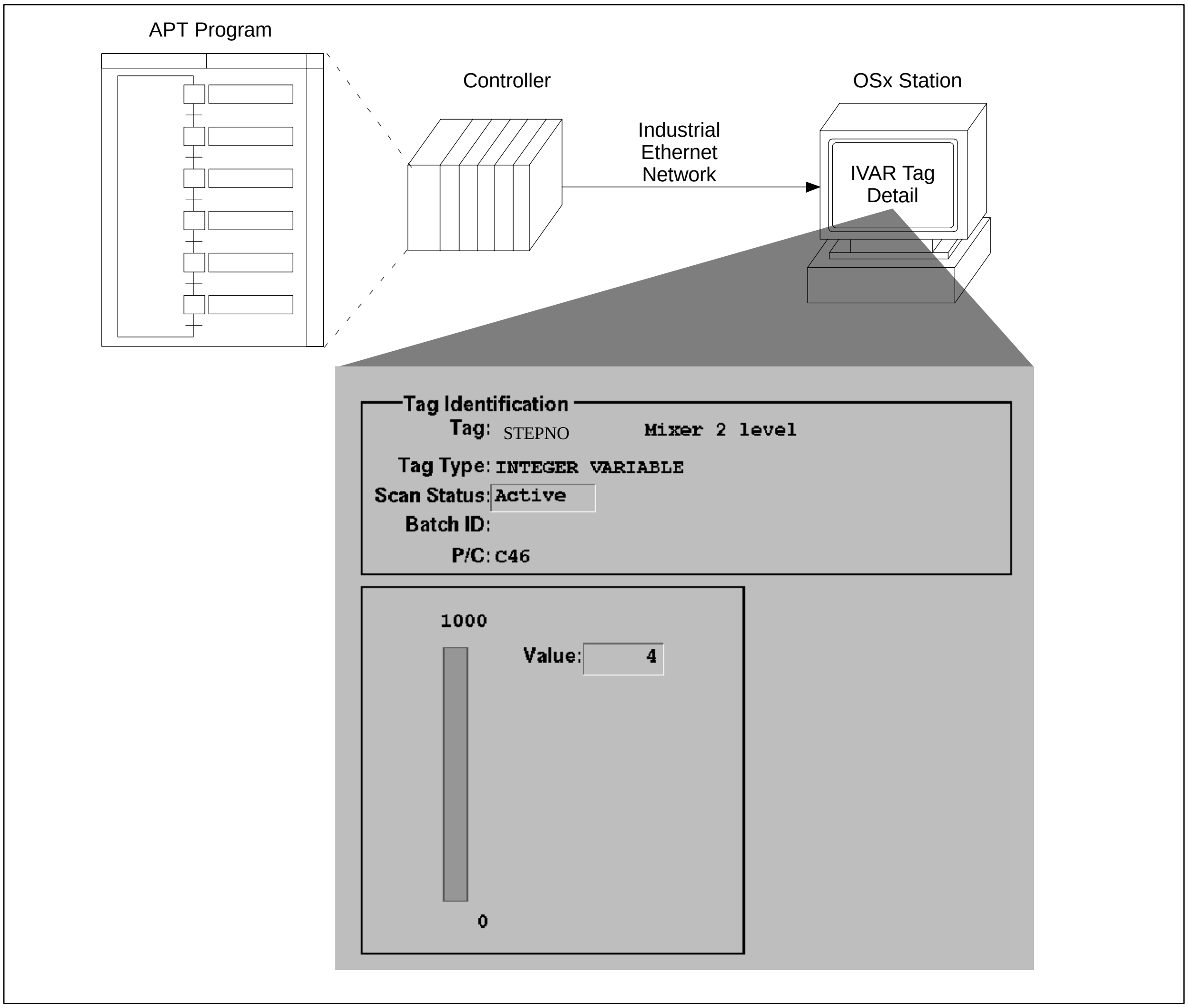


Figure 10-17 Example 2: IVAR Tag Detail

Display the Alarm Group display and see the alarm group State Number and Description change as the APT Sequential Function Chart runs (Figure 10-18). Note the alarms that have been triggered in the group too.

NOTE: Although non-networked tags can be configured as alarm tags and assigned to alarm groups, they do not generate alarms. Only tags networked to a controller and OSx tags generate alarms.

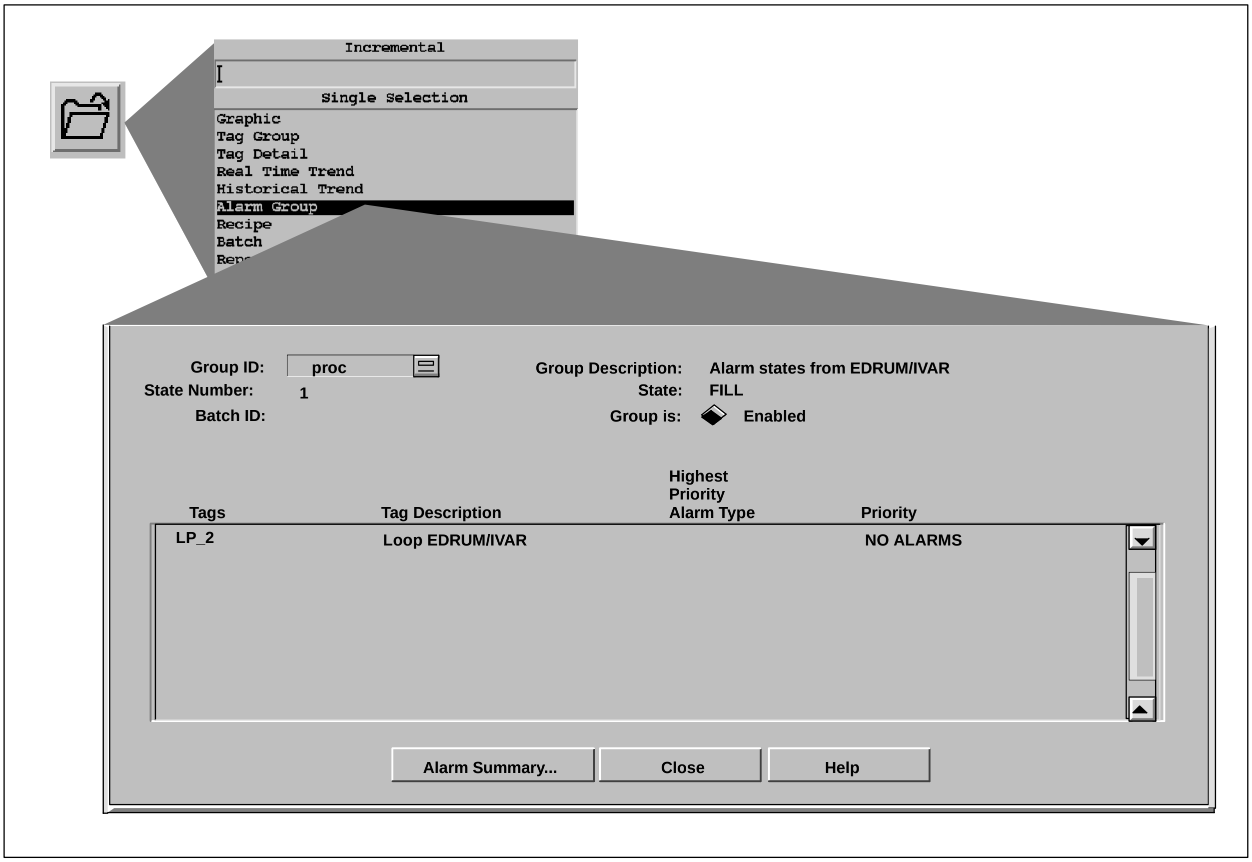


Figure 10-18 Example 2: Alarm Group Display

10.7 Configuring the Annunciation of External Alarms

Overview

The OSx alarm board provides a means of connecting an external alarm, such as a speaker, light, or siren. The manner in which the alarm is annunciated depends on the configuration of several variables in the **term_def** relation in the database. This relation is preconfigured to annunciate critical and warning alarms from the OSx station speaker. When a critical alarm occurs, the speaker sounds a high-pitched, rapid beep. The default configuration for the external alarm uses a pulsed mode with approximately one second between pulses.

⚠ WARNING

If your operating environment involves considerable background acoustical noise, you might not hear an important alarm.

Missing an important system alarm could result in death or serious injury to personnel, and/or damage to equipment.

Turn off the OSx station speaker and install an external alarm that is adequate for the environment in which it operates.

You can change the configuration to annunciate both critical and warning alarms and to change the pulse rate, tone, and duration of the annunciator.

You can also change alarm annunciation from the pulse mode to a latched mode. In latched mode, alarm annunciation toggles between the signals for critical and warning alarm. For example, if a critical alarm occurs after a warning alarm, the audible or visual alarm would automatically switch signals to produce the alarm for critical alarms.

To customize the operation of the alarm board or OSx station speaker, see your system administrator, or refer to the *SIMATIC PCS 7 OSx System Administration Manual.*

10.8 Printing the Alarm Configuration Report

Overview of the Configuration Report

The Alarm Configuration Report (Figure 10-19) lists all configuration information for alarms.

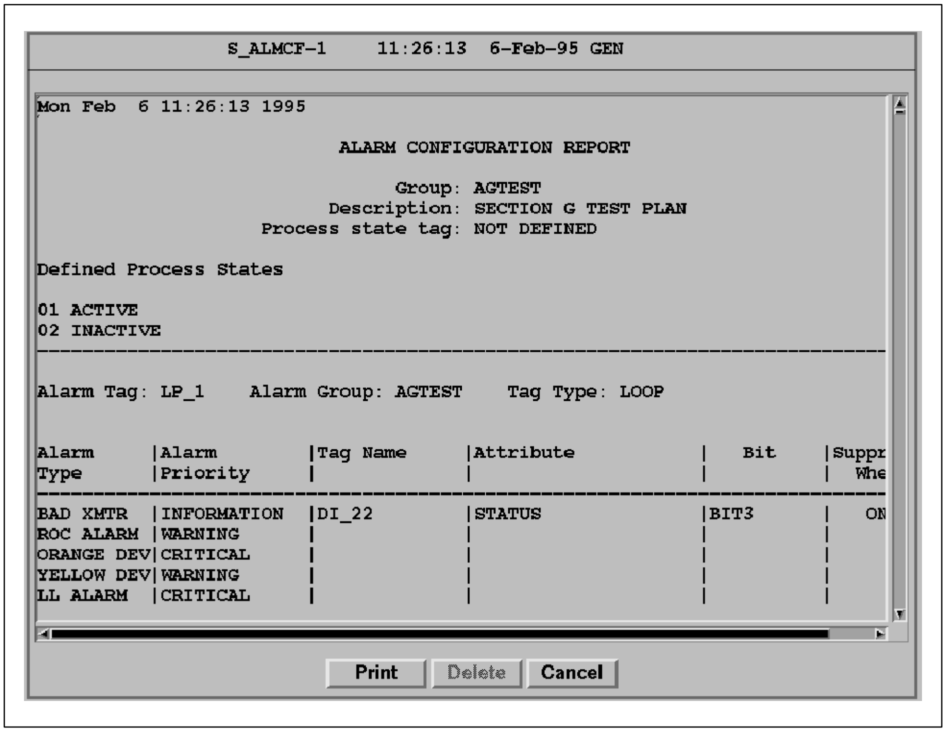


Figure 10-19 Sample Alarm Configuration Report

Accessing the Report Formats Directory

To access the Report Formats Directory, select **Data->Report** on the main menu. The screen displays the Reports Directory, which contains a listing of all executable reports.

The name of the Alarm Configuration Report is S_ALMCF.

Figure 10-20 shows the names of the standard tag configuration reports. The highlighted format is the current selection.

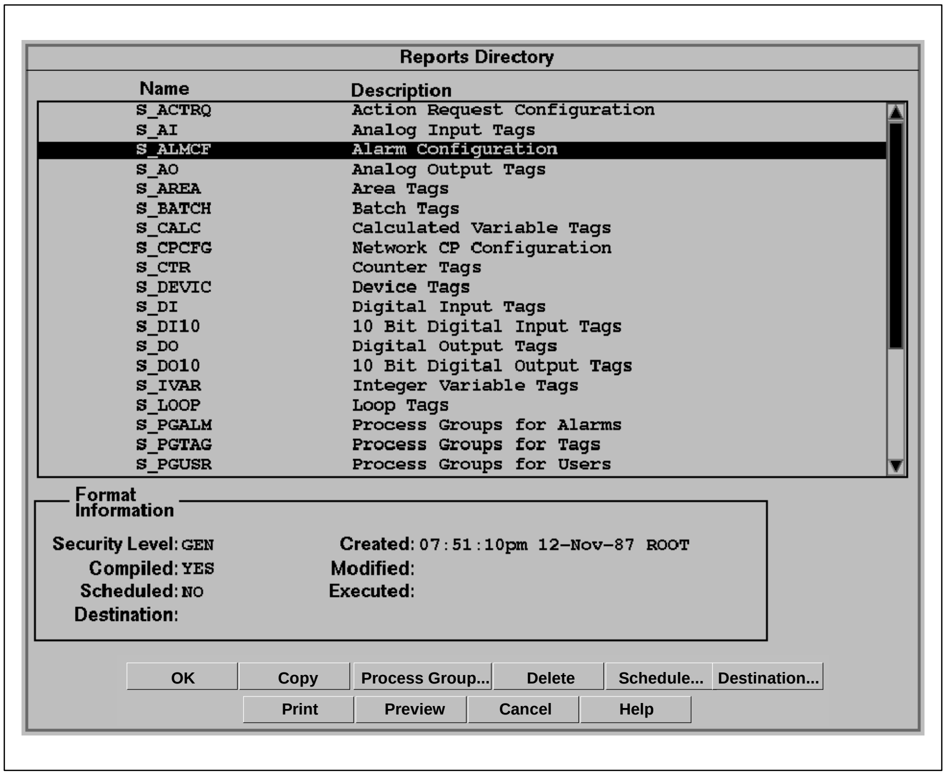


Figure 10-20 Reports Directory

Printing the Configuration Report

Like all other standard reports, you can place the Alarm Configuration Report output in a file or print it. To generate the report output, follow the steps for scheduling or previewing and printing reports outlined in the *SIMATIC PCS 7 OSx Reports Manual*. The same manual gives the steps for printing a report.

Chapter 11

Configuring Screen Hierarchies

11.1 Overview of Associated Displays

Accessing Displays with Screen Hierarchies and Associations

OSx allows you to link process graphics and other associated displays logically in screen hierarchies. Only the main graphic on the screen can be linked in this way (Section 11.4). The structure, or hierarchy, that you create allows an operator to take shortcuts when paging from screen to screen. The operator can monitor a process more easily and respond to problems with greater efficiency. Figure 11-1 illustrates an example of how you can structure graphics to provide views of your process. The sample hierarchy progresses from the plant overview, the home display, to close-ups of each major area of the process and to units within each area. The arrows indicate possible paging patterns: up, down, left, right.

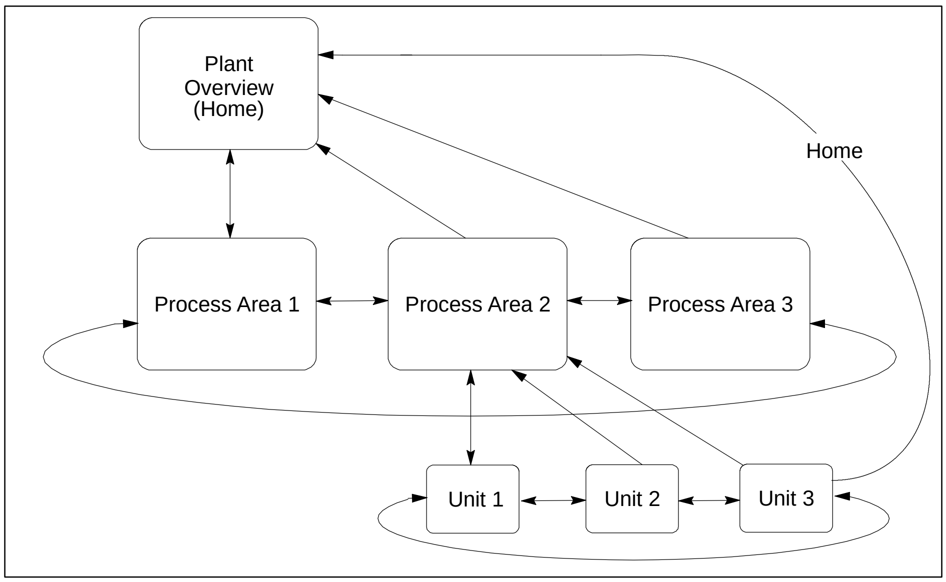


Figure 11-1 Sample Paging Hierarchy of Process Graphics

Each display type (graphic, real-time trend, historical trend, tag group, alarm group, and alarm summary) can have one or more displays associated with it. Associated displays allow an operator to move through the screens quickly and conveniently.

Figure 11-2 shows an example of how displays could be associated. The current display can be an overview graphic, for example, with real-time and historical trends and help screens. A page to the right could show another production line or area with its own associated displays.

The associated displays for each of the display types can be related or they can stand independently. Two displays representing different production lines can share the same associated report, such as a shift summary for a given production area.

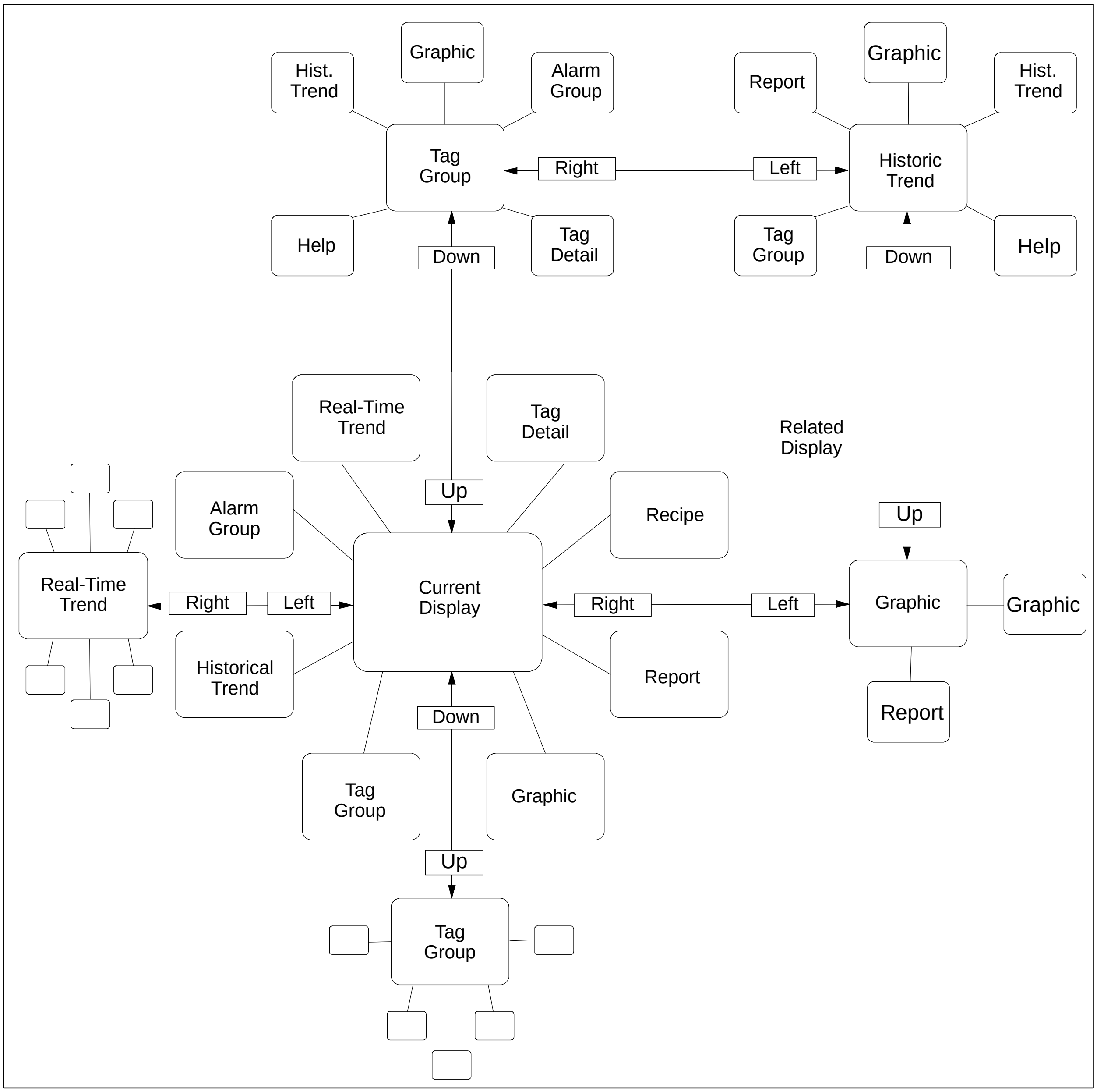


Figure 11-2 Example of a Screen Hierarchy Design

Overview of Associated Displays (continued)

When the system is in the Operate state and a user is logged in, the screen hierarchy for the current display appears in the navigation area of the screen (Figure 11-3). This allows an operator to use the pointing device to select an associated display from the screen.

If you create a hierarchy while the system is in the Operate state, you must access the top display (the display for which the hierarchy is created) before the hierarchy items appear in the navigation area.

NOTE: Only the screen types configured for the hierarchy of the current display are accessible in the navigation area while the system is in the Operate state.

In the Operate state, OSx displays the navigation icons without text labels by default. You can configure your system to display navigation icons with or without text labels defining their functions. Follow these steps.

1. Select **Startup->Event Preferences** from the menu bar. The Event Preferences dialog box appears.

2. In the Navigation Type field, select one of the following:

   Icon Only to display the icons without text (Figure 11-3, top).

   Icon with Text Descriptor to display the icons with text (Figure 11-3, bottom).

3. Click **OK** to save your configuration.

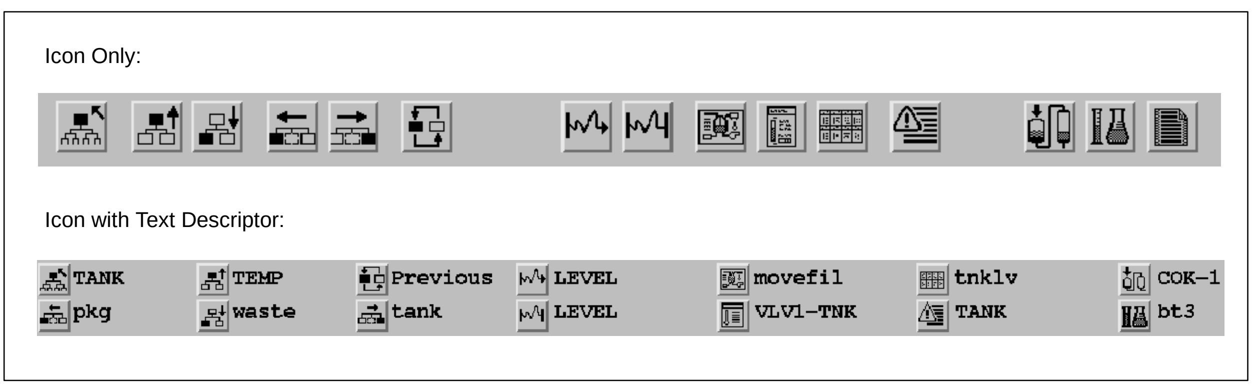


Figure 11-3 Screen Hierarchy Page Area

11.2 Preparing to Configure Displays in Screen Hierarchies

Before you begin to configure displays for operator access, name all the information you plan to use in displays and write it down for easy reference. You can access the following OSx components through displays:

- Alarm groups
- Reports
- Graphic displays
- Trends
- Tag details and tag groups
- Recipes

After you have planned the hierarchy, you are ready to configure displays.

11.3 Configuring Screen Hierarchies

To configure a screen hierarchy, you must have either the Database Administration or the System Configuration security privilege. The system can be in either the Offline or Operate state. Before you can configure a screen hierarchy, you must first create the display screens that you want to include in the hierarchy.

To configure a screen hierarchy, select **Startup->Screen Hierarchy** from the menu bar (Figure 11-4). Then fill in the fields described below.

Display Type Select the type of display for which you are creating the hierarchy: graphic, tag group, real-time trend, or historic trend.

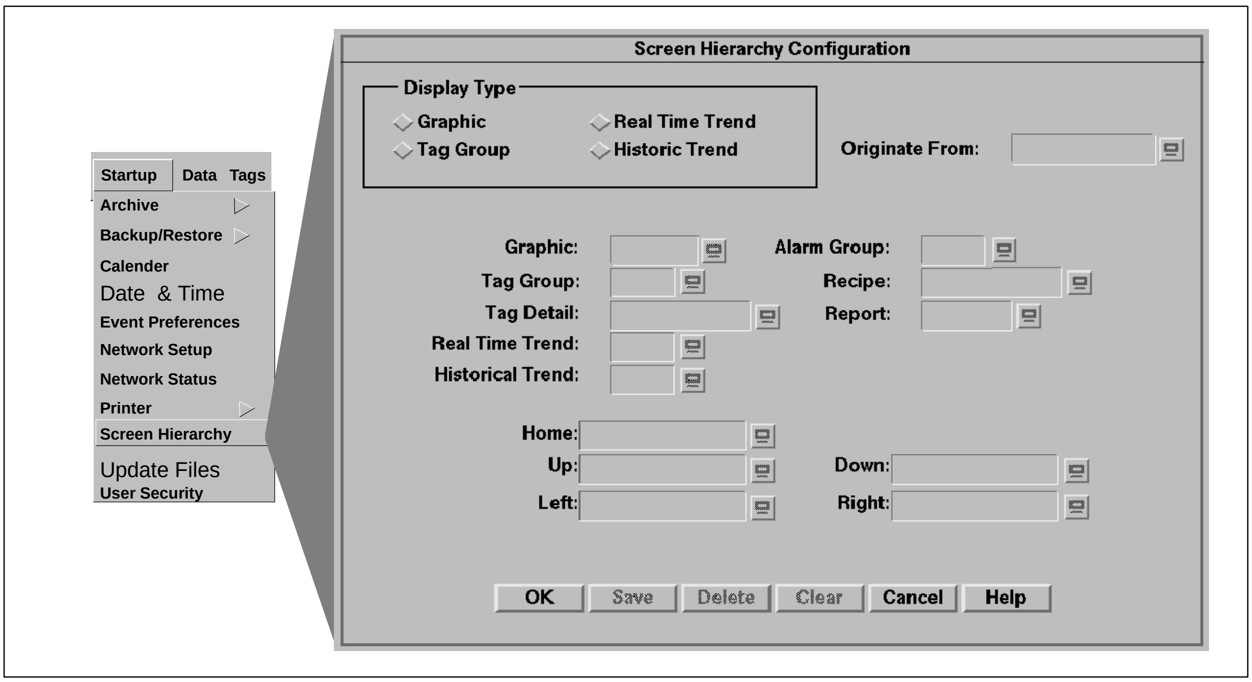


Figure 11-4 Screen Hierarchy Configuration Dialog Display

Originate From Enter the name of the display for which you are creating the hierarchy. If the hierarchy already exists, the existing hierarchy appears. If no hierarchy exists by the current name, you can create one. You can create a hierarchy with the names of the associated displays first and then create the actual associated displays later. To associate a display with your hierarchy, enter its name in one of the following fields.

Graphic
Tag Detail
Historical Trend
Recipe
Home
Left
Right
Tag Group
Real-Time Trend
Alarm Group
Report
Up
Down

The associated displays appear in the Operate state when the operator presses the appropriate key on the operator keyboard or selects the appropriate option in the navigation area of the screen.

If you link an associated display to **Home, Up, Down, Right,** or **Left**, you must enter the prefix for the display type, unless you use the select list, which enters the prefix for you. See Table 11-1 for valid prefixes.

Table 11-1 Prefixes for Named Displays

Display Type	Prefix	Display Type	Prefix	Display Type	Prefix
Historical Trend	HT	Real-Time Trend	RT	Alarm Summary	AS
Recipe	BT	Tag Group	TG	Alarm Group	AG
Detail	TD	Report	RP	Graphic	GR

Figure 11-5 shows a sample hierarchy configuration. The lower portion of the diagram illustrates how this sample configuration allows you to access various displays from the graphic display GRMIXER.

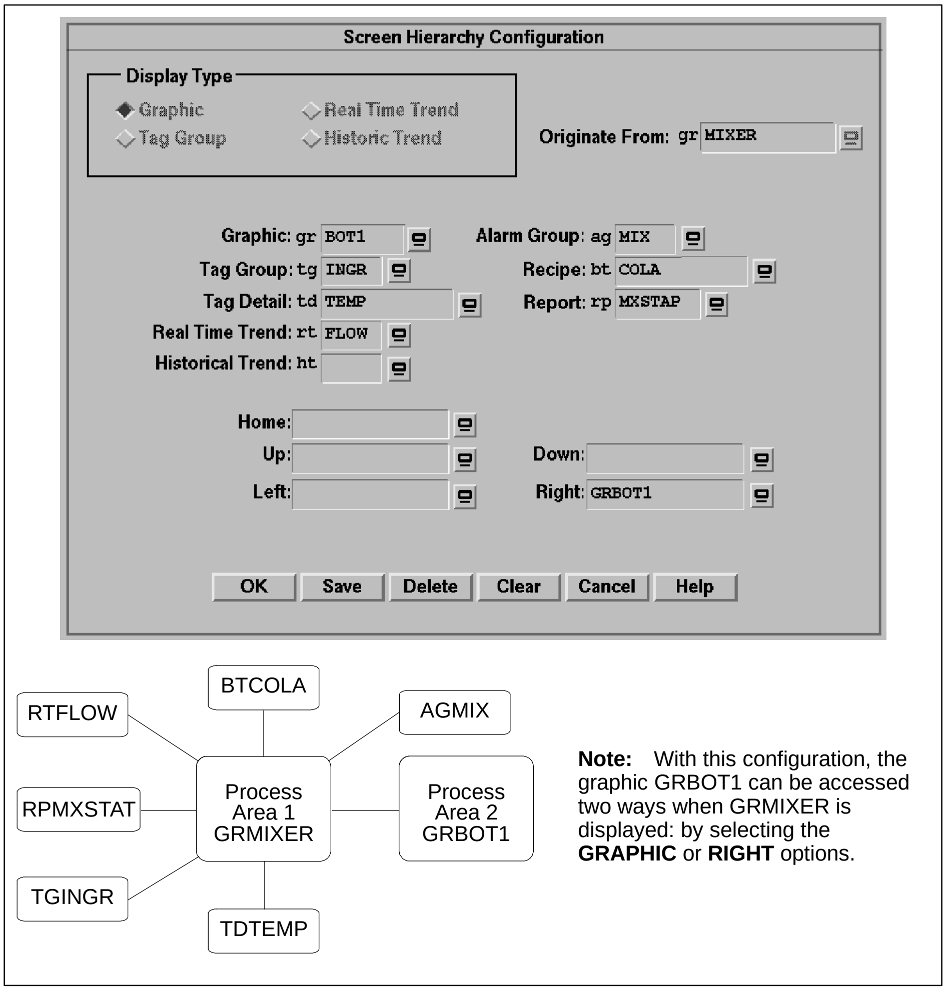


Figure 11-5 Sample Hierarchy of Associated Displays

NOTE: Configuring the directional navigational buttons (Up, Down, Left, Right) for one display does not imply the opposite navigational direction for an associated display. In Figure 11-5, if you press the Right button from GRMIXER, GRBOT1 is displayed, but pressing the Left button from GRBOT1 does not redisplay GRMIXER unless you have so configured the Left button in the screen hierarchy for GRBOT1. You can always use the Previous button to return to the previous display.

Accessing Displays from the Hierarchy

To access associated displays in a screen hierarchy, the operator selects one of the hierarchy items in the navigation area of the screen, or presses the appropriate key (**Report**, for example) on the operator keyboard.

11.4 Designing Screen Layouts

Understanding OSx Windows

PCS Releases 3.1.1 and greater allow you to display three graphic windows at a time: the main graphic window, and two subgraphic windows. The main graphic window is the only one for which you can configure a screen hierarchy and is the only one that responds to the hierarchy keys on the operator keyboard.

Table 11-2 lists the types of graphic windows, whether they can appear as a main window or as a subgraphic window, and whether they can be resized and repositioned. Keep in mind that the fonts used on the standard OSx screens are not scalable and may become distorted when resized.

Table 11-2 Graphic Window Types

Type	Prefix	Type Can Be Used in This Window		Can Be Resized and Repositioned
		Main	Subgraphic	
Alarm Group [1,2]	AG	✔		
Alarm Summary [1,2]	AS	✔		
Batch [1,3]	BA	✔		
Graphic	GR	✔	✔	✔
Tag Detail	TD	✔	✔	✔
Tag Group [1]	TG	✔		
Recipe [1,2]	BT	✔		
Report [1,2]	RP	✔		
Trend (Historical)	HT	✔	✔	✔
Trend (Realtime)	RT	✔	✔	✔

1 When writing a command to display this window type, always use the –M flag to send the window to the main graphic window.
2 This window appears in addition to the main graphic, rather than replacing it.
3 When you select the Batch Navigation icon at runtime, the screen displays the tag detail for the unit tag associated with the batch (default). If you have associated a graphic with the batch, the screen displays the graphic instead. You associate a graphic with a batch within the BCL code, using the BA prefix when referencing the graphic.

To move, resize, or close a display, see the appendix on navigation in the *SIMATIC PCS 7 OSx System Administration Manual*. To add a graphic, historical trend, real-time trend, or tag detail to a location or size other than the default, see the **add** command in the appendix on OSx commands in the *SIMATIC PCS 7 OSx Graphical Editor Manual*.

NOTE: Graphics that were created on PCS/OSx prior to Release 3.1.1 do not automatically have the graphic display menu and thus cannot be easily moved, closed, or resized. If you have such graphics and want to add this menu to them, contact the Siemens Energy & Automation, Inc., Technical Services Group in the U.S.A. at 800-333-7421. Outside the U.S.A., call 49-911-895-7000.

Accessing OSx Windows

An operator can access a window at runtime by either of two methods: by clicking on the Directory icon and selecting a window type (Figure 11-6), or by clicking on a button or menu item that is configured to display the window (Figure 11-7). See the appendix on OSx commands in the *SIMATIC PCS 7 OSx Graphical Editor Manual* for information on commands for adding and deleting windows.

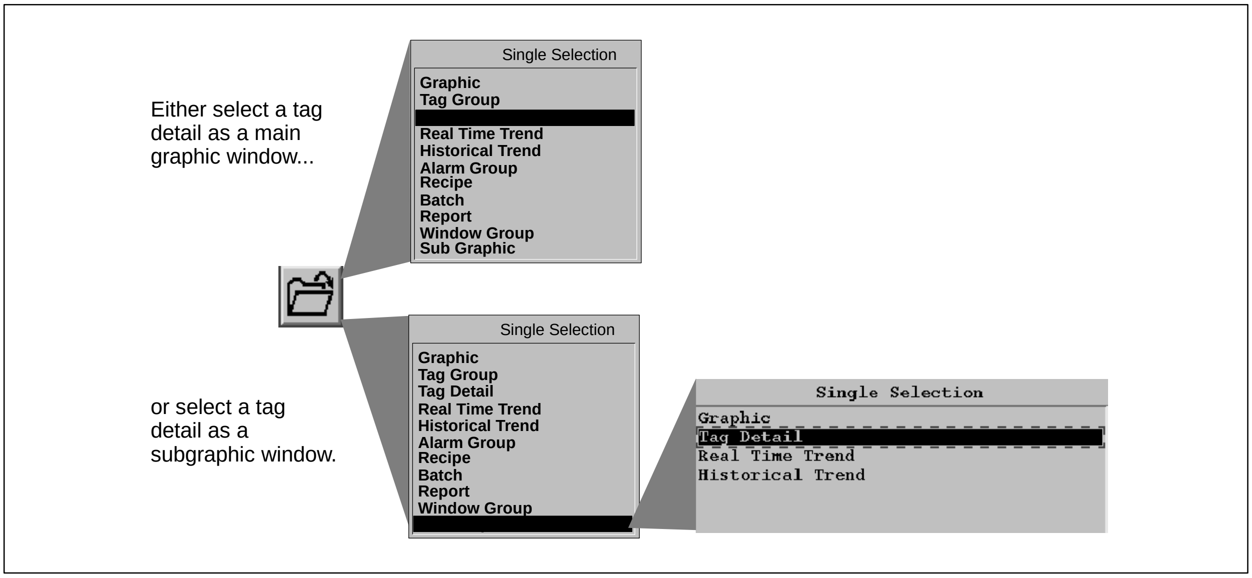


Figure 11-6 Accessing a Window from the Directory Icon

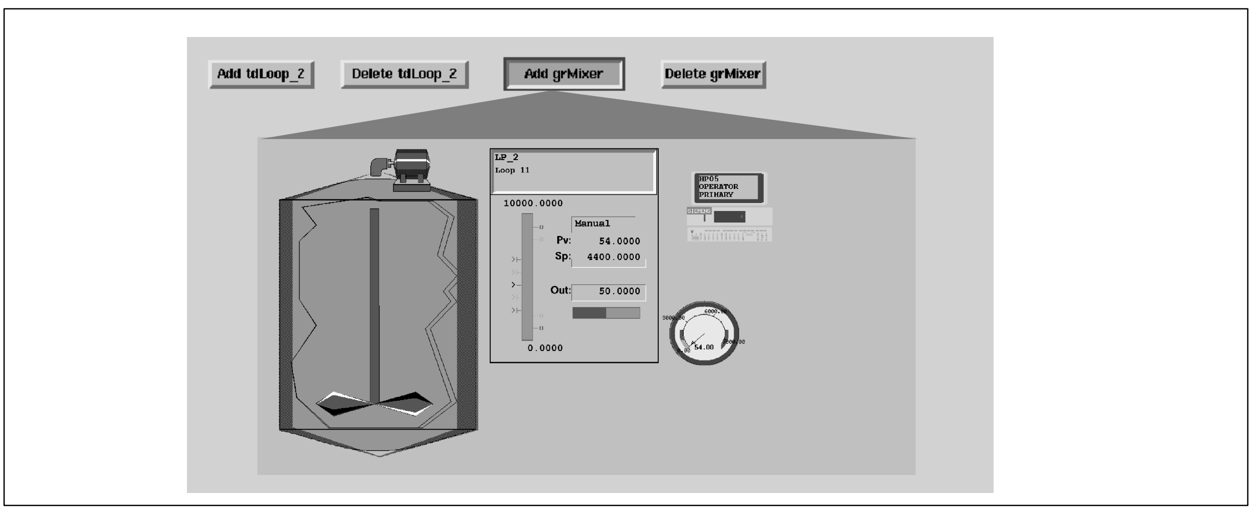


Figure 11-7 Accessing a Window from a Toolbar Button

Designing Screen Layouts (continued)

Arranging OSx Windows

You can arrange the multiple graphic windows in a variety of layouts on the screen (Figure 11-8). To customize your layout, you can designate the size and location of the graphic, tag detail, and trend window types in the following ways.

- You can designate the size and position of graphic windows by editing the Format Properties panel when you create or edit the graphic. Refer to the chapter on basic functions in the *SIMATIC PCS 7 OSx Graphical Editor Manual*.
- You can designate the size and position of graphic windows when you configure the Add command in a DDO, such as Button or Menu. Refer to the appendix on OSx commands in the *SIMATIC PCS 7 OSx Graphical Editor Manual*.

OSx determines the initial size and position of tag and alarm groups, reports, recipes, batch windows, and the alarm summary. You cannot reconfigure these, but you can move the alarm summary, alarm groups, reports, and recipes by clicking the left mouse button on the title bar of the window and dragging it to a new position.

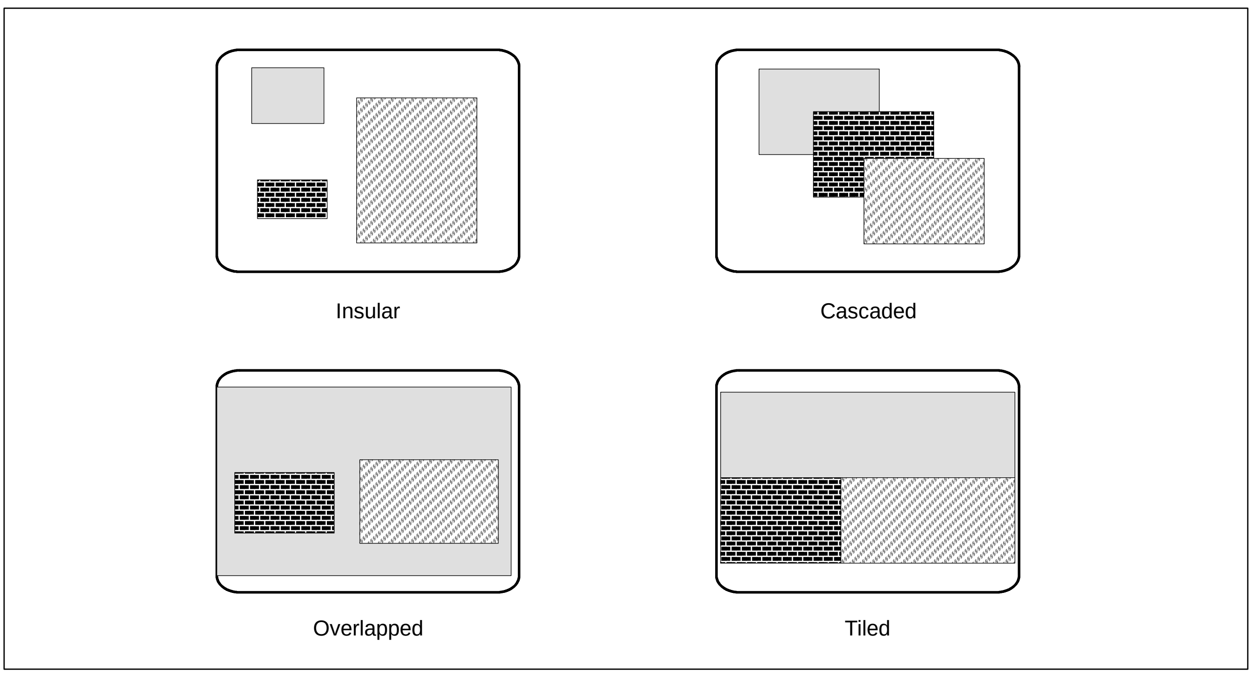


Figure 11-8 Options for Screen Layouts

Chapter 12

Configuring Action Requests

12.1 Understanding Action Requests

Overview

In most instances, a batch process runs with minimal human interaction. Occasionally, people need to make decisions about a process while it runs. To accomplish this task, OSx uses action requests that enable an operator to interact with a process.

Action requests are OSx components that you define to control a process. With action requests, you can actively and effectively interact with a batch process from a central location.

During batch processing, action requests provide the following interactions:

- They inform you about the status of the process.
- They allow you to stop the process and perform quality checks, product analysis, and other hands-on tasks.
- They prompt you for specific responses and acknowledgements.

Figure 12-1 demonstrates where action requests fit in a process.

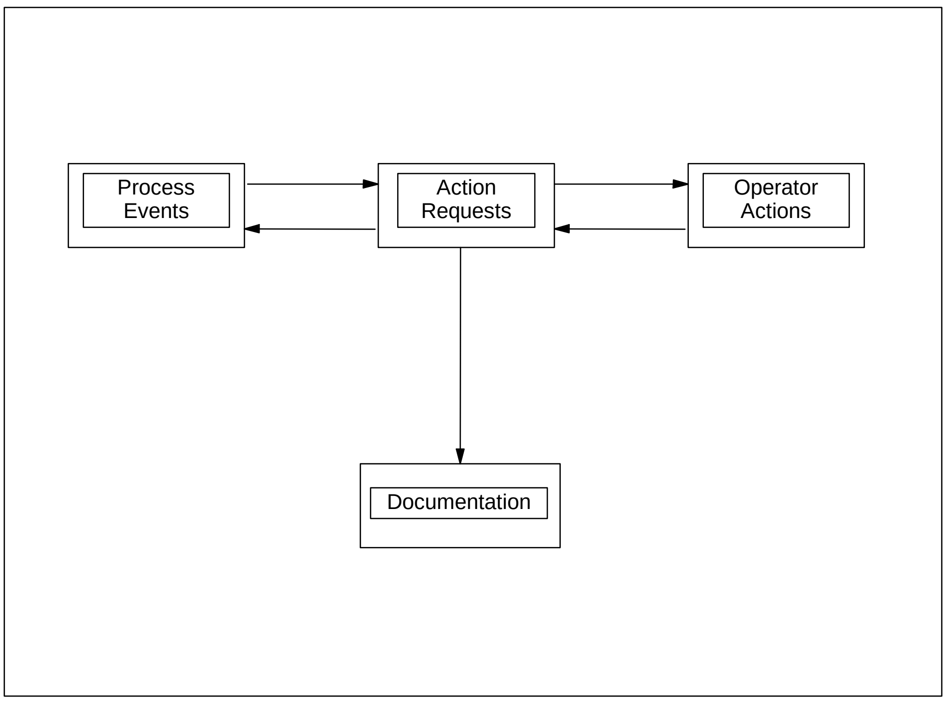


Figure 12-1 Action Requests and Process Interaction

An action request occurs when a defined process event is triggered. If the system is in the Operate state when an event occurs, it displays the action request icon in the lower right-hand corner of the OSx screen and places information about the request on the Operator Change Log, which logs changes made to the OSx system.

The action request icon tells the operator two things.

- An action request is triggered and needs attention.
- The severity of the request is either Urgent (red) or Normal (yellow), and, therefore, may or may not require immediate attention.

When the operator selects the icon, a summary screen appears that allows the operator to respond to the action request. Depending on the request that is triggered, the following responses are possible:

- Choose an option from up to four pre-defined choices.
- Enter an attribute value for a specific device (for example, a setpoint value of 500 psi for a pressure tank).
- Acknowledge that an event has taken place or that a process is ready to start.
- View information about the process.

As the operator responds to an action request, the Operator Change Log records the action request, the response, and any error that has occurred. See Chapter 14 on OSx logs for more information about the Operator Change Log.

Creating Action Requests

SIMATIC PCS 7 OSx supports 3,000 action requests. Before you can use an action request, you must first create one. Use the following procedures to create an action request:

- Decide what parts of your process benefit by using an action request. For example, if you enter a reactor temperature or add a catalyst, an action request helps you accomplish this task.
- Make sure the controller running the process is programmed, and that the tags you plan to use as event triggers are configured.
- Define the action request using the Action Request Configuration screen (Figure 12-2). Refer to Section 12.2 on adding an action request for more information.
- Check the action request after you have created it to make sure it is correct.

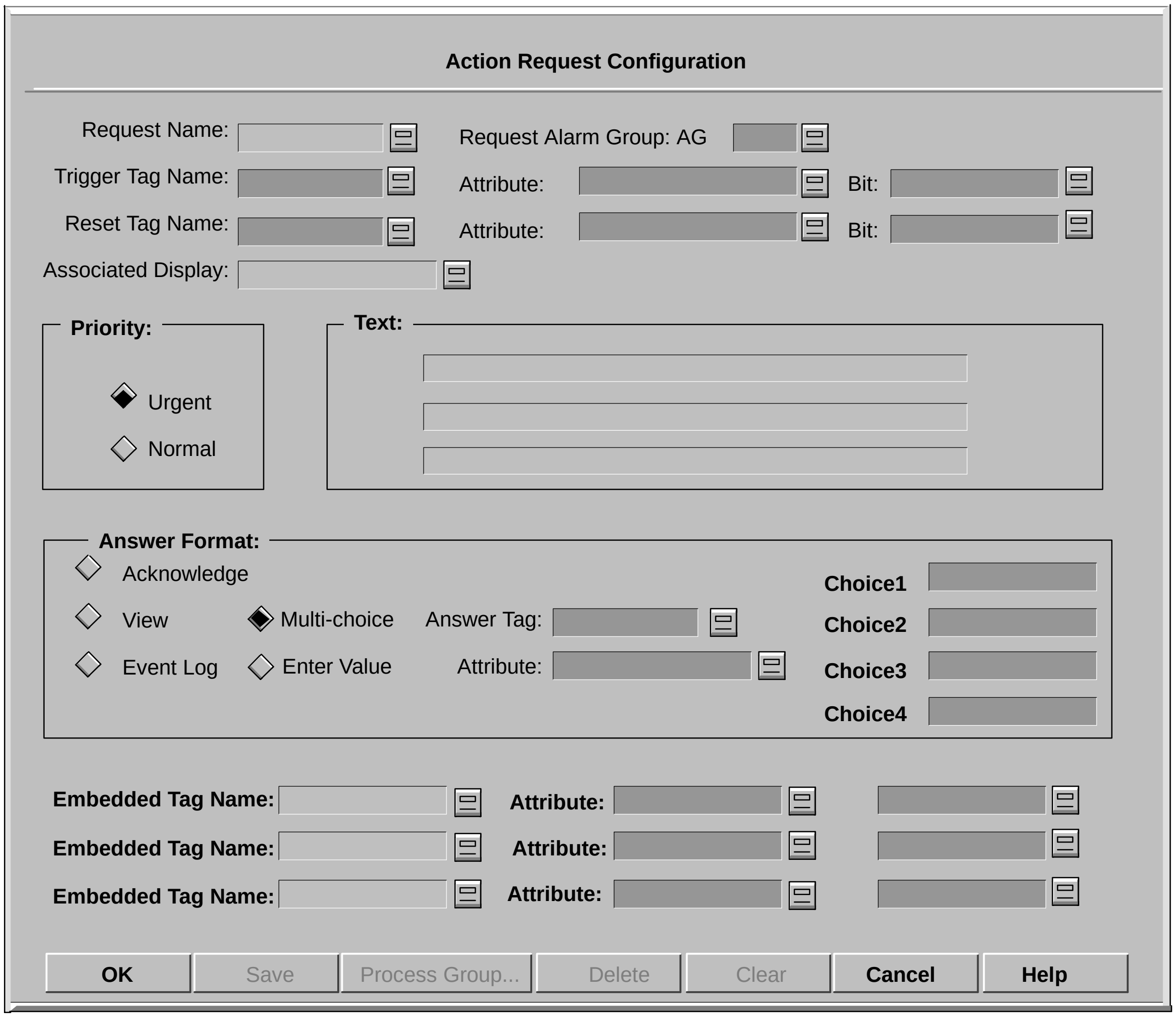


Figure 12-2 Action Request Configuration

Responding to an Action Request

After you define and save action requests, they are ready to be used by the process. With the system in Operate state, the operator can answer action requests after they are triggered.

To respond to an action request, follow these steps.

1. Click the action request icon. The screen displays the Action Request Summary dialog box in the lower left part of the screen (Figure 12-3).

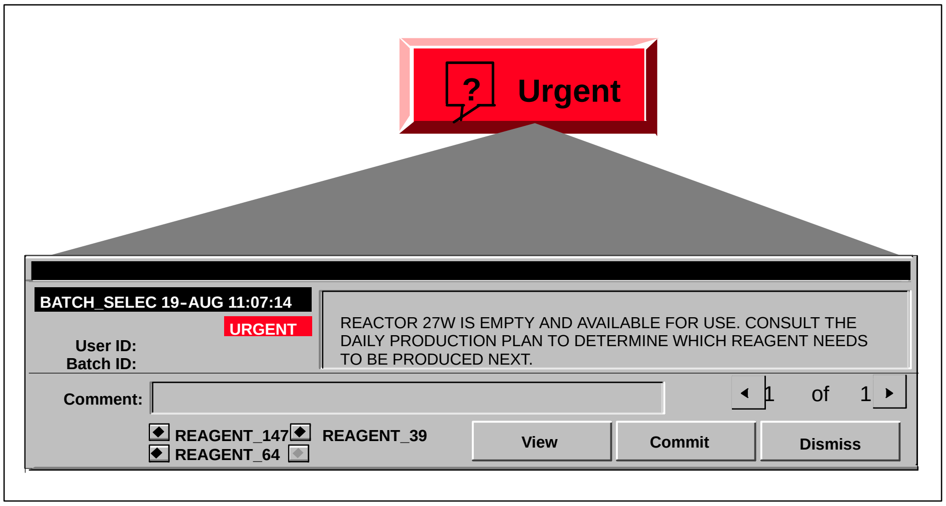


Figure 12-3 Action Request Summary Dialog Box

2. Select an action request entry and respond with the appropriate answer. The left and right arrow buttons allow the operator to access other action requests. The action request package allows the operator to define five types of requests.

   - The Multi-Choice request consists of up to four different predefined answers, from which the operator chooses one.
   - The Enter Value request prompts the operator to enter a new value for a particular tag attribute.
   - The Acknowledge request asks the operator to acknowledge that a request occurred.
   - The View request answers itself automatically when the operator displays it on the screen.
   - The Event Log request does not turn on the action request icon. However, it does send a message to the system log when triggered.

3. Commit a response or dismiss it.

   Committing an action request sends the response to the database, where it is used by the process. Dismissing an action request stops the answer process, but does not cancel the request.

   OSx commits the View request automatically when an operator views it.

4. Dismiss the Action Request Summary screen and continue to monitor the process.

12.2 Creating Action Requests

Preparing to Configure Action Requests

To configure or modify an action request, you need either the Database Administration or the Action Request Configuration security privilege. You can add new action requests or modify existing ones while the system is in the Offline state. When the system is in the Operate state, you can add new action requests, and you can view or modify certain fields of existing action requests. When you are configuring action requests, you cannot change the system state.

As you define each action request, make sure the tags you use match the type of activity you need to accomplish. For example, the attribute that resets the request must have write permission. If you try to reset a read-only attribute, OSx generates an error message. See Section 12.7 for more information about read-only attributes. Chapter 3 lists attributes for each tag and indicates whether each attribute is read-write or read-only.

You cannot delete a tag that is configured in an action request. This includes trigger, reset, answer, or embedded tags. If you attempt to delete a tag that is in an action request, OSx generates an error. You must first either delete the action request or replace the tag in the action request with another tag. In the case of a non-essential embedded tag, you can simply remove the tag from the action request. Then you can delete the tag from OSx.

NOTE: When you configure a tag for an action request, whether it is a trigger tag, reset tag, or answer tag, be sure that you do not also configure it as a component in a recipe. Otherwise, recipe download may interfere with the action request operation.

Before you configure a request, carefully plan your controller programs and your tag characteristics. The controller program performs the following functions during normal operation.

- It sets the trigger bit to trigger the action request in OSx, and sets the reset bit to notify the controller program when the action request is answered. If a separate reset bit was configured, it must be set before the request is triggered. See Section 12.7 for information about trigger and reset logic.
- It detects state changes of the reset bit.
- It resets the trigger bit when the state of the reset bit changes from one to zero.
- It resets the trigger and reset bits if the request becomes invalid before it is answered by the operator.

Creating Action Requests (continued)

Configuring a New Action Request

To add an action request, select **Data->Action Request** from the main menu bar, and fill in the fields on the Action Request Configuration screen (Figure 12-4) described on the following pages.

Request Name Enter a unique name for the action request. It must be the first field you enter on the action request. You can use up to 12 characters; however, when you assign the request to a process group, the Request field in the Process Group Assignment dialog box displays only the first 7 characters of the action request name. The space, the character combinations `->` and `<-` and the characters ; \ , ” are invalid for action request names.

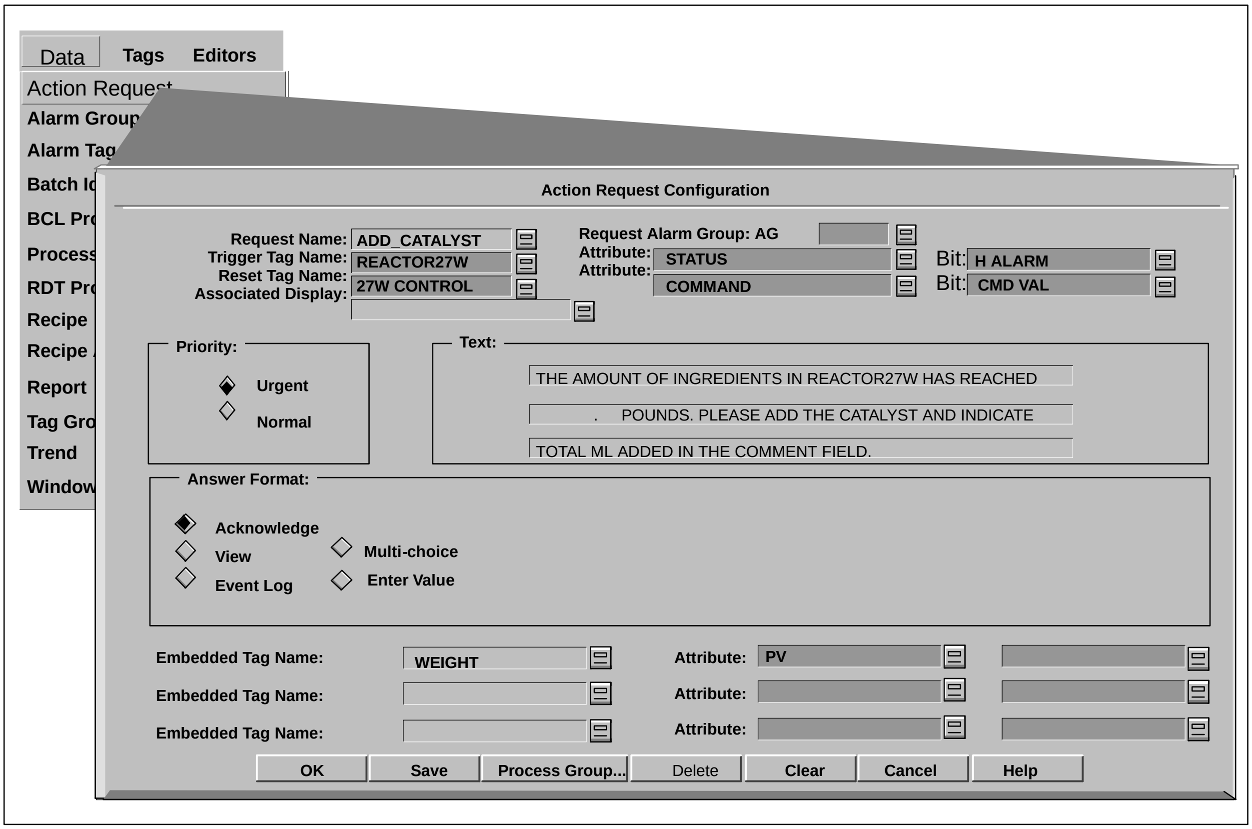


Figure 12-4 Accessing the Action Request Configuration Screen

Request Alarm Group You can display mini-windows for alarms and action requests with a graphic. To display this action request with a graphic, enter the name of the alarm group that you associated with the graphic in this field. See Chapter 15 for more information.

Trigger Tag Name Select the name of the tag that is to be used to trigger the action request. The name in this field must be a configured tag. See Section 12.7 for more information about trigger logic.

Trigger Attribute Select the tag attribute that is to be used to trigger the action request. This field is dependent on the Trigger Tag Name field. This attribute must be a data type that supports logical bits (for example: BIT16, scalar, and status). Click the long-list display tool to the right of the Trigger Attribute field for a list of valid attributes.

Trigger Bit Bit mask or name that is to be used to trigger the action request. A valid bit consists of one of the following values (see Chapter 3 on defining tags):

- Bit Name—the bit depends on the tag and attribute that you select. Click the long-list display tool to the right of the Trigger Bit field for a list of valid bit names.
- Hex Value—a hexadecimal number beginning with 0x or 0X that specifies a single bit location (for example: 0x0001, 0x0800).
- Decimal Value—a number that specifies a single bit location. Each location is a power of two beginning with the number one (for example: 1, 2, 4, 8, 16, 32, ..., 32,768).

Reset Tag Name Enter the name of the tag that is responsible for resetting the trigger tag and indicating the status of the request. If the trigger tag, attribute, and bit have read and write permission, they can also be used to reset the action request. To use the trigger fields as resets, allow the unmodified reset fields to default to the trigger values. See Section 12.7 for more information about reset logic.

Reset Attribute Enter the name of the reset tag attribute. This attribute controls the trigger tag reset and indicates the status of the request.

Configuring a New Action Request (continued)

Refer to Figure 12-4 and continue by filling in the fields described below.

Reset Bit Enter the bit mask, value, or name of the reset bit. This bit controls the trigger reset and indicates the status of the request. A valid bit consists of one of the following values (see Chapter 3 on defining tags):

- Bit Name—must be a bit name that is assigned to this tag.
- Hex Value—a hexadecimal number beginning with 0x or 0X that specifies a single bit location.
- Decimal Value—a number that specifies a single bit location. Each location is a power of two beginning with the number one (for example: 1, 2, 4, 8, 16, 32, ...32768).

CAUTION

Read-only tag attributes cannot be used in the Reset or Answer tag of the action request configuration. If you change specific tag attributes from non-networked to networked, OSx does not allow you to answer or reset the request.

An unanswered action request could halt your process. OSx produces an error message if this condition occurs. See Section 12.7 for more information about read-only attributes.

Do not use read-only tag attributes in the Reset or Answer tag of the action request configuration.

Associated Display Select the name of a graphic or tag detail that you want to be able to access from the Action Request Summary dialog box. When the operator selects the **View** button on the Action Request Summary, this graphic or tag detail displays on the screen.

Priority Select the priority of the action request. Urgent and Normal are used to determine the color of the action request icon and the order that the action requests are listed on the Action Request Summary screen. The icon flashes red for urgent requests and yellow for normal requests. Urgent requests are always listed first on the summary screen.

Text Enter the text that you want to be displayed when the action request is triggered. The characters ; \ ” are invalid for action request text. This field is up to three lines long (55 characters per line). The first line of text appears next to the request name in the select list that appears when you click the long-list display tool to the right of the Request Name field. You must configure it for each request.

The Text field can contain embedded formats that allow actual values from the database to be substituted in place of the formats when the text is displayed on the summary screen. You can enter a maximum of three embedded formats in this field. See Section 12.5 for more information about embedded formats.

Answer Format Select the type of answer that is to be displayed when the action request is displayed on the Action Request Summary screen.

- Acknowledge requires the operator only to acknowledge the message.
- View does not require a response. When the action request appears on the Action Request Summary screen, OSx automatically answers it.
- Event Log does not turn on the action request icon when triggered. The Event Log answer format sends a message to the system log when triggered and then automatically resets itself.
- Multi-Choice allows the operator to choose from up to four options having predefined answer values.
- Enter Value requires the operator to enter a value from a specified range for attributes with a defined range, such as a loop setpoint.

Answer Tag Enter the name of the tag that receives the response to the action request. This field is available only when the Multi-Choice or Enter Value format is selected. This field is not applicable for the Acknowledge, View, or Event Log answer formats.

Answer Attribute Enter the name of the tag attribute that receives the response to the action request. This field is not applicable for the View, Acknowledge, or Event Log answer formats. The data type of the answer attribute must match the answer format you choose. For example, a Multi-Choice format is not compatible with an attribute having a string data type. Also, avoid using a read-only attribute for this field. If you enter a read-only attribute, OSx cannot accept the operator's response, and it generates an error message. Table 12-4 at the end of the chapter lists the valid reset, trigger, and answer bits for all the tag types that you can use in configuring action requests.

Configuring a New Action Request (continued)

Refer to Figure 12-4 and continue by filling in the fields described below.

Choice Enter the text for the Multi-Choice answer. The characters ; \ ” are invalid for this text. These choices are the operator’s options when the action request is triggered. Each choice has a corresponding hexadecimal value that is written to the database when the answer is selected and committed. The predefined hexadecimal values are

- Choice 1—0x1000
- Choice 2—0x2000
- Choice 3—0x4000
- Choice 4—0x8000

Embedded Tag Name Enter the tag name that is to be used with the embedded format within the text. Embedded Tag 1 corresponds to the first embedded format, Embedded Tag 2 corresponds to the second embedded format, and Embedded Tag 3 corresponds to the third embedded format.

Embedded Attribute Enter the tag attribute that is to be associated with the embedded formats in the text field. Each of the embedded tags has a corresponding attribute.

Mapname The Mapname fields, in the lower right corner of the Action Request Configuration screen, have no labels unless you enable them by entering a tag:attribute combination that has a map associated with it, such as a COMMAND or STATUS attribute.

In the Mapname field, enter the name of the map file that is to be associated with the embedded values in the Text field. Each of the embedded tags can have a different map file associated with it, but a map file is not allowed in all cases. If the data type of the embedded attribute supports the use of a map, OSx enables the Mapname field. See Section 12.6 for information about embedded map files.

NOTE: You can make changes in the text field and in fields associated with embedded tags of an existing action request while OSx is in the Operate state.

Assigning Action Requests to Process Groups

A process group does not have to have a description before you assign an action request to it. Therefore, you can assign an action request to one or more process groups, whether or not they have descriptions, and then enter the descriptive text for the process groups at another time. Refer to Chapter 2 for detailed information about how to assign action request membership in a process group.

NOTE: Associate an action request with at least one process group. If you do not associate an action request with any process group, then no user can acknowledge the action request.

You can assign new action requests to process groups or modify the process group assignments for existing action requests while the system is in the Offline state. When the system is in the Operate state, you can assign new action requests to process groups, and you can view, but not modify, process group assignments for existing action requests.

To simplify configuration, an action request is linked by default to the trigger tag process group assignment. To change membership to a process group, follow these steps.

1. Select the **Process Group** pushbutton on the Action Request Configuration display (Figure 12-5).
2. Select **Yes** to link the action request with the associated process group. Select **No** to disable the link between the action request and the process group.
3. Click **OK** to save the process group assignment.

NOTE: For proper operation, ensure that the trigger tag, reset tag, answer tag, and embedded tags are associated with the same process groups.

Always set unused process groups to **No**. Otherwise, unintended memberships may occur between users, tags, alarm groups, and action requests.

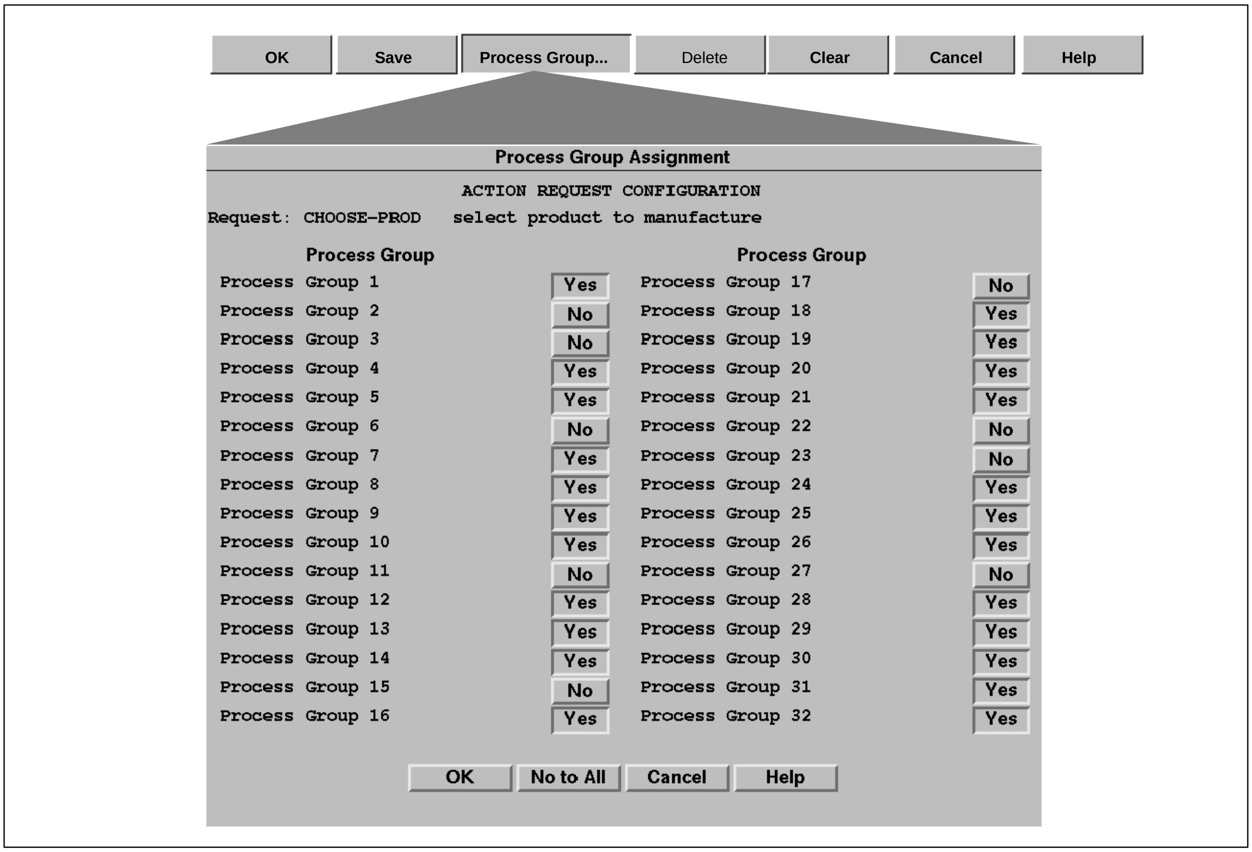


Figure 12-5 Process Group Assignment Display

Modifying an Action Request

To modify an action request that you have already created, follow the steps below:

1. Select **Data–>Action Request** from the main menu bar. The Action Request Configuration screen appears.
2. Click the long-list display tool to the right of the Request Name field. A list of configured action requests appears.
3. Double-click the name of the action request that you want to modify. The name appears in the Request Name field, and the other fields on the Action Request Configuration screen fill in automatically.

   In the Offline state, if you have the security privilege to modify action requests, you can now modify any fields that you want to change.

   In the Operate state, only the text fields and fields associated with embedded tags can be modified.

   If you change the Request Name field, the following prompt appears:

   Create a new action request? (A “No” response will change the name of this one.)

   Respond **Yes** to create a new action request, **No** to simply rename the current request.
4. Click **Save** to save the modified action request, or **OK** to save and exit.

Deleting an Action Request

To delete an action request, follow the steps below:

1. Select **Data–>Action Request** from the main menu bar. The Action Request Configuration screen appears.
2. Click the long-list display tool to the right of the Request Name field. A list of configured action requests appears.
3. Double-click the name of the action request that you want to delete. The name appears in the Request Name field, and the other fields on the Action Request Configuration screen fill in automatically.
4. Click the **Delete** button, and respond **Yes** to the prompt for confirmation. The action request is deleted.

Viewing an Action Request

To view an existing action request in either the Offline or the Operate state, follow the steps below:

1. Select **Data->Action Request** from the main menu bar. The Action Request Configuration screen appears.

2. Click the long-list display tool to the right of the Request Name field. A list of configured action requests appears.

3. Double-click on the name of the action request that you want to view. The name is entered in the Request Name field, and the other fields fill in automatically.

Action Request Mini-Windows

You can attach an action request to a graphic with the mini-windows feature. When you enable mini-windows from the Event Preferences screen, any action request attached to a graphic is displayed in a list at the lower right of the graphic if the request has been triggered and not reset or answered. Refer to Chapter 15 to configure mini-windows for an action request.

12.3 Example of Adding an Action Request

To help you understand action requests and their use in process control, the following example appears throughout the rest of this and the next chapter. The example shows practical applications for adding, changing, deleting, and answering action requests.

The example process consists of a reactor tank where two ingredients are combined and heated to create one of three different reagents. A single controller runs the process. The controller uses a temperature loop to control the temperature of the reagent by modulating the steam supply that heats the vessel (Figure 12-6).

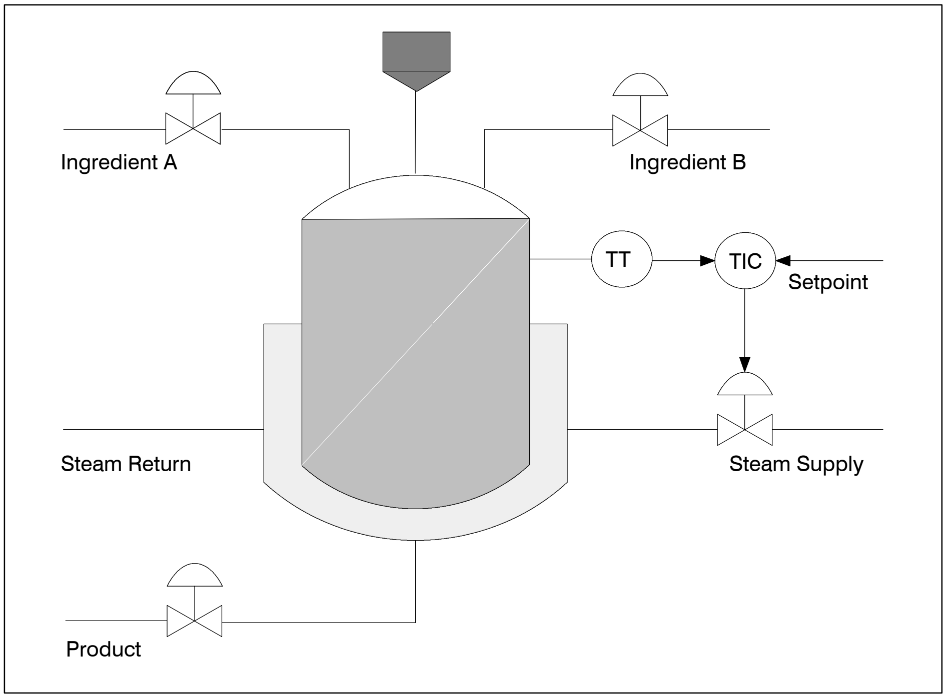


Figure 12-6 Sample Reactor Process

Preparing to Add a Request

First, you need to figure out the best places for action requests in this process. For this example, there is human interaction in the following situations:

- When the reactor is ready to combine a new batch of ingredients, the operator decides which of the reagents should be created.
- When the amount of reagent reaches a predetermined amount, the operator stops the process, adds a catalyst, and allows the process to resume.
- When the reactor is full, the operator enters a temperature to control the characteristics of the reagent mixture.
- When the mixture is done, the operator opens the outlet valve to drain the combined reagent.
- If the batch is temporarily suspended at some point, the event is noted in the daily log.

Each of the situations listed above corresponds to one of the action request types:

- Multi-Choice (Operator chooses reagent 147, reagent 64, or reagent 39)
- Acknowledge (Operator acknowledges that the amount of reagent has reached a specified number of pounds, and that the correct amount of catalyst has been added to the mixture)
- Enter Value (Operator enters the correct soak temperature for the chosen reagent)
- View (Note that the reactor process is complete, so the operator or the controller can open the outlet valve and empty the reactor tank)
- Event Log (The event is automatically logged in the daily log file; no operator interaction is required)

Now that you have figured out what action requests you need to interact with your process, you are ready to define the action requests. The next several pages describe the procedures for creating the action requests listed above.

Adding the Multi-Choice Request

The Multi-Choice request allows the operator to select the reagent the process makes. Define the tags that you want to use to configure the action request. Then follow the steps below.

1. Select **Data->Action Request** from the menu bar. The Action Request Configuration screen appears.

2. Enter `BATCH_SELECT` in the Request Name field. This field defines the name of the action request.

3. Enter data in the rest of the fields so the screen looks like Figure 12-7.

4. Click **Save**. OSx saves the action request.

NOTE: Set Multi-Choice values carefully when you configure action requests. If an operator tries to commit a Multi-Choice value that is out of range, the **Commit failed** message appears, and all the other choices are desensitized. The operator must leave that action request and return to it before making another selection.

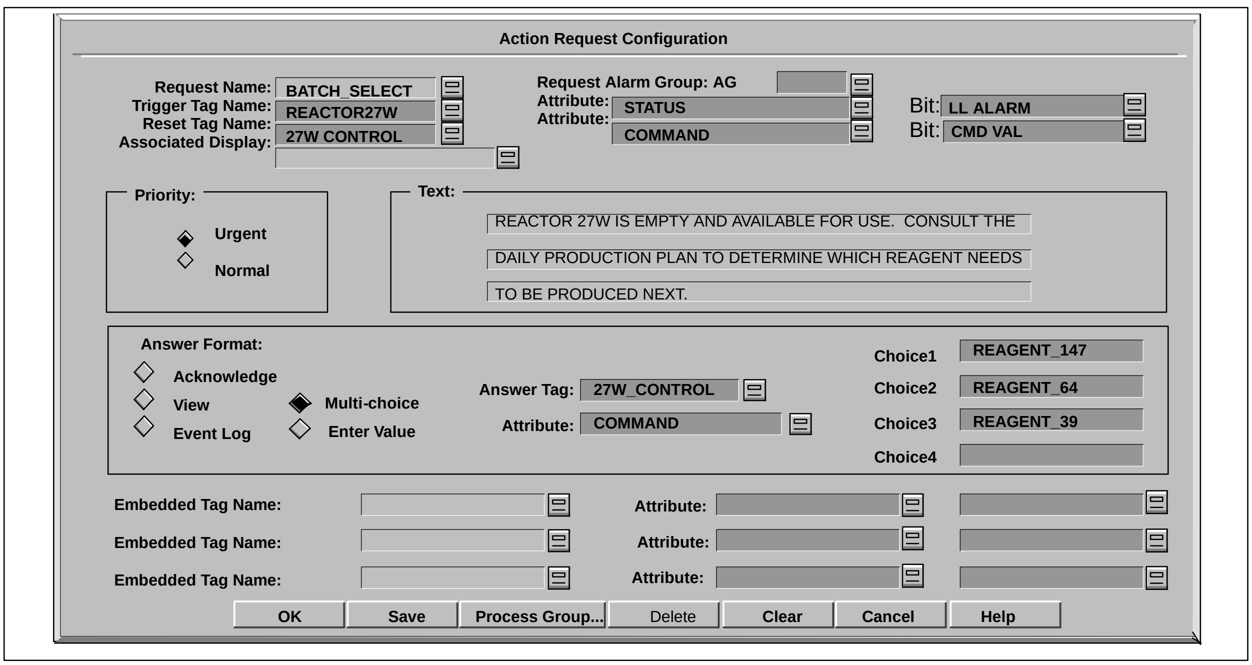


Figure 12-7 Multi-Choice Action Request (BATCH_SELECT)

Adding the Acknowledge Request

Now you are ready to define the Acknowledge action request. This request notifies the operator when the process has stopped after the reactor is filled with a specified amount of reagent. At this point, the operator adds the catalyst and acknowledges the action request; the process continues.

To create the Acknowledge request, follow these steps:

1. Select the **Clear** pushbutton. The fields of the Action Request Configuration screen are cleared.
2. Enter ADD_CATALYST in the Request Name field. This field defines the name of the action request.
3. Enter data in the rest of the fields so the screen looks like Figure 12-8.
4. Click **Save**. OSx saves the action request.

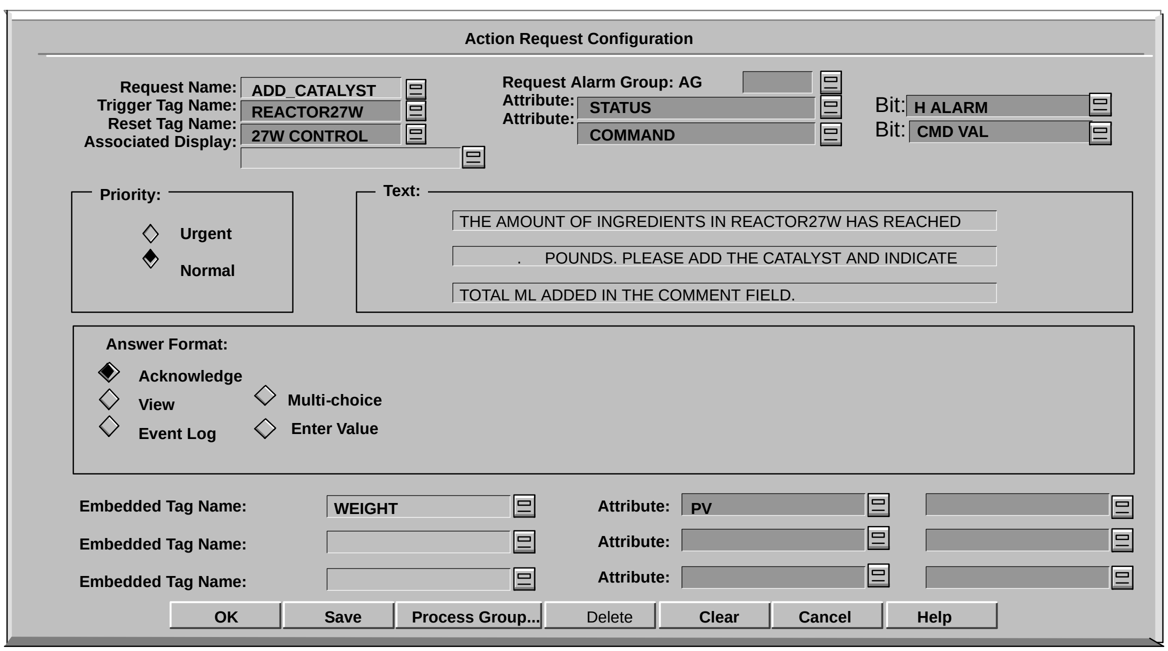


Figure 12-8 Acknowledge Action Request (ADD_CATALYST)

Adding the Enter Value Request

Define the Enter Value action request next. This request allows the operator to enter a soak temperature for the reagent batch. To create the Enter Value request, follow these steps.

1. Select the **Clear** pushbutton. The fields of the Action Request Configuration screen are cleared.

2. Enter `SET_TEMP` in the Request Name field. This field defines the name of the action request.

3. Enter data in the rest of the fields so the screen looks like Figure 12-9.

4. Click **Save**. OSx saves the action request.

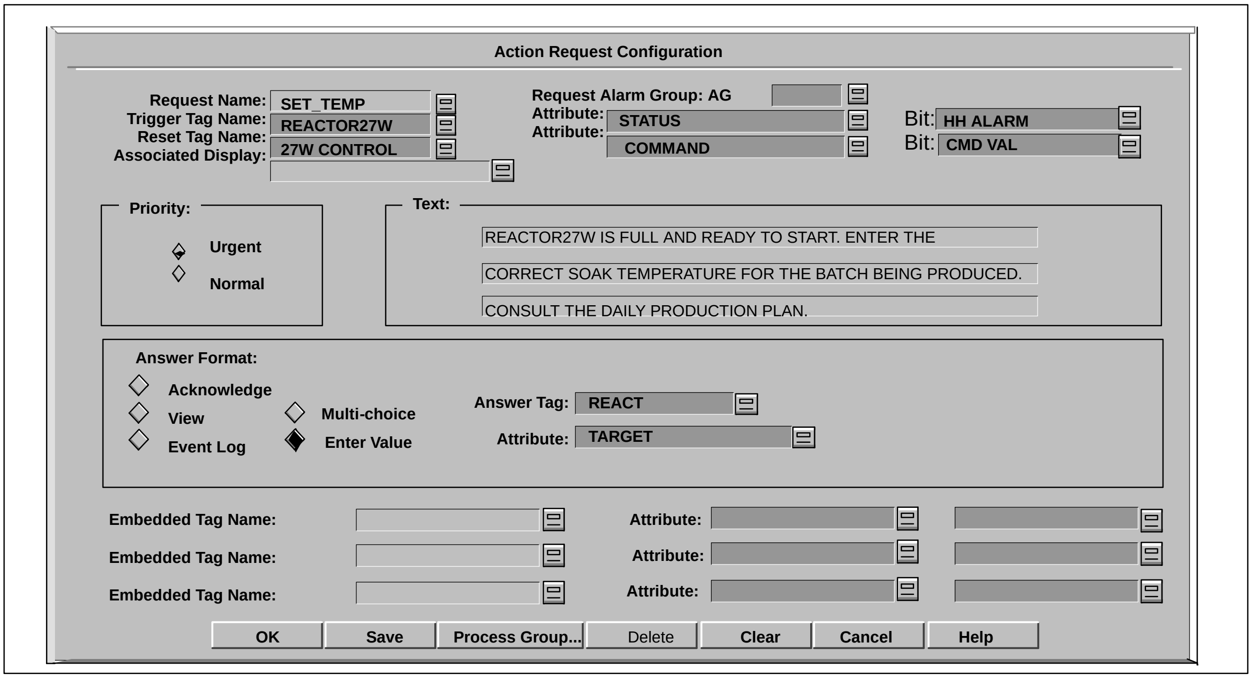


Figure 12-9 Enter Value Action Request (SET_TEMP)

Adding the View Request

Define the View action request. This request notifies you when batch processing is finished. To create the View request, follow these steps.

1. Select the **Clear** pushbutton. The fields of the Action Request Configuration screen are cleared.
2. Enter `BATCH_DONE` in the Request Name field. This field defines the name of the action request.
3. Enter data in the rest of the fields so the screen looks like Figure 12-10.
4. Click **Save**. OSx saves the action request.

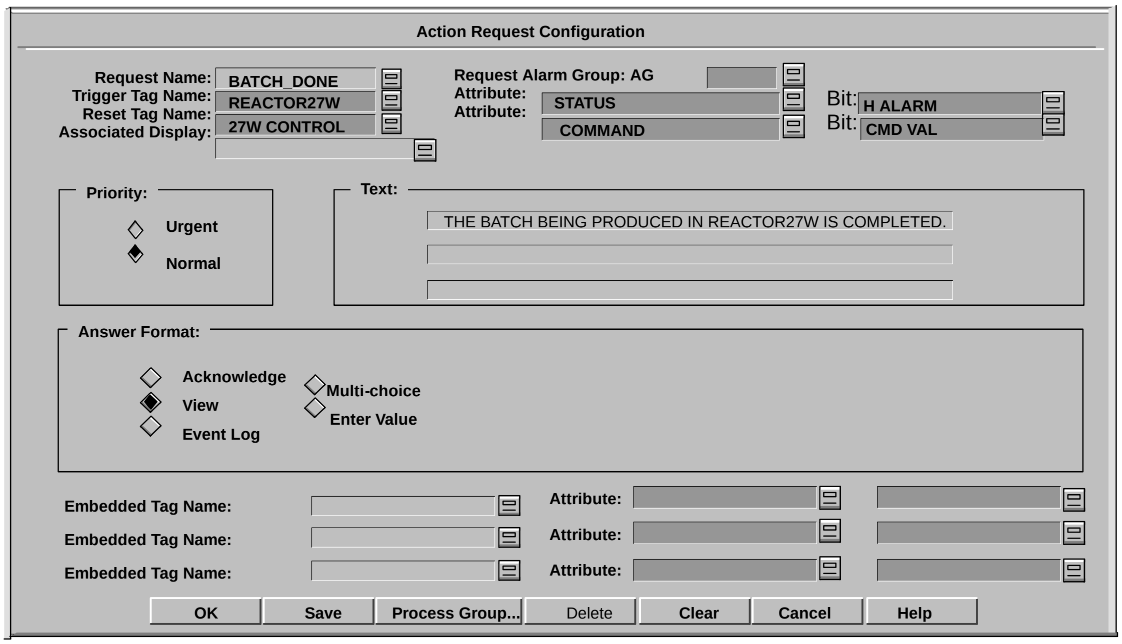


Figure 12-10 View Action Request (BATCH_DONE)

Example of Adding an Action Request (continued)

Adding the Event Log Request

To add Event Log requests, follow these steps.

1. Select the **Clear** pushbutton. The fields of the Action Request Configuration screen are cleared.
2. Enter BATCH_HOLD in the Request Name field. This field defines the name of the action request.
3. Enter data in the rest of the fields so the screen looks like Figure 12-11.
4. Click **OK**. OSx saves the action request and dismisses the Action Request Configuration screen.

Now that you have defined the action requests for the example process, you can change the system state to the Operate state so you can test/respond to defined action requests. See Chapter 13 on responding to action requests for more information about this topic.

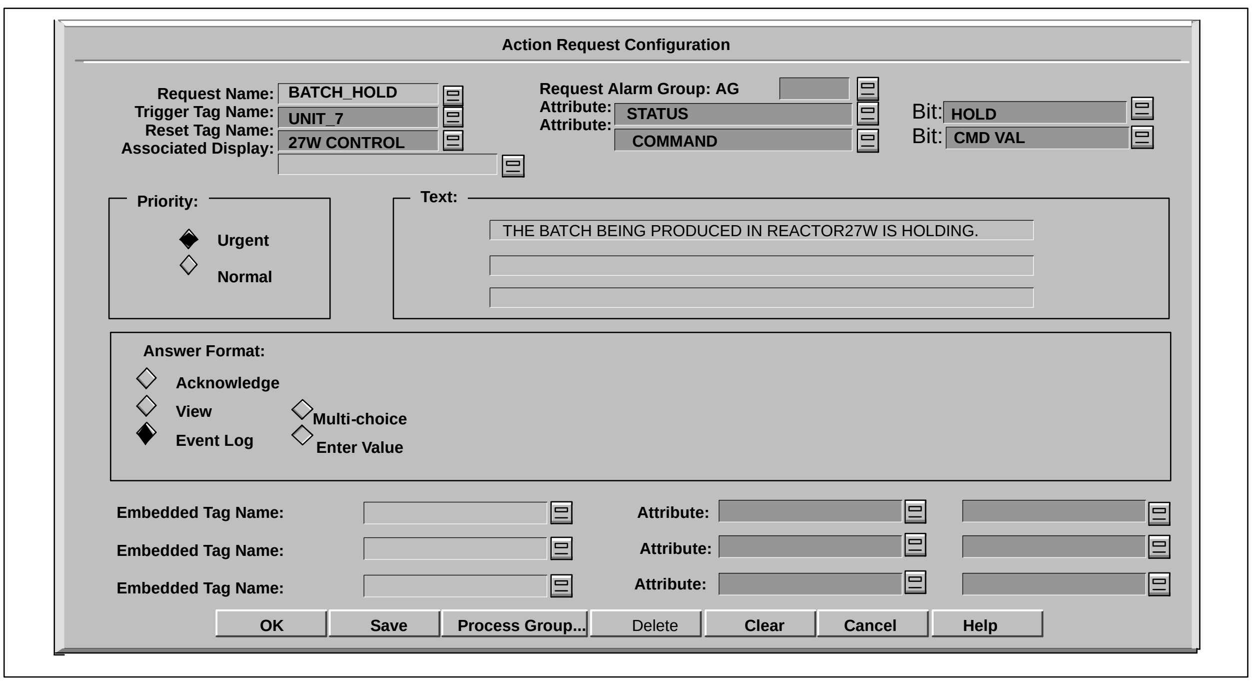


Figure 12-11 Add Event Log Request (BATCH_HOLD)

12.4 Printing the Action Request Configuration Report

Overview of the Configuration Report

The Action Request Configuration Report (Figure 12-12) lists all configuration information for all the action requests.

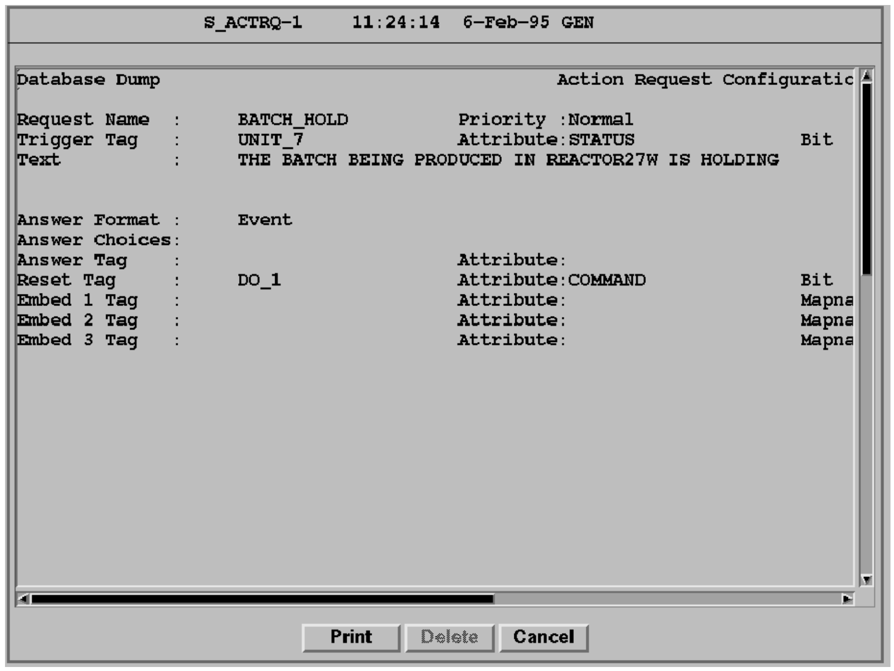


Figure 12-12 Sample Action Request Configuration Report

Accessing the Report Formats Directory

To access the Report Formats Directory, select **Data->Report** on the main menu bar. The screen displays the Reports Directory, which contains a listing of all executable reports.

The name of the Action Request Configuration Report is S_ACTRQ.

Figure 12-13 shows the names of the standard tag configuration reports. The highlighted format is the current selection.

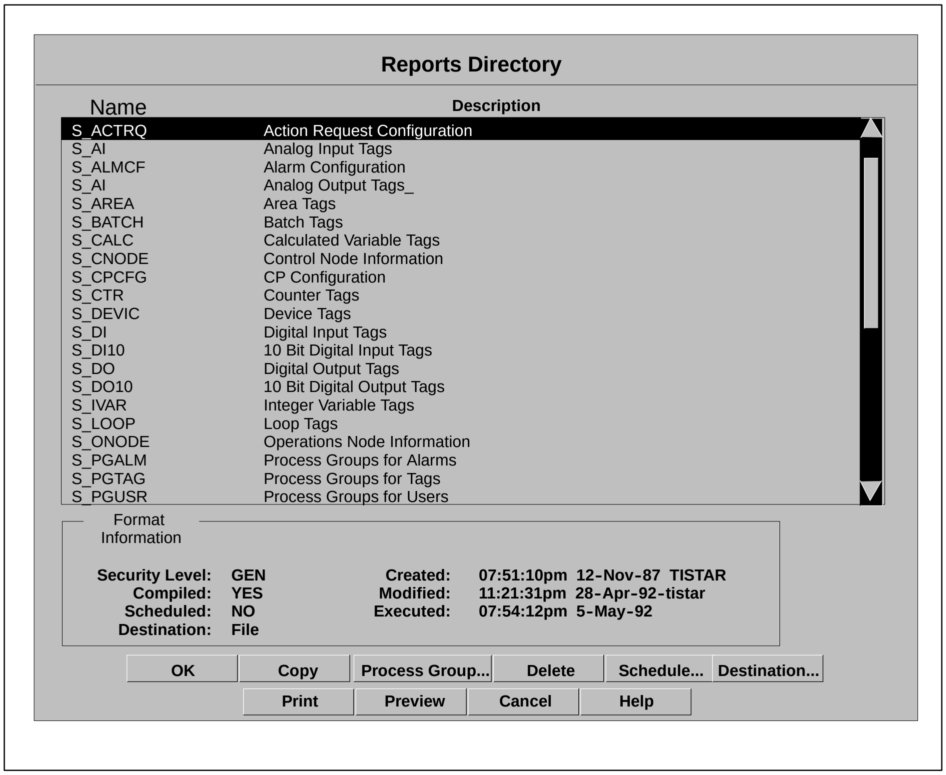


Figure 12-13 Reports Directory

Printing the Configuration Report

Like all other standard reports, you can place the Action Request Configuration Report output in a file or print it. To generate the report output, follow the steps for scheduling or previewing and printing reports outlined in the *SIMATIC PCS 7 OSx Reports Manual*. The same manual gives the steps for printing a report.

12.5 Defining Embedded Formats

Overview

Embedded formats are characters that represent variable values in the text of action requests. For example, you can use an embedded format to display the current temperature of a reactor tank in the text of its corresponding action request.

Some formats have maps that allow you to use text from the **stcm_dict** relation from the database in the action request text. Both embedded formats and their associated map files are discussed in this section.

Characteristics of Embedded Formats

Embedded formats have several characteristics that affect the way you use them in action requests. Each format contains:

- A maximum length of 30 characters.
- A minimum length corresponding to the number of characters needed to display the format value when the action request is triggered. For example, if the attribute value varies from 0 to 1,000, the minimum length of the format must be four characters.
- A minimum length of six characters if a hexadecimal status value must be displayed. More characters may be required if you are associating a map file with the embedded format.
- An associated data type that must match the type of data that OSx tries to substitute for the format.

Data Types and Their Use

For each embedded format placed in the request's text, you must define a matching tag and attribute. If the attribute has a corresponding map file, you can define it. The tag attribute corresponds to a single format that you choose from the data types in Table 12-1. The formats match up with the tag/attributes on a first-found, first-matched basis.

If you use an embedded tag value in the action request, and you want to ensure that all stations display the same value, use a bit in the tag's status attribute as the trigger, and use one of the tag attributes listed in Table 12-3 on page 12-34 for the embedded value. These attributes are linked to the RBE change event and are reported to all stations at the same time as the status attributes.

NOTE: If you receive an error message that says **Tag information configured but no embedded format found**, check the number of embedded formats configured against the number of embedded tags configured. Be sure that a floating point format includes a decimal point. Also check that the embedded format characters are correct for the data type of the embedded tag attribute.

Table 12-1 Embedded Format Data Types

Format Type	Character	Data Type
Integer	*	int
		long
		SINT16
		UINT16
		ushort
		SINT32
		UINT32
		BIT16
		BIT32
		SINT8
		UINT8
		SCALAR (no map defined)
Floating Point (not valid without decimal point)	^.^	FLOAT32
Status	$	BIT16 (status attribute)
		SCALAR (map defined)
ASCII Text	@	STRING
		CISTRING

Sample Formats

The following examples demonstrate how embedded formats and their corresponding attributes are used to define action requests.

PV ^^^^^.^^ Displays real values from 0.00 through 99,999.99 in the text. A floating point format without a decimal point is invalid. For example, ^^^ is invalid, but ^^^. or .^^^ is valid.

STATUS $$$$$$ Displays the status as 0X0000 through 0XFFFF in the text.

UNITS @@@@@@@@ Displays the text string from the units attribute in the action request text.

IVAR tag:VALUE ******* Displays integer values from –32,768 through +32,767 in the text.

12.6 Defining Embedded Maps

Characteristics of Embedded Maps

Embedded map files allow you to use text from the **stcm_dict** relation from the database in the action requests you define. The values in this file and database support the process of mapping numeric status values to character strings. This process allows OSx to present data to the operator in a meaningful and easy to understand format.

Embedded map files have the following characteristics:

- The $ (status) data type format specifies that an embedded map file can be used.
- OSx checks all map names to ensure that they exist in the STCM database table. If a map cannot be found in the database, the action request configuration utility displays an error message on the screen.
- The length of the text string determines the minimum length of the embedded format.
- In the Operate state, if you encounter a status value that is not recorded in the STCM relation, the embedded field contains the hexadecimal representation of the value (i.e., 0x0000 through 0xffff).

The appendix containing the **stcm_dict** table in the *SIMATIC PCS 7 OSx Reports Manual* lists the strings available in the database.

12.7 Using Trigger and Reset Logic in Action Requests

Overview

When defining an action request, you specify a trigger tag and a reset tag. The trigger tag denotes which attribute turns on a request. The reset tag denotes which attribute notifies the controller that the request was answered.

OSx monitors the trigger bit and responds to state changes in that bit. For example, if a status bit for a trigger tag changes from zero to one, the corresponding action request occurs.

Tag Attribute Considerations

Before you use a particular tag attribute to reset an action request, make sure the attribute does not have read-only permission. If a tag attribute has read-only permission, the action request does not reset because OSx cannot write its answer to an attribute that does not change.

Status attributes are read-only if they have been configured as networked tags. Several other tag attributes are also read-only, but they consist of analog values. The values cannot be used by action requests as reset attributes.

Tag attributes that can be used to reset an action request are ones that have read-write permission. Table 12-4 at the end of this chapter lists the valid reset, trigger, and answer bits for all the tag types that you can use in configuring action requests.

Bit Logic and Action Requests

The values of the trigger bit and the reset bit determine the actions that OSx takes. If the trigger bit for a tag is set to one by the controller, OSx triggers the appropriate action request.

The bit logic used in action requests appears in Table 12-2.

Table 12-2 Bit Logic and Action Requests

Event Name	Trigger Bit	Reset Bit
Trigger Event	1 (controller)	1 (controller)
Reset Event	0 (controller)	0 (controller)
Answer Event	No Action	0 (Action Request)
Controller Response to Answer	0 (controller)	No Action

The combined logic of the controller and the action request package determines the event that takes place. Consider the following guidelines.

- The trigger event occurs when a controller program determines that some operator activity is needed and changes the state of the trigger and reset bits from zero to one. Only a specified bit in the attribute changes state and triggers the event.
- The reset event occurs when a controller determines that the trigger event is no longer valid. In this case, the controller changes the state of both the trigger bit and the reset bit from one to zero.
- The trigger and reset bits may be the same bit for the same tag attribute. If this is true, OSx automatically resets the action request when the answer event occurs. In this case a controller is not required to change the reset bit from one to zero since OSx does this automatically.
- Never use the special OSx bits (Active, Alrm_UnAck, and Man_Set) when configuring the trigger and reset bits.
- Be sure that each action request uses a trigger tag/attribute/bit and reset tag/attribute/bit that are not used by other action requests.

- The answer event occurs when the operator uses the action request package to successfully commit an answer. The action request changes the state of the reset bit from one to zero.

  If the reset bit is networked to the controller when an answer is committed, OSx changes the reset bit in the controller as it updates the database. If the reset bit is not networked to the controller, the controller is not involved in the reset process.

- The controller detects whether an answer event has occurred by monitoring any changes in the reset bit. The controller program is responsible for changing the state of the trigger bit after an action request is answered. If the trigger and reset bits are the same, the controller program does nothing more than note that the request is ready to be triggered again.

Keep the following protocols in mind while programming a controller for action requests.

- The controller controls all trigger bit changes.
- The controller sets the trigger bit and reset bit to trigger a request.
- If the controller resets the trigger bit, it also changes the state of the reset bit to zero.
- The controller monitors the reset bit and responds to state changes. If the bit changes state from one to zero, the controller is programmed to recognize that the request/answer handshake protocol is complete.

Considerations for Multiple-Station Systems

If you want to ensure that action requests occur at the same time on different stations, follow these guidelines.

Trigger Tag Attribute Use a bit in the tag status attribute to trigger the action request. Only a status attribute can be an RBE value, which is reported to all stations at the same time. If you choose not to use the status attribute as the action request trigger, use an attribute with a deadband value equal to zero or as low as possible.

Trigger Event Duration If you choose not to use the status attribute as the action request trigger, be sure that the triggering event has a duration that is longer than the Event Scan Period. This ensures that all OSx stations are notified of the event at the same time.

Embedded Values If you use an embedded tag value in the action request, and you want to ensure that all stations display the same value, use a bit in the tag's status attribute as the trigger, and use one of the attributes listed in Table 12-3 from the same tag as the trigger for the embedded value. These attributes are linked to the RBE change event and are reported to all stations at the same time as the status attribute.

Table 12-3 Attributes Linked to RBE Change Events

Tag Type	Attributes	Tag Type	Attributes
AI	PV, Target	DO	Command
AO	Out, Mode	DO10	Data_Val1, Data_Val2, Data_Val3, Data_Val4
AREA	Plc_Rcp_Req, Hold_Req, Scale_Factor	IVAR	Value
CALC	Value	LOOP	PV, SP, Out, Mode
CTR	Preset, Current	TEXT	Text_1, Text_2, Text_3
Device Tags	Override, Mode_Cmd, Setpoint	TMR	Preset, Current
DI	None	UNIT	Command, Mode_Cmd, Bch_Req, Bch_Req_Info
DI10	None		

Triggering Action Requests with System Tags

If you configure the trigger tag for an action request to be a system tag, the action request is initiated by a system tag on any station, but the action request appears only on the station where the system tag triggered. This allows you to define a station-specific action request.

For example, if you use the H_Alarm bit of the _OSX_FILSYS system tag, the action request appears only at the station on which the **hist** file system enters high alarm.

Since action request messages are logged only from the primary, the RPLOG file contains only those action requests that the primary detects. This is so that normal action requests (not triggered by system tags), do not cause other stations to report the same information to the daily log file.

NOTE: You can use any non-networked tag that is not shadowed to other stations as a station-specific trigger for action requests. OSx does not shadow system tags since they are intended to be unique to a given station. OSx does shadow process I/O tag types such as LOOP, IVAR, or AO, even if they are not networked.

⚠ WARNING

Using a system tag as a reset tag in an action request causes the system tag bit to be reset when the action request is answered.

With the bit reset to zero, the alarm is effectively masked. This makes it possible to miss the alarm condition, which could cause serious injury or death to personnel, and/or damage to equipment.

Never use a system tag as the reset bit in an action request.

Table 12-4, beginning on page 12-36, lists the valid reset, trigger, and answer bits for all the tag types that you can use in configuring action requests. An x in a column means that you can use that tag attribute for that kind of bit.

Table 12-4 Tag Attributes Relationship for Action Request Reset, Trigger, and Answer Bits

TAG	ATTRIBUTE	DATA TYPE	RESET	TRIGGER	ANSWER	BIT NAME	HEX
AI	STATUS	status16		x		HH_Alarm	0x2000
				x		H_Alarm	0x1000
				x		L_Alarm	0x0800
				x		LL_Alarm	0x0400
				x		L_Dev (yellow)	0x0200
				x		H_Dev (orange)	0x0100
				x		ROC_Alarm	0x0080
				x		Bad_Xmtr	0x0040
	TARGET	float32			x		
AO	MODE[1]	scalar	x	x	x	Auto_Man	0x8000
	STATUS	status16		x		Auto_Man	0x8000
	OUT	float32			x		
CALC	VALUE	float32			x		
CTR	PRESET	float32			x		
DI	STATUS	status16		x		Data_Val	0x8000
DO	COMMAND	scalar	x	x	x	Cmd_Val	0x8000
	STATUS	status16		x		Data_Val	0x8000
DI10	STATUS	status16		x		Data_Val1	0x8000
				x		Data_Val2	0x4000
				x		Data_Val3	0x2000
				x		Data_Val4	0x1000
				x		Data_Val5	0x0800
				x		Data_Val6	0x0400
				x		Data_Val7	0x0200
				x		Data_Val8	0x0100
				x		Data_Val9	0x0080
				x		Data_Val10	0x0040

1 APT does not use the mode attribute; TISOFT does.

Table continued on next page.

Table 12-4 Tag Attributes Relationship for Action Request Reset, Trigger, and Answer Bits (continued)

TAG	ATTRIBUTE	DATA TYPE	RESET	TRIGGER	ANSWER	BIT NAME	HEX
DO10	DATA_VAL1	scalar	x	x	x	Cmd_Val 1	0x8000
	DATA_VAL2		x	x	x	Cmd_Val 2	0x8000
	DATA_VAL3		x	x	x	Cmd_Val 3	0x8000
	DATA_VAL4		x	x	x	Cmd_Val 4	0x8000
	DATA_VAL5		x	x	x	Cmd_Val 5	0x8000
	DATA_VAL6		x	x	x	Cmd_Val 6	0x8000
	DATA_VAL7		x	x	x	Cmd_Val 7	0x8000
	DATA_VAL8		x	x	x	Cmd_Val 8	0x8000
	DATA_VAL9		x	x	x	Cmd_Val 9	0x8000
	DATA_VAL10		x	x	x	Cmd_Val 10	0x8000
	STATUS	status1		x		Data_Val1	0x8000
				x		Data_Val2	0x4000
				x		Data_Val3	0x2000
				x		Data_Val4	0x1000
				x		Data_Val5	0x0800
				x		Data_Val6	0x0400
				x		Data_Vall7	0x0200
				x		Data_Val8	0x0100
				x		Data_Val9	0x0080
				x		Data_Val10	0x0040
IVAR	VALUE	sint16			x		

Table continued on next page.

Table 12-4 Tag Attributes Relationship for Action Request Reset, Trigger, and Answer Bits (continued)

TAG	ATTRIBUTE	DATA TYPE	RESET	TRIGGER	ANSWER	BIT NAME	HEX
LOOP	STATUS	status16		x		Auto_Man	0x8000
				x		Cascade	0x4000
				x		Error_Dev	0x2000
				x		HH_Alarm	0x1000
				x		H_Alarm	0x0800
				x		L_Alarm	0x0400
				x		L_Dev (yellow)	0x0200
				x		H_Dev (orange)	0x0100
				x		ROC_Alarm	0x0080
				x		Bad_Xmtr	0x0040
	SP	float32			x		
	OUT				x		
TEXT	TEXT_1	string			x		
	TEXT_2				x		
	TEXT_3				x		
TMR	PRESET	float32			x		
VLV1	MODE_CMD	scalar	x	x	x	Mode	0x8000
	OVERRIDE		x	x	x	Ovrd_Fdbk	0x8000
	SETPOINT		x	x	x	Setpoint	0x8000
	STATUS	status16		x		Opened	0x8000
				x		Closed	0x4000
				x		NotOpened	0x2000
				x		NotClosed	0x1000
				x		Interlockd	0x0800
				x		Mode	0x0400
				x		In_Use	0x0200
				x		Setpoint	0x0100
				x		Ovrd_Fdbk	0x0080
				x		Travel	0x0040

Table continued on next page.

Table 12-4 Attributes Relationship for Action Request Reset, Trigger, and Answer Bits (continued)

TAG	ATTRIBUTE	DATA TYPE	RESET	TRIGGER	ANSWER	BIT NAME	HEX
VLV2	MODE_CMD	scalar	x	x	x	Mode	0x8000
	OVERRIDE		x	x	x	Ovrd_Open	0x8000
			x	x	x	Ovrd_Close	0x4000
	SETPOINT		x	x	x	Setpoint	0x8000
	STATUS	status16		x		Opened	0x8000
				x		Closed	0x4000
				x		NotOpened	0x2000
				x		NotClosed	0x1000
				x		Interlockd	0x0800
				x		Mode	0x0400
				x		In_Use	0x0200
				x		Setpoint	0x0100
				x		Ovrd_Open	0x0080
				x		Ovrd_Close	0x0040
				x		Travel	0x0020
				x		Failed	0x0010
MTR1	MODE_CMD	scalar	x	x	x	Mode	0x8000
	OVERRIDE		x	x	x	Ovrd_Fdbk	0x8000
	SETPOINT		x	x	x	Setpoint	0x8000
	STATUS	status16		x		Running	0x8000
				x		Stopped	0x4000
				x		Not Running	0x2000
				x		NotStopped	0x1000
				x		Interlockd	0x0800
				x		Mode	0x0400
				x		In_Use	0x0200
				x		Setpoint	0x0100
				x		Ovrd_Fdbk	0x0080
				x		Travel	0x0040

Table continued on next page.

Table 12-4 Attributes Relationship for Action Request Reset, Trigger, and Answer Bits (continued)

TAG	ATTRIBUTE	DATA TYPE	RESET	TRIGGER	ANSWER	BIT NAME	HEX
RMTR	MODE_CMD	scalar	x	x	x	Mode	0x8000
	OVERRIDE	scalar	x	x	x	Ovrd_Fwd	0x8000
	OVERRIDE	scalar	x	x	x	Ovrd_Rev	0x4000
	SETPOINT		x	x	x	Setpoint	0x8000
			x	x	x	Direction	0x4000
	STATUS	status16		x		Running	0x8000
				x		Stopped	0x4000
				x		Not Running	0x2000
				x		NotStopped	0x1000
				x		Interlockd	0x0800
				x		Mode	0x0400
				x		In_Use	0x0200
				x		Setpoint	0x0100
				x		Direction	0x0080
				x		Ovrd_Fwd	0x0040
				x		Ovrd_Rev	0x0020
				x		Travel	0x0010
				x		Failed	0x0008

Table continued on next page.

Table 12-4 Attributes Relationship for Action Request Reset, Trigger, and Answer Bits (continued)

TAG	ATTRIBUTE	DATA TYPE	RESET	TRIGGER	ANSWER	BIT NAME	HEX
MTR2	MODE_CMD	scalar	x	x	x	Mode	0x8000
	OVERRIDE		x	x	x	Ovrd_High	0x8000
			x	x	x	Ovrd_Low	0x4000
	SETPOINT		x	x	x	Setpoint	0x8000
			x	x	x	Direction	0x4000
	STATUS	status16		x		Running	0x8000
				x		Stopped	0x4000
				x		Not Running	0x2000
				x		NotStopped	0x1000
				x		Interlockd	0x0800
				x		Mode	0x0400
				x		In_Use	0x0200
				x		Setpoint	0x0100
				x		Speed	0x0080
				x		Ovrd_High	0x0040
				x		Ovrd_Low	0x0020
				x		Travel	0x0010
				x		Failed	0x0008

Chapter 13

Responding to Action Requests

13.1 Monitoring Action Requests

General Guidelines

After the action requests are defined, and the system is placed in the Operate state, the action request package begins monitoring each of the trigger tags. Any time a trigger bit for a tag attribute changes state from 0 to 1, the OSx station screen displays the action request icon in the lower righthand corner. (Figure 13-1).

The icon has the following characteristics:

- It is yellow and contains the text Normal when the highest priority of any unanswered action requests is normal. The icon flashes on all OSx station monitors whose currently logged-in users have a process group in common with the triggered request until someone acknowledges it.
- It is red and contains the text Urgent when the highest priority of any unanswered action requests is urgent. It flashes on all OSx station monitors whose currently logged-in users have a process group in common with the triggered request until someone acknowledges it.
- The higher priority Urgent icon will always replace the Normal icon if an urgent request has been triggered.
- The icon stops flashing on the OSx station monitor (all monitors in a multiple-station system) when someone selects it to display the Action Request Summary.
- The icon disappears if the operating state changes from the Operate state to the Offline state or if all action requests are answered.

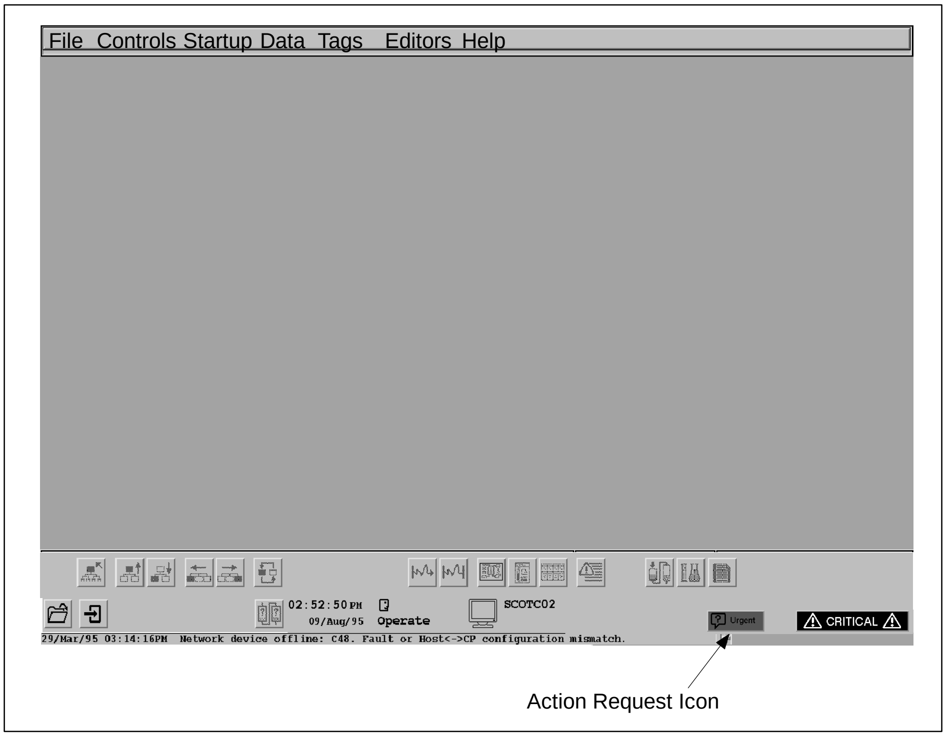


Figure 13-1 Location of Action Request Icon

When an operator selects the action request icon, the Action Request Summary display appears. This display allows the operator to respond to the request triggered by the process. If a new action request is triggered when this screen is displayed, the operator can select the icon to display the new request's summary in the lower left part of the screen (Figure 13-2).

From this display an operator can answer an action request and type comments associated with it. The icon disappears when all requests are answered.

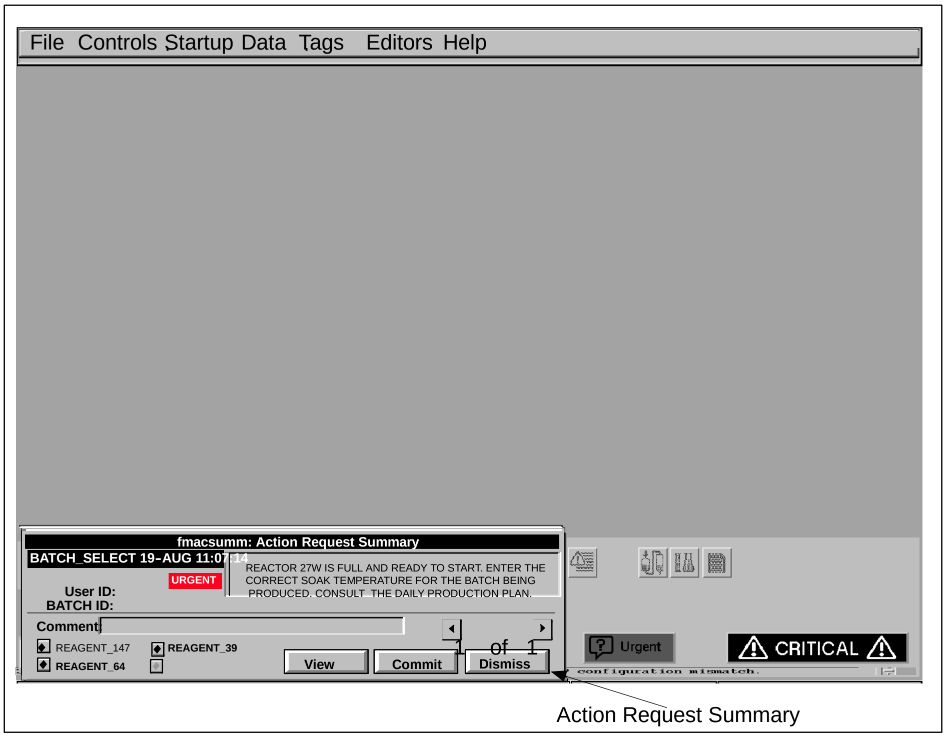


Figure 13-2 Location of the Action Request Summary Display

If an operator chooses to answer an action request and commit the answer to the database, OSx updates the database, changes the state of the reset bit from 1 to 0, and records the action on the Operator Change Log. At this point, the action request is ready to be triggered again when the controller program requests interaction. Refer to Section 12.7 for more information about reset logic.

Occasionally, a controller program resets a triggered action request before the operator has answered it. In this situation, the action request software removes the request and its corresponding icon from the OSx station and makes a record of the activity on the Operator Change Log.

During normal operations, one or more action requests can be triggered at the same time. In most situations, try to configure the requests so they occur at different times.

NOTE: In order to answer an action request, the operator must have both the action request privilege and the privilege to modify the tag attributes involved. Refer to Chapter 6 on viewing tag information to determine which security privileges allow an operator to modify particular tag attributes.

An operator with the privilege to respond to action requests, but without the privilege to modify the tag attribute involved, can respond to view-only action requests.

Action Request Mini-Windows

You can attach an action request to a graphic using the mini-windows feature of OSx. When you enable mini-windows from the Event Preferences screen, any action request attached to a graphic is displayed in a list at the lower right of the graphic if the request has been triggered and not reset or answered. Refer to Chapter 15 to configure mini-windows for an action request.

13.2 Answering an Action Request

Selecting the Action Request Icon

To select a flashing action request icon from the OSx station screen, use one of the following methods:

- Click the icon.
- Press the **Action Request** key on the operator keyboard.

Each choice displays the Action Request Summary in the lower left part of the OSx screen. Figure 13-3 shows the general format of the Action Request Summary. Descriptions of the fields for this display follow.

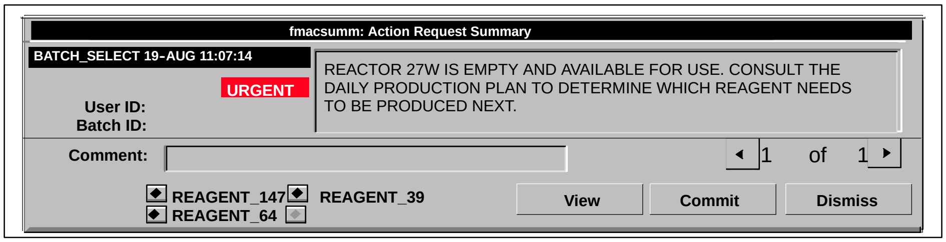


Figure 13-3 Action Request Summary Dialog Box

Request Name Name of the action request that is triggered.

Date Date the action request is triggered by an event. Format: DD-MMM, where DD=day and MMM=three letter name of the month (for example, APR for April).

Time Time the action request is triggered by an event. Format: HH:MM:SS, where HH=hours, MM=minutes, and SS=seconds.

Priority Priority status of the action request. It may have the value Urgent (red) or Normal (yellow). The Action Request Summary displays Urgent requests before Normal requests. In addition, within each urgency level, the Action Request Summary displays requests by date and time; the most recent requests are listed first. If mini-windows are enabled, the status for the View action request is displayed in blue, regardless of the urgency.

User ID Identity of the operator or node answering the action request. This field is displayed only after a request is answered.

Batch ID Batch ID of the trigger tag. If the trigger tag is not part of a batch this field is blank.

Text Text associated with the action request. It may include embedded values.

Comment The operator can enter a comment in this field. Comments are useful for documenting action request activities because they are placed on the Operator Change Log with the request answer.

 Select this button to access the next action request in the system.

 Select this button to access the previous action request.

of # Current page number and the total number of pages in the action request summary; for example, 2 of 8.

View Select this button to display a graphic or tag detail that you have associated with the action request. If no associated display is configured, this button is grayed out.

Commit Select this button to commit an answer to an action request. OSx places the answer in the database and uses that value to update the process.

Dismiss Select this button to cancel an answer sequence any time prior to committing the answer. This function does not cancel the request.

The following two fields appear only when the action request is the Enter Value type and the operator selects the **Enter Value** button: Last Value and New Value. Figure 13-4 shows an Action Request Summary for an Enter Value request.

Last Value Last value of the answer attribute for the Enter Value answer type.

New Value New value entered for the answer attribute of the Enter Value answer type.

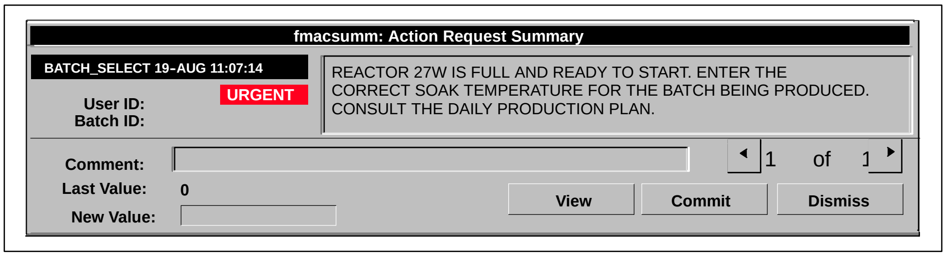


Figure 13-4 Action Request Summary for Enter Value

Choices When the action request is the Multi-Choice type, four radio buttons with the available choices are displayed at the lower left of the Action Request Summary. Buttons that have no choices configured are grayed out. Figure 13-5 shows an Action Request Summary for a Multi-Choice request.

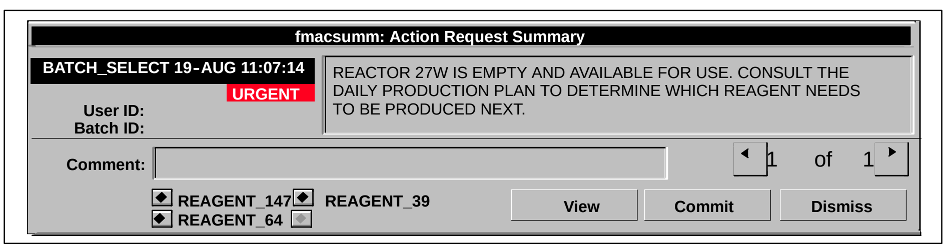


Figure 13-5 Action Request Summary for Multi-Choice

General Guidelines for Answering an Action Request

Certain general guidelines apply to answering all action requests:

- The OSx station must be in the Operate state.
- To select an action request, click the action request icon in the lower right corner of the screen.
- The station displays only one action request at a time. To view other requests, click  or  to select the next or previous action request.
- Once you initiate an answer, the  and  buttons are desensitized.
- To cancel an answer, select **Dismiss**. To see the request again, you must select the action request icon again.

Action Requests during Synchronization

When the database is locked during synchronization, you need to be aware of the following points concerning action requests.

- During synchronization, you cannot access action requests from any station. If you attempt to access an action request while the database is locked, the screen displays a message indicating that the summary cannot be accessed while the database is locked.

  If you have displayed the summary before OSx locks the database, and then you press the **Commit** button, the system does not commit the answer until after the database is unlocked. The system does not process any further action request selections until after it unlocks the database.
- The action request button appears on all stations except the primary and the station that is synchronizing.
- The action request button appears on the primary for new alarms and action requests after OSx unlocks the database.

Responding to Different Action Request Types

On the Action Request Summary screen, an operator can encounter up to four different types of action requests. Each type requires a different method for answering the request.

Before OSx accepts a response to a request, the operator must have sufficient security privileges to modify the following tags:

- Answer tags for the Multi-Choice and Enter Value requests.
- Reset tags for all types of requests. See Section 12.7 for information about trigger and reset logic.

The operator also has the option to enter comments in the Comment field when answering each request. This practice allows the operator to document all action request transactions.

The Operator Change Log contains a complete record of each transaction. Refer to Chapter 14 on OSx Logs for more information.

The paragraphs on the following pages describe the five action request types and the procedures for answering each.

Multi-Choice

The operator chooses one response from up to four possible options. OSx writes the answer value to the database attribute defined during the configuration of the action request. The operator should follow these steps:

1. Read the text of the action request thoroughly.
2. Select the appropriate response from the options listed at the lower left of the Action Request Summary. If an error is made, the operator can select a different choice before committing it.
3. Enter a comment in the Comment field. The operator has the option of entering a comment at any time before committing the answer.
4. If you want to commit the answer to the system, select **Commit**. The value corresponding to the selected choice is written to the Answer attribute. Then select **Dismiss** to close the form.

   If you want to cancel the answer procedure, select **Dismiss** without committing the answer.

NOTE: If an operator tries to commit a Multi-Choice value that is out of range (or fails for any other reason), the **Commit failed** error message appears, and all the other choices are grayed out. The operator must leave that action request (for example, by selecting) and return to it (in this case, by selecting) before making another selection.

Acknowledge

The operator acknowledges that a triggered event has occurred. The operator should follow these steps:

1. Read the text of the action request thoroughly.
2. Enter a comment in the Comment field. This step is optional.
3. To acknowledge the request, select **Commit**. Then select **Dismiss** to close the form.

   If you want to cancel the answer procedure, select **Dismiss** without committing the answer.

Answering an Action Request (continued)

Enter Value

The Enter Value Request is a multi-step operation. The operator enters a response in the New Value field. If the value entered is out of range, an error dialog box appears showing the limits. The Last Value appears above the New Value field. The operator should follow these steps:

1. Read the text of the action request thoroughly.
2. Enter the new value in the New Value field and press **Return**. The screen validates the entry to ensure that it falls within defined limits. If an error is made, the operator can reenter the value before committing it.
3. Enter a comment in the Comment field. This step is optional.
4. If you want to commit the answer to the system, select **Commit**. The value that you entered is written to the Answer attribute. Then select **Dismiss** to close the form.

   If you want to cancel the answer procedure, select **Dismiss** without committing the answer.

View

The operator can simply view the action request text. If the operator has the appropriate security privilege for the reset tag, the system answers a View request automatically when the Action Request Summary screen is displayed. The operator does not have to commit the answer to the system.

Event Log

No acknowledgement is required for this action request since it does not turn on the action request icon when triggered. Instead, it logs a message to the Operator Change Log and then resets itself.

13.3 Example of Responding to Action Requests

In Chapter 12, you created five action requests for a sample reactor process. This section describes how to respond to each type of request requiring an answer for the same example.

The example described below leads you step by step through the process. It shows where each action request is triggered in the process and lists procedures for answering each request. Figure 13-6 represents the process graphic the operator sees when the action requests are triggered.

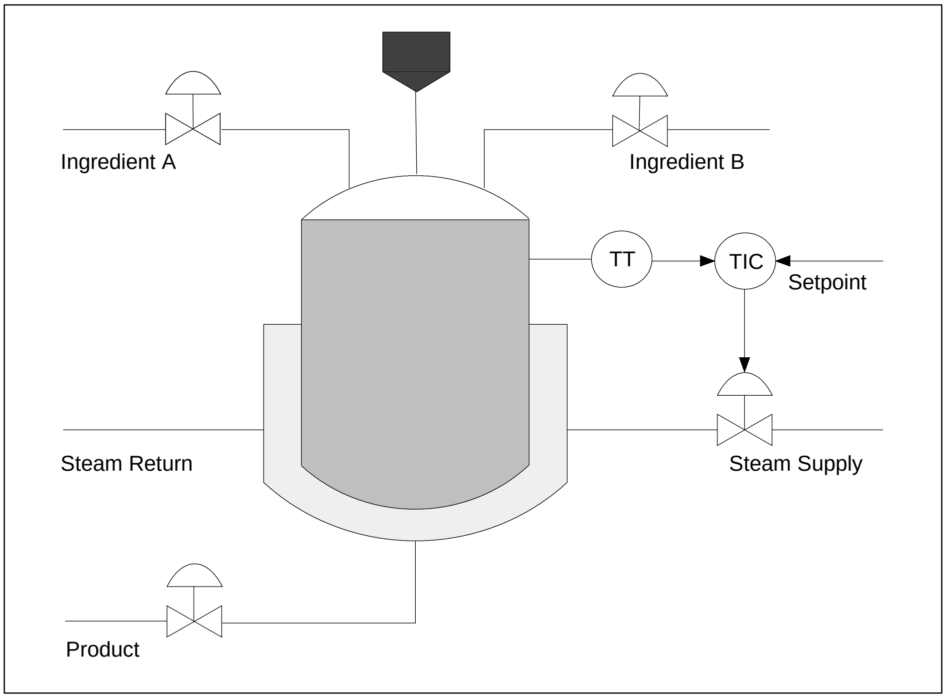


Figure 13-6 Sample Reactor Process

NOTE: Before the operator can respond to any of the requests, the system must be in the Operate state.

Example of Responding to Action Requests (continued)

Starting the Process

When the system enters the Operate state, the controller in this example checks whether or not the reactor tank is empty.

If the tank is empty, the controller closes the output valve and triggers the BATCH_SELECT action request with the LL_Alarm bit of trigger tag REACTOR27W. The Operator Change Log records the trigger event, including the date, time, text (with embedded values), answer options, and the request's urgency.

If the tank is not empty, the output valve remains open until all the fluid drains from the tank.

Answering the BATCH_SELECT Action Request

The first action request is triggered by an empty reactor tank. To answer the request, the operator should follow these steps:

1. Select the action request icon. The screen displays the Action Request Summary. The summary shows the BATCH_SELECT action request (Figure 13-7). It is a Multi-Choice request.

2. If the Daily Production Plan calls for Reagent 64 to be produced, place the cursor on the Reagent_64 option and select it.

3. Enter a comment stating the reason why Reagent 64 was chosen.

4. Select the **Commit** option. OSx commits the choice by writing the value 0x2000 (for Reagent 64) to the attribute of tag 27WCONTROL. OSx changes the reset bit for the BATCH_SELECT request from 1 to 0.

   The Operator Change Log records the comment and the answer. The two chemicals that make up the Reagent 64 mixture begin to flow into the reactor tank.

The request is now complete.

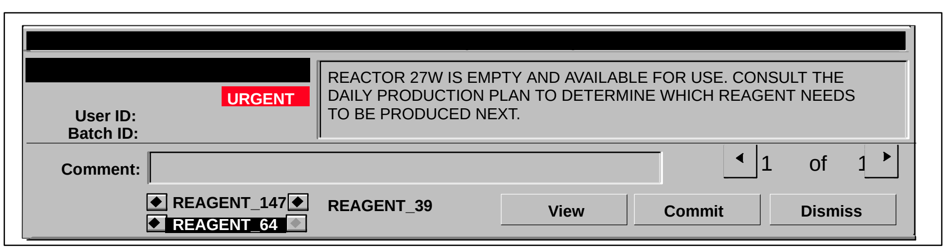


Figure 13-7 BATCH_SELECT Action Request

Adding a Catalyst to the Reactor Tank

The two chemicals continue to flow into the reactor tank until the H_Alarm bit of the REACTOR27W trigger tag turns on. At this point, the process stops and triggers the ADD_CATALYST action request. The operator must add a catalyst to the tank before the two ingredients can react properly.

Answering the ADD_CATALYST Action Request

The second action request is triggered by the high alarm of the reactor tank. The high alarm triggers the request so the operator can stop the process, add a catalyst, and allow the two ingredients to continue entering the tank. To answer the ADD_CATALYST request, the operator should follow these steps:

1. Select the action request icon. The screen displays the Action Request Summary. The summary shows the ADD_CATALYST request (Figure 13-8). It is an Acknowledge request.

2. Follow the directions given by the text of the request. Add the catalyst to the reactor tank.

3. In the Comment field, enter the milliliters of catalyst added to the reagent mixture.

4. Select the **Commit** option to acknowledge the request. OSx commits the answer and changes the reset bit from 1 to 0.

   The comment and answer are written to the Operator Change Log after the action request is committed. The two reagent chemicals continue to fill the tank.

The request is now complete.

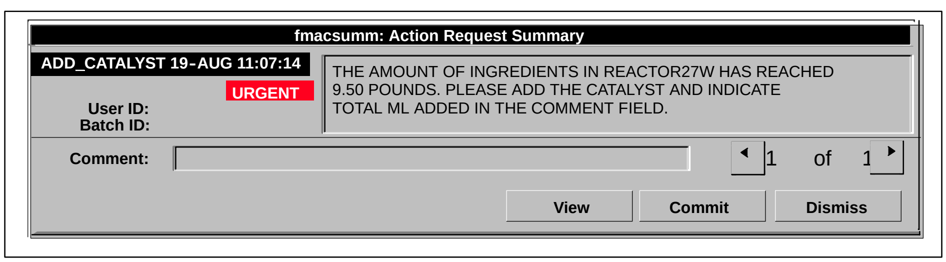


Figure 13-8 ADD_CATALYST Action Request

Filling the Reactor Tank

The two chemicals flow into the reactor tank until the HH_Alarm bit of the REACTOR27W trigger tag turns on. At this point, the process stops and triggers the SET_TEMP action request.

Answering the SET_TEMP Action Request

The third action request is triggered when the reagent mixture reaches the high-high alarm level. To answer the SET_TEMP action request, the operator follows these steps:

1. Select the action request icon. The screen displays the Action Request Summary dialog box. The summary shows the SET_TEMP action request (Figure 13-9). It is an Enter Value request.

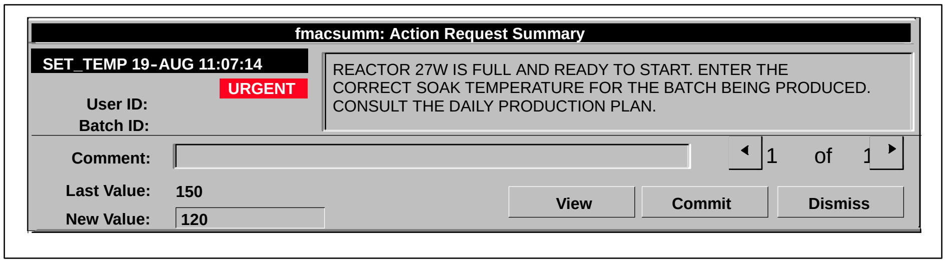


Figure 13-9 SET_TEMP Action Request

2. In the New Value field at the bottom left of the Action Request Summary, enter a temperature value between 50° and 210° F.

   In Figure 13-9, the new value is 120° F. The temperature varies according to the reagent being created. If an error is made, the operator can reenter the value before committing it.

3. In the Comment field, enter an appropriate comment about the choice of temperature.

4. Select the **Commit** option. OSx commits the value entered. The Operator Change Log records this comment. Steam begins to flow into the steam jacket and heats the mixture.

Heating the Mixture

The process continues to heat the mixture until the selected temperature is reached. When the temperature is reached, the H_Alarm bit of the REACT tag turns on. This, in turn, triggers the BATCH_DONE action request.

Answering the BATCH_DONE Action Request

The fourth action request is triggered when the mixture in the reactor tank reaches the selected temperature. To answer the request, the operator selects the action request icon. The screen displays the Action Request Summary dialog box (Figure 13-10). The summary shows the BATCH_DONE action request. It is a View request.

The operator does not have to commit this action request. All that is required to answer the request is to view it. The Operator Change Log records this request.

The request is now complete. The operator clicks the **Dismiss** button to dismiss the Action Request Summary.

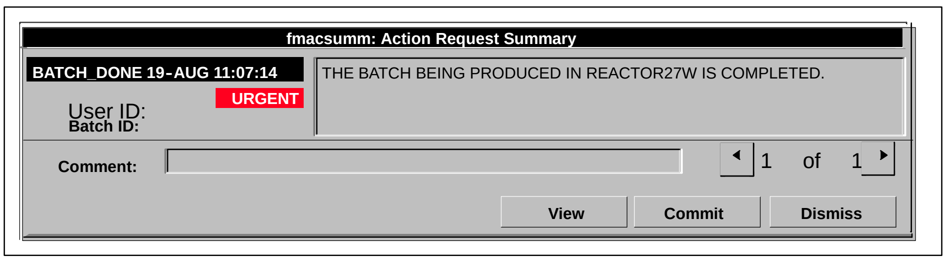


Figure 13-10 BATCH_DONE Action Request

The BATCH_HOLD Action Request

The fifth action request is triggered only if the batch is temporarily suspended for some reason. A log message is sent to the daily log file indicating that the batch was suspended. No operator response is required for this action request. It is an Event Log request.

Finishing the Process

After the defined temperature has been reached, the process is done. The operator can empty the tank either manually by opening the outlet valve, or automatically by allowing the controller to do it. After the tank has been drained, it is ready for a new reagent batch.

Chapter 14
Using OSx Logs

14.1 Using the Alarm Log

Identifying Alarm Log Messages

The Alarm Log lists all alarms, regardless of priority. The Alarm Log is part of a standard OSx report called the System Log Report. When any alarm occurs, OSx records alarm data in the Alarm Log and sends the Alarm Log to a printer if you assigned one during printer configuration.

OSx creates a new log file every day and stores each file for the number of days that is specified in the File Duration field of the Printer Configuration Destination display.

OSx stores the Alarm Log, Operator Change Log, Batch Log, and System Messages in a file that is stored in the **/usr/tistar/hist/rpt/output** directory and is named **RPLOG.<date>-<sequence number>**, for example, **RPLOG.06APR-1** for the first file of April 6th. This file is also listed in the Reports Directory.

Accessing the Reports Directory

To access the directory of reports, follow these steps.

1. Select the **Directory** pushbutton from the navigation area. A directory of display types appears.
2. Select the **Report** option. The screen displays a list of reports (Figure 14-1).
3. Select the report. The screen displays the report you selected.

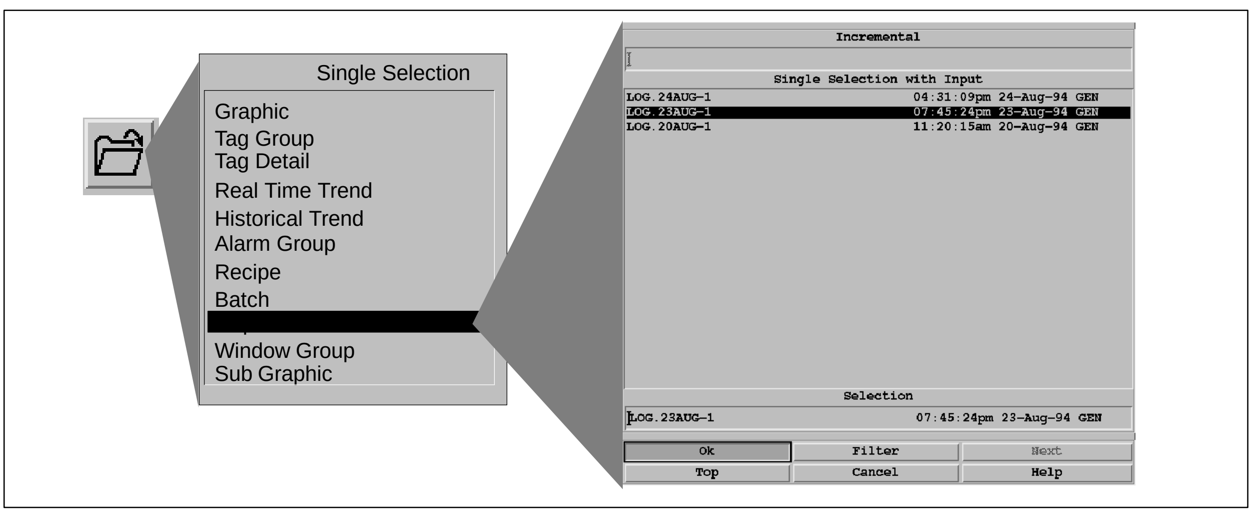


Figure 14-1 Reports Directory Display

Table 14-1 lists the eight alarm log entry types and the four alarm priorities. Figure 14-2 shows an example of the Alarm Log. The type of information that can appear in each column of the report varies with the log entry type. The different information types appear at the top of Figure 14-2.

Table 14-1 Alarm Log Entry Types and Alarm Priorities

Log Entry Type		Alarm Priorities
IN = In Alarm	ST = Alarm Tag Enable/Disable	CRIT = Critical
OUT = Out of Alarm	SEQ = Alarm State Sequencing	INFO = Information
ACK = Alarm Acknowledged	OOS = Out of Service	WARN = Warning
PRI = User Priority Change	SUP = Alarms Suppressed	MAIN = Maintenance

Log Entry Type	Time	Node Name	User ID or Alarm Group	Tag Name	Alarm Type	"Blank" or Old Priority or Old En/Disable State or Old Value or Current Value	"Blank" or New Priority or New En/Disable State or New Value or Violated Limit	Tag Descriptor	Alarm Priority
IN	00:41:33	EVCTRL	EVAP	LIC-601	ORANGE DEV	12.000	3.000	Evaporator Level Control	WARN
IN	00:41:33	EVCTRL	EVAP	LIC-601	YELLOW DEV	12.000	2.000	Evaporator Level Control	INFO
IN	00:41:33	EVCTRL	EVAP	LIC-601	LL ALARM	12.000	13.000	Evaporator Level Control	CRIT
IN	00:41:33	EVCTRL	EVAP	LIC-601	L ALARM	12.000	14.000	Evaporator Level Control	WARN
ACK	00:44:18	EVCTRL	ROOT	LIC-601	LL ALARM			Evaporator Level Control	CRIT
ACK	00:44:18	EVCTRL	ROOT	LIC-601	ORANGE DEV			Evaporator Level Control	WARN
ACK	00:44:18	EVCTRL	ROOT	LIC-601	L ALARM			Evaporator Level Control	WARN
OUT	00:45:57	EVCTRL	EVAP	LIC-601	LL ALARM	13.225	13.000	Evaporator Level Control	CRIT
OUT	00:46:00	EVCTRL	EVAP	LIC-601	L ALARM	14.494	14.000	Evaporator Level Control	WARN
OUT	00:46:06	EVCTRL	EVAP	LIC-601	ORANGE DEV	17.665	3.000	Evaporator Level Control	WARN
OUT	00:46:07	EVCTRL	EVAP	LIC-601	YELLOW DEV	18.210	2.000	Evaporator Level Control	INFO

Figure 14-2 Sample Alarm Log Information

14.2 Using the Operator Change Log

Identifying Operator Change Log Messages

OSx produces an Operator Change Log that records information about all your daily activities, such as Action Request Summary activities. The Operator Change Log is part of a standard OSx report called the System Log Report. Each time a request is triggered by the process, answered, or reset, the Operator Change Log updates automatically (Figure 14-3).

The time stamp for active action requests that have not been answered and committed is affected by the following events.

- A new station joins the system. The time stamp for the unanswered action requests is the time when the station joins, not the time when the trigger event occurs. Only action requests appearing on the station joining the system are affected.
- A station resynchronizes. The time stamp for the unanswered action requests is the time when the station completes synchronization, not the time when the trigger event occurs. Only action requests appearing on the station that resynchronizes are affected.
- OSx transitions to the Offline state and then back to Operate. The time stamp for the unanswered action requests is the time when OSx completes the transition to the Operate state, not the time when the trigger event occurs. All action requests are affected.

NOTE: If an action request is answered on a non-primary, then the Operator Change Log contains two answer entries for the action request. One is the expected answer log message. The other contains the primary's station name and indicates that the request was automatically reset. You can ignore the second answer log message.

OSx records all tag value changes that are initiated by @aGlance clients with the name of the client in the place of the User ID in the Operator Change Log.

OSx stores the Operator Change Log, Alarm Log, Batch Log, and System Messages in a file that is stored in the **/usr/tistar/hist/rpt/output** directory and is named **RPLOG.<date>-<sequence number>**, for example, **RPLOG.06APR-1** for the first file of April 6th. This file is also listed in the Reports Directory. To access the directory of reports, follow the steps listed on page 14-2.

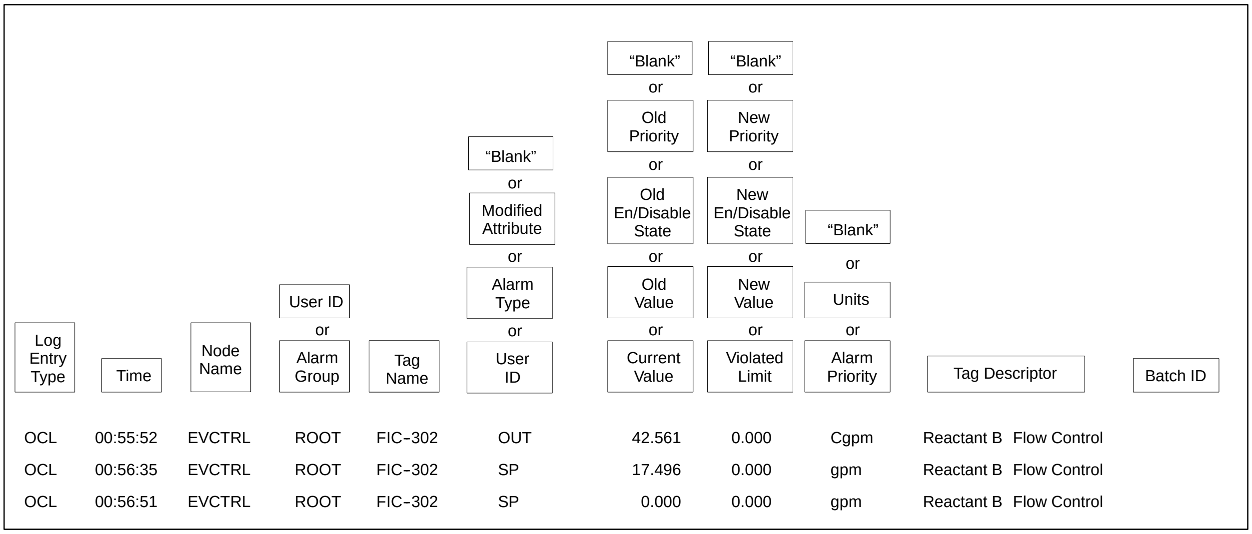


Figure 14-3 Operator Change Log Example

14.3 Using the Batch Log

Identifying Batch Log Messages

The batch log consists of all messages that OSx generates during batch processing. Like the other logs that OSx generates, the batch log is not a separate group of messages. Instead, the batch log characterizes those messages within the system log that are related either to a specific batch ID or to all batch IDs. If a system-generated message does not relate to a batch ID, then the message is not part of the batch log. See Figure 14-4.

Identifying Batches

The batch ID field in the daily system log file contains the internal ID of the batch, not the user-defined custom batch ID. However, reports reading the log file translate this internal ID to the customer-defined batch ID.

Accessing the Batch Log Report

OSx stores the Batch Log, Operator Change Log, Alarm Log, and System Messages in a file that is stored in the **/usr/tistar/hist/rpt/output** directory and is named **RPLOG.<date>-<sequence number>**, for example, **RPLOG.06APR-1** for the first file of April 6th. This file is also listed in the Reports Directory. To access the directory of reports, follow the steps listed on page 14-2.

Log Entry Type	Time	Node Name	User ID or Alarm Group	Object Name or Tag Name	"Blank" or Modified Attribute or Unit Qualifier	Tag Descriptor or Event Type	"Blank" or Function Name	"Blank" or Units	Batch ID	Old Value	New Value
OCL	23:06:58	EVCTRL	EVAP	_BCH_3	BATCH_ID	LINE 1 BATCH			BID990526-000	BID990526-000	BATCH_1
OCL	23:06:58	EVCTRL	EVAP	_BCH_3	PRODUCT_NAME	LINE 1 BATCH			BID990526-000	NONE	NONE
OCL	23:06:58	EVCTRL	EVAP	_BCH_3	DESCRIPTOR	LINE 1 BATCH			BID990526-000	Bch dscr_BCH_3	LINE 1 BATCH
TRK	23:06:59	EVCTRL	EVAP	BATCH		STATE_CHANGE	COMMIT		BID990526-000		
TRK	23:06:59	EVCTRL	EVAP	BATCH		STATE_CHANGE	AUTO		BID990526-000		
TRK	23:07:59	EVCTRL	EVAP	UNIT		ASSIGNED		LINE_1	BID990526-000		
OCL	23:08:00	EVCTRL	EVAP	_BCH_4	BATCH_ID	Bch dscr _BCH_4			BID990526-000	BID990526-000	SUB_BATCH_1
OCL	23:08:00	EVCTRL	EVAP	_BCH_4	PRODUCT_NAME	LINE 1 BATCH			BID990526-001	NONE	NONE
OCL	23:08:00	EVCTRL	EVAP	BCH_4	DESCRIPTOR	LINE 1 BATCH			BID990526-001	Bch dscr_BCH_4	LINE 1 BATCH
OCL	23:08:00	EVCTRL	EVAP	_BCH_3	COMMAND	LINE 1 BATCH			BID990526-000	RESET	SPLIT

Figure 14-4 Example of Batch ID Fields in the Batch Log

14.4 System Messages

Identifying System Messages

System Messages are messages associated with general system operations, such as Start-of-Day and Autolog information. System Messages are also a part of a standard OSx report called the System Log Report. Figure 14-5 shows an example of a system message report.

Accessing System Messages

OSx stores System Messages, Batch Logs, Operator Change Logs, and Alarm Logs in a file that is stored in the **/usr/tistar/hist/rpt/output** directory and is named **RPLOG.<date>-<sequence number>**, for example, **RPLOG.06APR-1** for the first file of April 6th. This file is also listed in the Reports Directory. To access the directory of reports, follow the steps listed on page 14-2.

Avoiding Unnecessary Messages

If you have a controller that is not online, you can set it to **Scan_Off** to avoid unnecessary error messages in **RPLOG**.

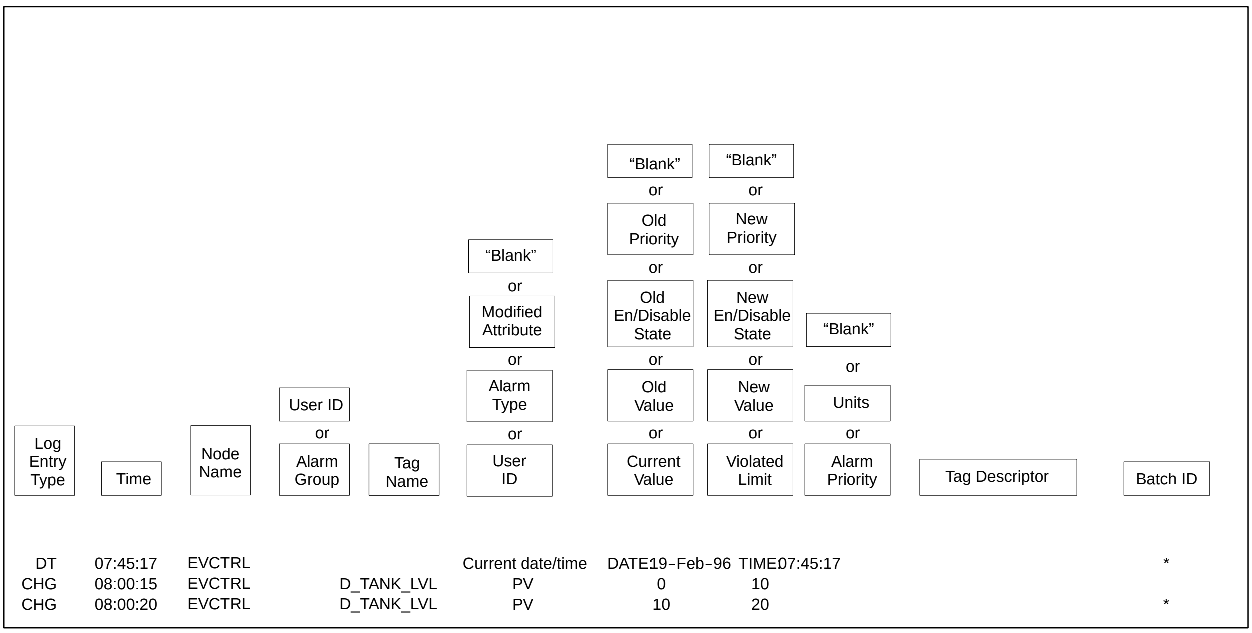


Figure 14-5 Example of a System Message Report

14.5 Using the Log File Filter

Filtering a Log File

The log file filter allows you to use several different criteria to select items from the daily log file and sort them into lists that can be viewed and saved.

For example, if you wish to check all entries concerning NODE6 between 8 AM and 2 PM, you can set up the log file filter so that all data concerning that node during the specified time period can be viewed, saved, and printed in a separate list.

To use the log file filter, follow these steps.

1. Select the **Directory** icon from the navigation area. A directory of display types appears.

2. Select **Report**. The screen displays a list of reports (Figure 14-6).

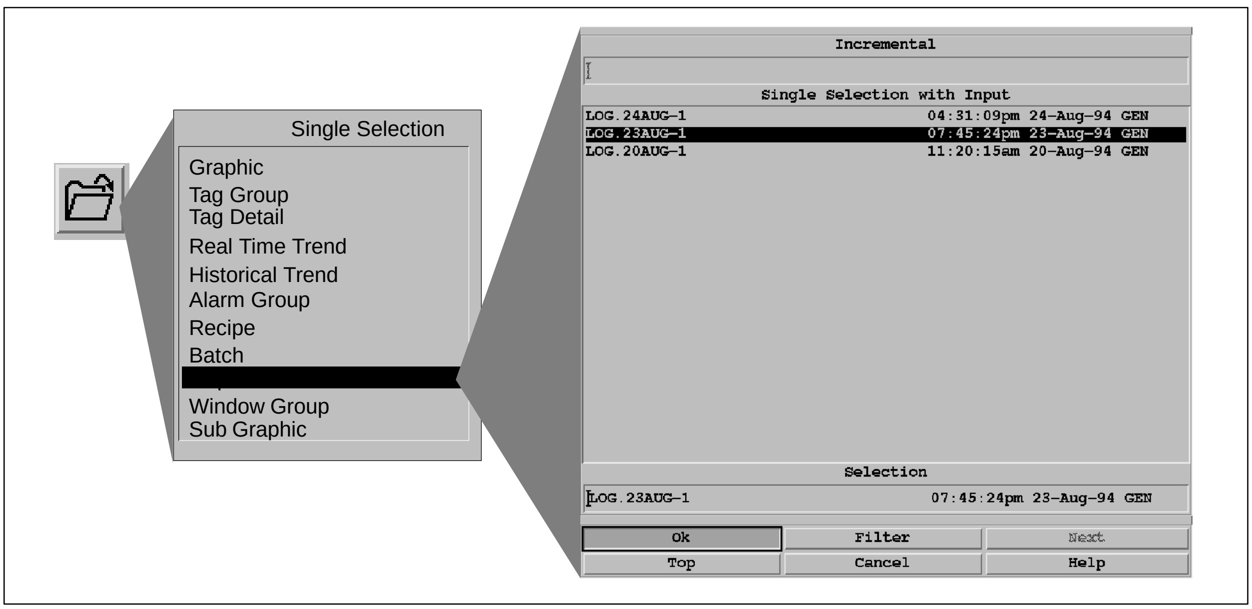


Figure 14-6 Reports Directory Display

3. Select the log file that you want to filter. The screen displays the report in the Log File Browser (Figure 14-7).

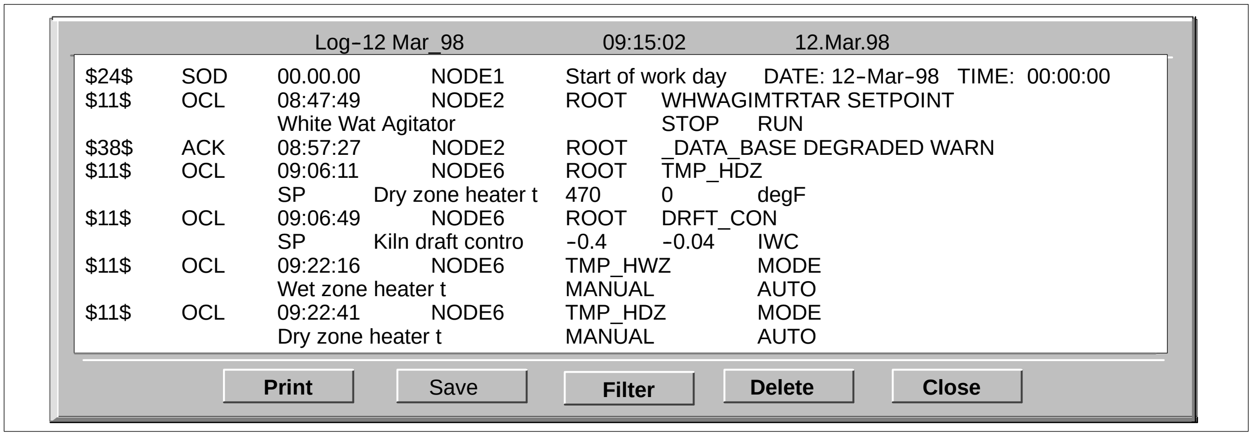


Figure 14-7 Log File Browser

4. To filter these log entries, select the **Filter** pushbutton in the Log File Browser (Figure 14-7). The Filter dialog box appears (Figure 14-8).

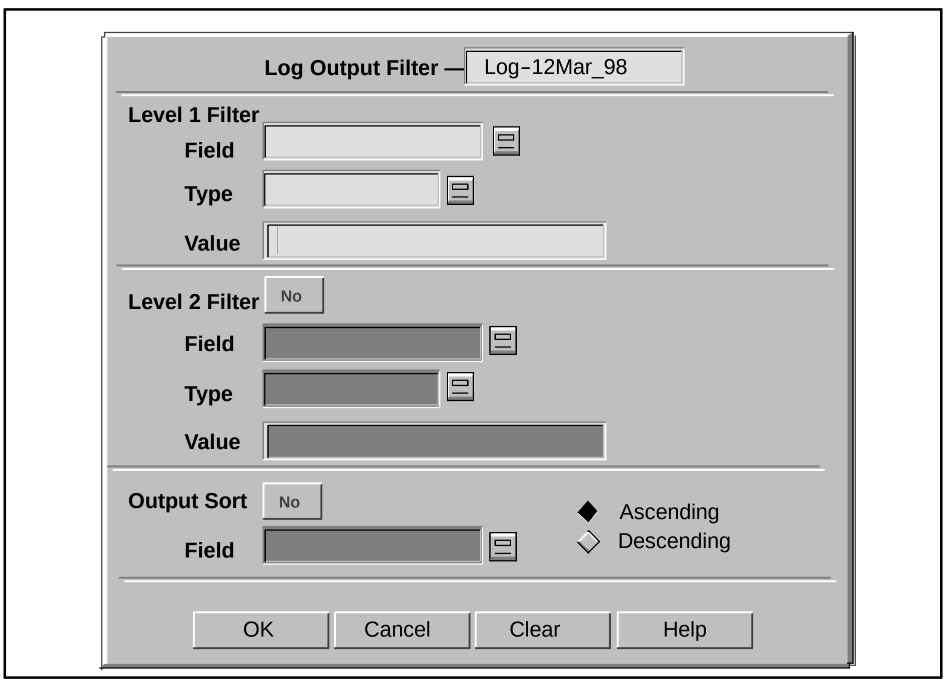


Figure 14-8 Log File Filter

5. In the Field box under Level 1 Filter, select the field that you want to use to filter the log file. For example, if you want to filter the log file by OSx station, use the long-list display tool to the right of the box to select **node**. If you type your entry instead of selecting it, be sure to use all lower-case letters.

   Table 14-2 lists valid search fields. For a particular type of log message, some fields are applicable and others are not. For example, a start-of-day log message contains a time field, but no tag_name or other tag-related fields.

Table 14-2 Log File Fields

Valid Search Fields				
actual_value	component	keyword	old_dest	state
alarm_name	curr_date	keyword	old_priority	state_num
answer_1	curr_time	limit_value	old_role	status_msg
answer_2	curr_value	line_number	old_value	tag_name
answer_3	description	log_message	priority_name	third_line
answer_4	err_num	message	proc_name	time
answer_attr	err_text	module_name	recipe_name	uname1
answer_string	error_flag	msec	request_name	uname2
answer_tag	event_type	msg_type	reset_attr	uname3
area_name	file_name	new_dest	reset_tag	unit_name
attr_name	first_line	new_priority	rpt_name	unit_qualifier
batch	fnc_name	new_role	rpt_output	urgency
batch_id	group_name	new_value	second_line	user_id
command_name	igt_mode	node	secondary_name	
comment	igt_number	object_name	start_finish	

6. In the Type box under Level 1 Filter, use the long-list display tool to the right of the box to select **MATCH**. If you type this entry instead of selecting it, be sure to use all upper-case letters. **MATCH** means that your filter will find entries that match the value that you enter in the Value box.

7. In the Value box under Level 1 Filter, type the name of the node that you want the filter to select; for example, `NODE6`. Remember that the filter is case-sensitive.

8. If you want to limit your search further, for example, if you want to see all log entries relating to NODE6 between the hours of 8:00 AM and 2:00 PM, use the Level 2 Filter. Toggle the box to the right of Level 2 Filter to read **Yes**.

9. In the Field box under Level 2 Filter, select **time**.

10. In the Type box under Level 2 Filter, select **RANGE**. This type means that your filter will find entries in the range between (and including) the values that you enter in the Value boxes.

    Table 14-3 shows valid choices for the Type field and how they work. All comparisons are text comparisons; for example, AI_10 is less than AI_2.

Table 14-3 Type Options

Type	Operation
MATCH	The filter searches for entries with an exact match to the value entered in the left Value box. *
<	The filter searches for entries less than the value entered in the left Value box.
<=	The filter searches for entries less than or equal to the value entered in the left Value box.
>	The filter searches for entries greater than the value entered in the left Value box.
>=	The filter searches for entries greater than or equal to the value entered in the left Value box.
NOT =	The filter searches for entries that do not match the value in the left Value box. *
RANGE	The filter searches for entries in the range between (and including) the lower value in the left Value box and the upper value in the right Value box.
* With the MATCH and NOT= options, you can use a wildcard character (*) to represent zero or more characters in the Value box. For example, DO* matches DO, DO_8, and DO10.	

11. In the first Value box under Level 2 Filter, type `8:00:00`. In the second Value box under Level 2 Filter, type `14:00:00`.

12. If you want to view the log entries in a particular order, toggle the box to the right of the Output Sort field to read **Yes**.

13. In the Field box, select the field you want to use to sort the log file. For example, if you want to view the file chronologically, select **time**.

14. To view the file starting with 8:00 AM and working forward to 2:00 PM, click the diamond to the left of **Ascending**.

    To view the file starting with 2:00 PM and working backward to 8:00 AM, click the diamond to the left of **Descending**. Figure 14-9 shows the completed Log File Filter.

15. Click **OK**. The filter creates a list of all log entries that concern NODE6 between the hours of 8:00 AM and 2:00 PM on the day of the log file that you selected. The Log File Browser displays the filtered list.

    Clear removes all entries in all fields. **Cancel** dismisses the form. When you reenter the form, the values that you entered before the last **OK** are displayed. **Cancel** will undo a **Clear** operation and any edit operation.

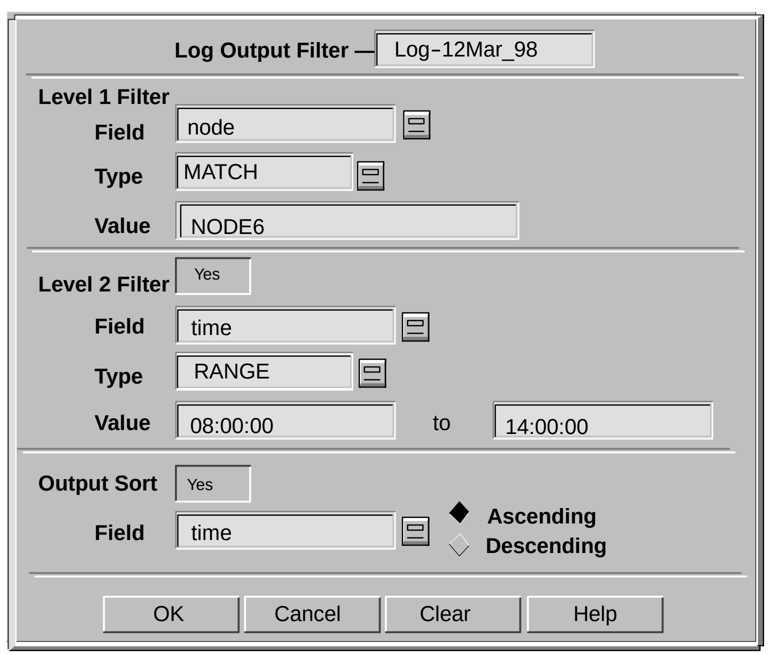


Figure 14-9 Log File Filter Example

You now have several options, indicated by the buttons at the bottom of the Log File Browser.

- **Print** prints the filtered log file.
- **Save** saves the filtered log file and adds it to the list of reports that you can access from the Reports Directory. The saved file is called **<original log file name>.Fxx**, where **xx** is a number from 01 to 99 indicating a specific saved filtered file; for example, **RPLOG.30AUG-1.F02**.

  The only difference between this filtered log file and a standard log file is that the filtered log file is not archived at end-of-day.

- **Filter** allows you to refine your search even further. For example, if you want to see only the entries concerning temperature on NODE6 between 8:00 AM and 2:00 PM, you can refilter the filtered log file. Follow the procedure on page 14-14.

  When you select **Filter**, a prompt appears, asking if you want to save the filtered log file before refiltering it. If you do not save it, the previously filtered file is deleted and replaced by the new filtered file.

- **Delete** deletes the filtered log file and closes the Log File Browser. As long as you remain in the Operate state, however, the Log File Filter keeps the settings that you entered, so you can easily remake a filtered file that was accidentally deleted.
- **Close** dismisses the Log File Browser. If you have not saved the filtered log file, **Close** asks if you want to save it before closing. If you do not save it, the file is deleted when the Log File Browser is closed.

Using the Log File Filter (continued)

Refiltering a Filtered Log File

To refilter the filtered log file of NODE6 between 8:00 AM and 2:00 PM to display only the entries concerning temperature, follow the steps below.

1. Click the **Filter** button at the bottom of the Log File Browser that displays your previously filtered log file.

   If you have already saved and closed the file, you can access it again by clicking the **Directory** icon and selecting the filtered log file from the **Report** directory. The file is displayed in the Log File Browser. Click the **Filter** button.

2. In the Field box under Level 1 Filter, select **tag_descr**.

3. In the Type field, select **MATCH**.

4. In the Value field, enter `*temp*`. The asterisk is a wildcard character that will match zero or more characters. Remember that the filter is case-sensitive.

NOTE: The asterisk character (*) can be used as a wildcard for a MATCH or NOT= comparison. It tells the filter to match zero or more characters. If you want to match a literal asterisk in these fields, you must place a backslash (\ ) character in front of it: *

The wildcard is not supported for comparison types other than MATCH or NOT=. The asterisk in the other comparison types retains its literal meaning, and no preceding backslash is needed.

---

5. Click **OK**. The filter creates a list of log file entries for NODE6 between 8:00 AM and 2:00 PM that include the text string “temp” somewhere in the tag description. The Log File Browser displays the filtered list and deletes the previously filtered log file unless you have saved it.

14.6 Log Messages and Process Groups

You can use the process group feature to arrange for log messages concerning different processes or plant areas to go to different printers. You must make sure that the printer and the appropriate log messages have at least one process group in common. You can assign a printer to a process group when you configure the printer. Refer to the chapter on printers in the *SIMATIC PCS 7 OSx System Administration Manual*.

All log messages appear in the daily log file, but only those log messages that have a process group in common with a printer will print to that printer. You do not assign log messages directly to a process group. Process group membership for log messages is determined in the following ways:

- Alarm and action request messages are printed based on the alarm group or action request sharing a process group with the printer.
- Other log messages that are based on tags are printed if the tag shares a process group with the printer.
- Batch messages and any log messages that do not have tags default to "all process groups" and will always print to any printer that is configured with at least one process group.

Therefore, to send specific log messages to a printer, make sure that the printer has at least one process group in common with any alarms, action requests, and tags that generate those log messages. To configure these process groups, refer to Chapter 10 for alarms, Chapter 12 for action requests, and Chapter 5 for tags.

Chapter 15
Mini-Windows

15.1 Displaying Alarms and Action Requests with a Graphic

OSx allows you to display two scrollable mini-windows in the Operate state that show the alarms and action requests associated with a graphic (Figure 15-1). You can enable these mini-windows from the Event Preferences dialog box. You can then configure particular alarms and action requests to appear with a graphic. As long as the mini-windows are enabled, the system displays them automatically whenever you open the graphic.

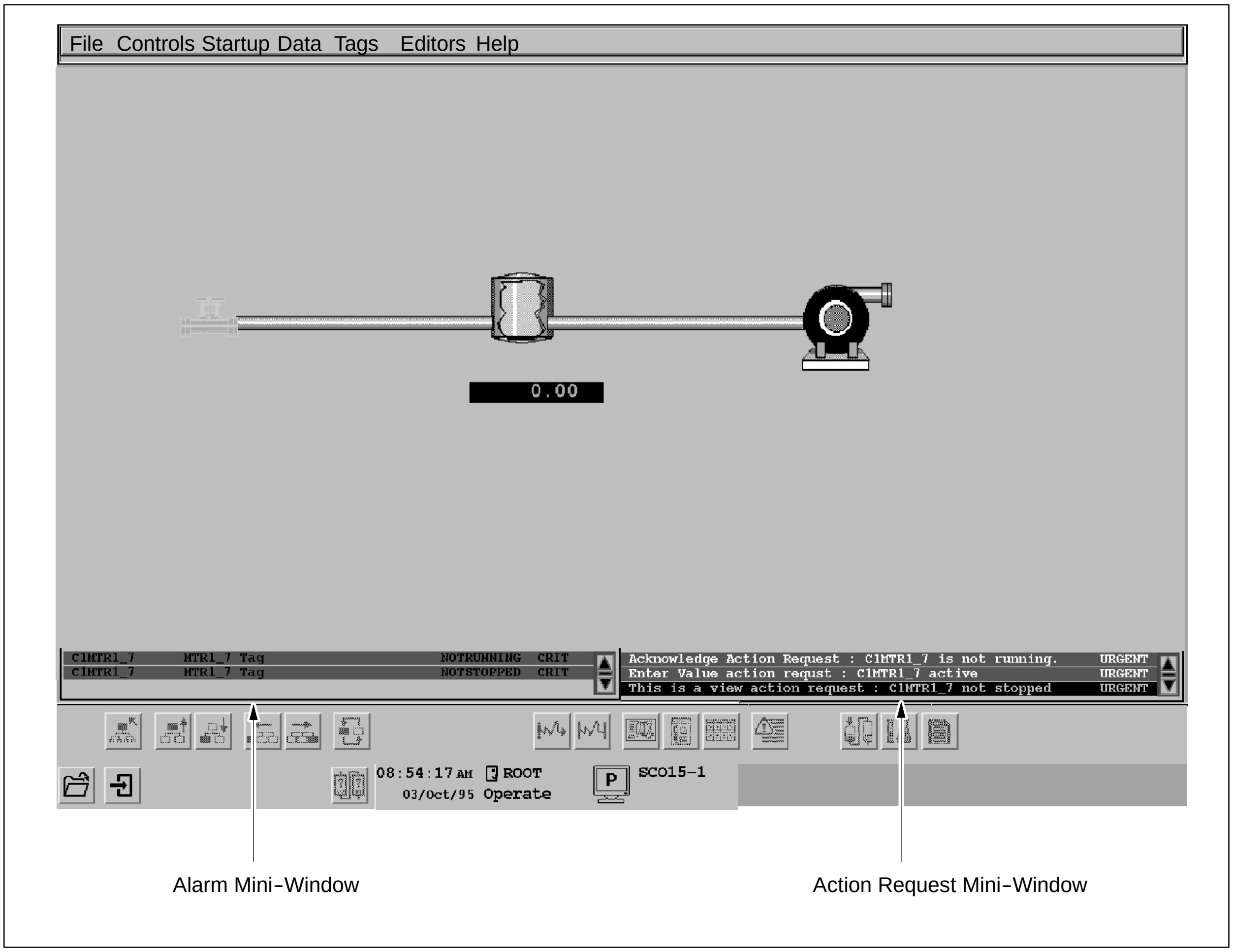


Figure 15-1 Mini-Windows Displayed in Operate State

15.2 Enabling Mini-Windows

Before you can associate alarms and action requests with a graphic, you must enable the mini-windows display. Follow the steps below:

1. Select **Startup->Event Preferences** from the main menu bar. The Event Preferences dialog box appears (Figure 15-2).

2. In the Event Preferences dialog box, toggle the Mini-Windows Display field to **Enabled**. You can now associate alarms and action requests with individual graphics. These alarms and actions requests will appear in two scrollable mini-windows whenever the graphic is displayed on the screen in the Operate state.

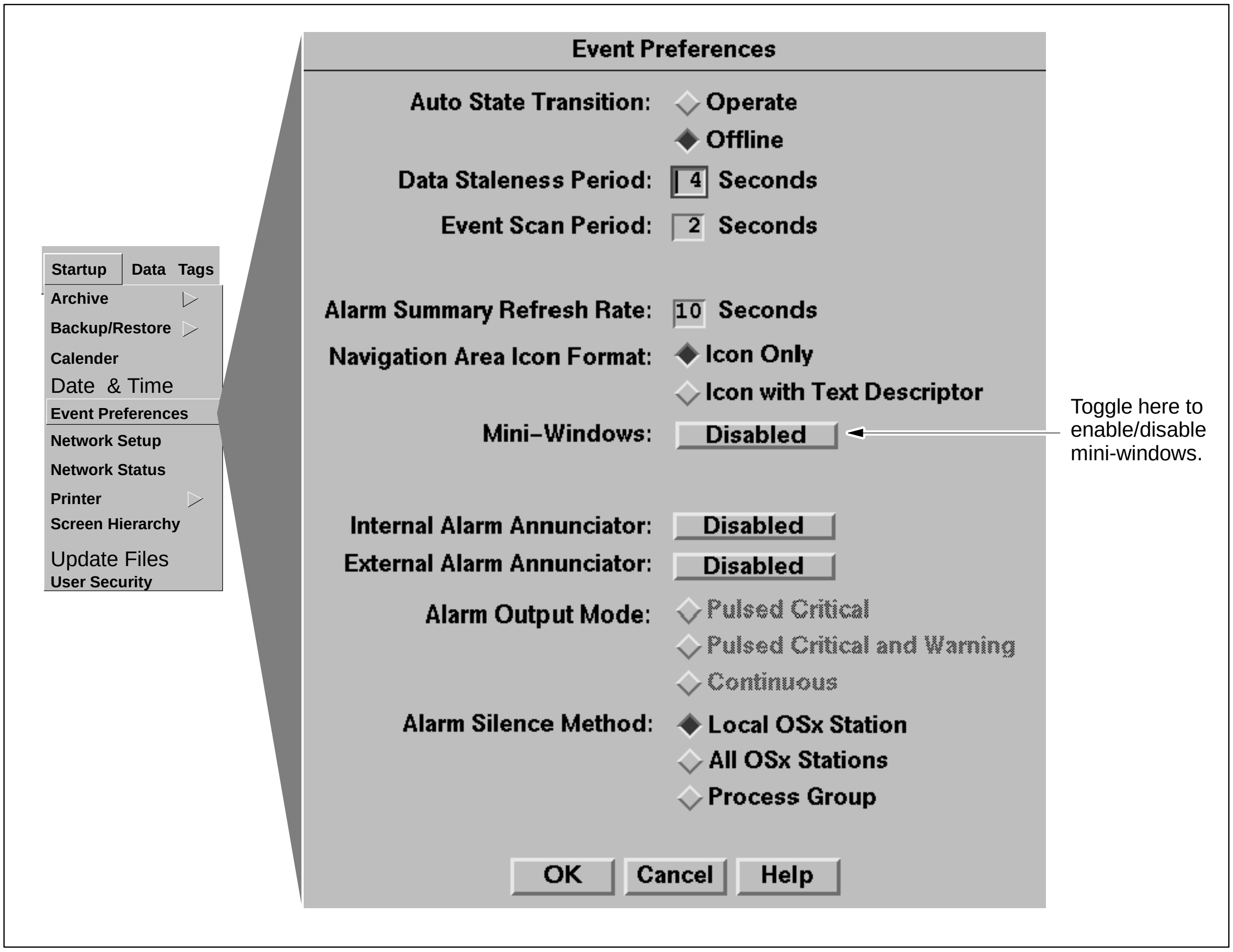


Figure 15-2 Enabling Mini-Windows

15.3 Attaching an Alarm Group to a Graphic

You can associate an alarm group with a graphic so that the alarms in that group appear in a scrollable mini-window on the OSx screen whenever the graphic is displayed. To associate an alarm group with a graphic, follow these steps.

1. Be sure that you have enabled mini-windows (page 15-3).
2. Select **Startup->Screen Hierarchy** from the main menu bar. The Screen Hierarchy Configuration dialog box appears (Figure 15-3).
3. In the Display Type field, select **Graphic**.
4. In the Originate From field, use the long-list display tool to select the name of the graphic.
5. In the Alarm Group field, select the alarm group that you want to associate with the graphic. Then click on **OK**.

   As long as mini-windows are enabled in the Event Preferences dialog box, the active alarms in this alarm group will appear in a scrollable mini-window whenever this graphic is displayed.

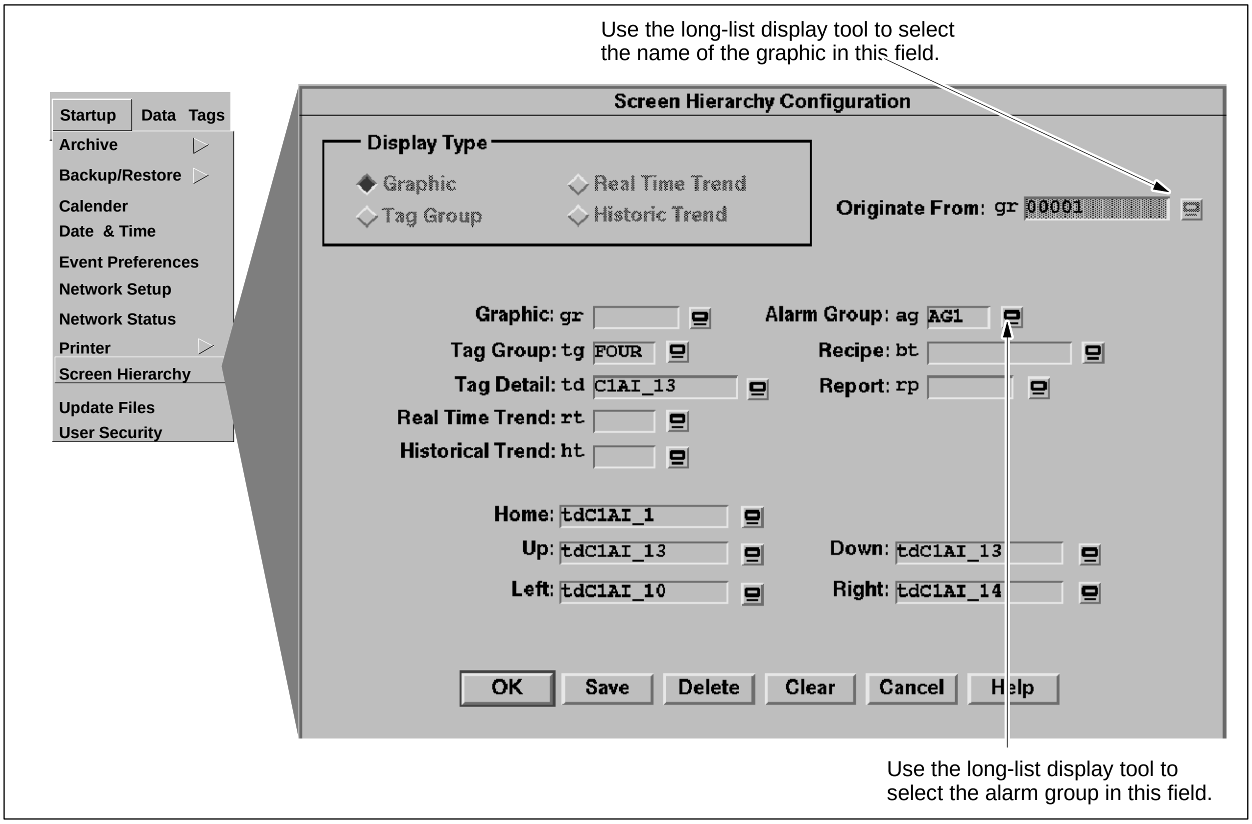


Figure 15-3 Attaching an Alarm Group to a Graphic

15.4 Attaching Action Requests to a Graphic

You can associate action requests with a graphic so they appear in a scrollable mini-window whenever the graphic is displayed. To associate an action request with a graphic, you must link it with an alarm group that you have attached to the graphic (page 15-4). Follow these steps.

1. Be sure that you have enabled mini-windows (page 15-3).
2. Select **Data->Action Request** from the main menu bar, and then select the **Add** button at the bottom of the Action Request Directory. The Action Request Configuration dialog box appears (Figure 15-4).
3. Enter the name of the Action Request in the Request Name field.
4. In the Request Alarm Group field, use the long-list display tool to select the alarm group that is associated with the graphic for which you want to display the action request. Then click on **OK** to save your configuration.

   As long as mini-windows are enabled in the Event Preferences dialog box, this action request will now appear in a scrollable mini-window, if the request has been triggered and not reset or answered, whenever the graphic is displayed.

Use the long-list display tool to select the alarm group associated with the graphic in this field.

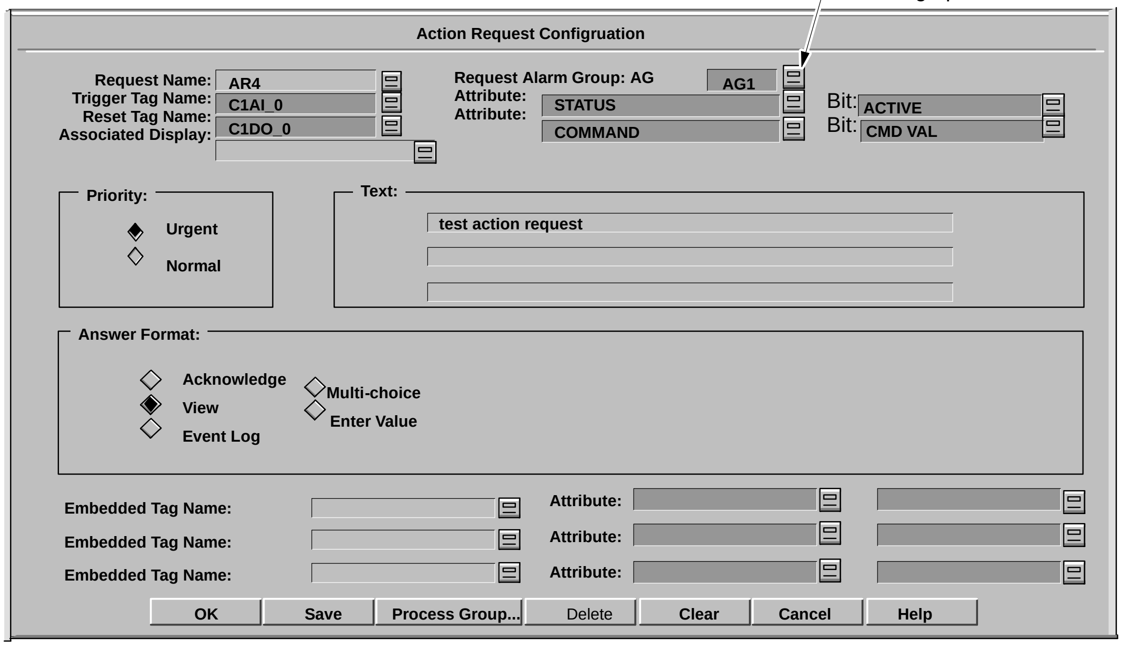


Figure 15-4   Attaching an Action Request to a Graphic

Chapter 16
Window Group Management

16.1 Using Window Groups

As you work with OSx, you may find that you use certain groups of graphic windows over and over. For example, you may frequently display a particular graphic with an associated tag detail and real-time trend. Or you may need to monitor two different trends at the same time. Or you may want to show a historical trend with a tag group of the tags in the trend.

Rather than call up each graphic window and reposition it separately every time you want to show a particular layout, you can configure a combination of graphic windows in a window group. You can then access this entire window group as if it were a single graphic window (Figure 16-1).

You can create as many as 256 window groups, and each window group can contain a main graphic window with up to two subgraphics, for a total of three graphic windows per window group. Only graphics, tag details, historical and real-time trends, and one tag group can be included in a window group. If a tag group is included, it must be the main graphic.

See the appendix on OSx commands in the *SIMATIC PCS 7 OSx Graphical Editor Manual* for information on how to perform the following operations from a button, menu, or select list on a graphic in the window group:

- Adding window groups
- Displaying the window group directory
- Adding and deleting subgraphics

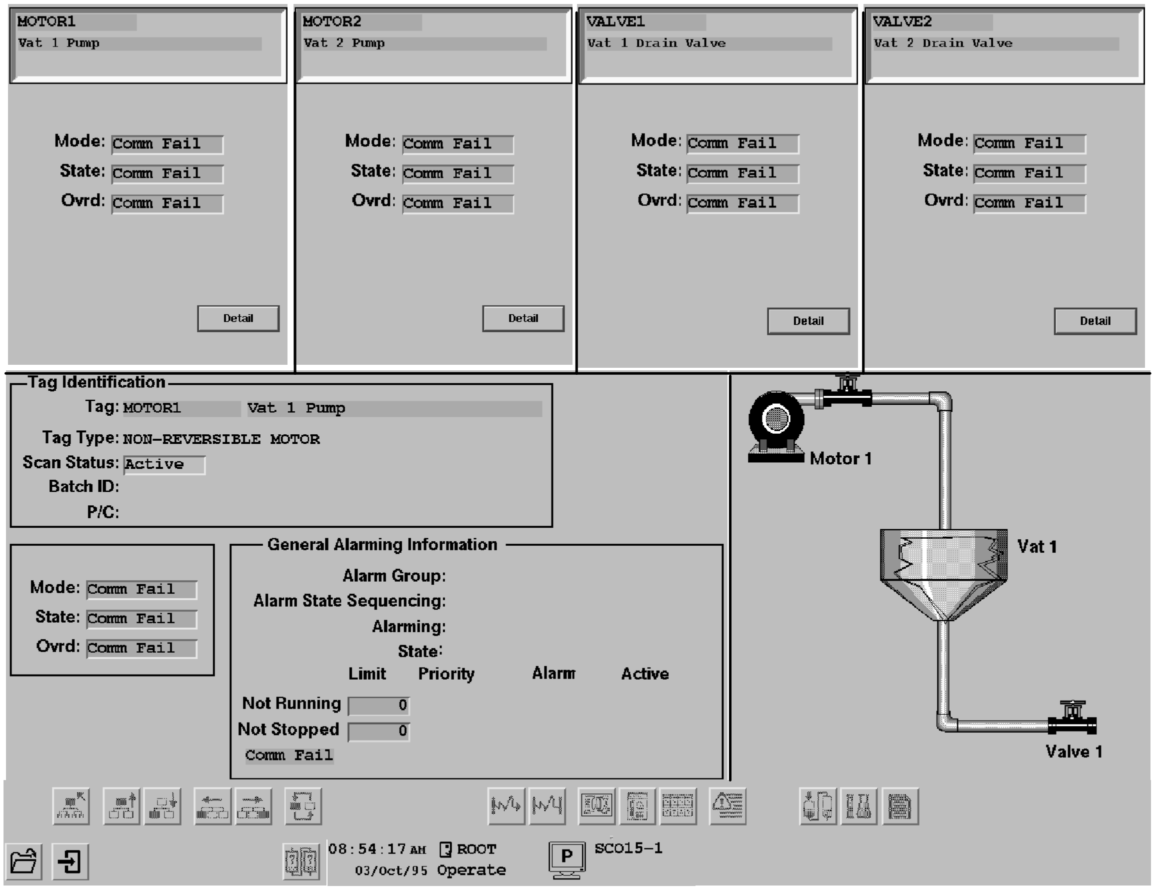


Figure 16-1 Window Group Example

16.2 Configuring a Window Group

Guidelines

As you plan your window group arrangement, follow these guidelines:

- Remember that you can use only graphics, tag details, trends, and a single tag group in each window group that you create.
- You can display up to three of these graphic windows together in a window group. One of the windows is called the main graphic, and the other two are called subgraphics.
- If a tag group is present, it must be the main graphic. The faceplates within the tag group cannot be rearranged.
- Only graphics, tag details, and trends can be resized. You cannot resize tag groups.
- You can arrange windows in overlapping positions.
- When OSx is in the Offline or Operate state, you can configure window group names and descriptions, and you can view, delete, and resave the descriptions of previously captured groups. However, you must set OSx to the Operate state in order to capture a window group configuration.
- If you position a graphic partially off the screen, you can still capture the window group. But when you display the window group, the graphic will appear positioned fully on the screen, potentially overlapping other graphic windows.

Scroll Bars on Graphics in Window Groups

If you configure a graphic whose initial size (width and height) is smaller than its maximum size, OSx displays the graphic with scroll bars when you access it as a graphic in the Operate state.

However, if the graphic is captured in a window group, no scroll bars are displayed in the window group, and some data may not be visible. In general, if you plan to use a graphic in a window group, do not edit its Format Properties to set its initial size smaller than its maximum size.

Window Group Configuration

To configure a window group, follow the steps below.

1. Be sure that OSx is in the Operate state.

2. Using the Directory icon in the navigation area, open the graphic windows that you want to display as a window group.

   The first graphic that you select becomes the main graphic. If you want to include a tag group in the window group, the tag group must be the main graphic.

   To open additional graphics in your window group, click the Directory icon and then click the **Sub Graphic** option in the list of directories that appears. From there you can select graphics, tag details, and real-time and historical trends.

3. Size the graphic windows as needed and arrange them in the positions where you want them to appear.

   Use **Alt F7** to select the graphic for repositioning. Move the cursor to the new position and click once to release the graphic.

   Use **Alt F8** to select the graphic for resizing. Then move the cursor to the edge of the graphic, hold the left mouse button down, and drag the graphic to the new size.

4. When your window group is arranged the way you want it to appear, click on **Data->Window Group** on the main menu bar. The Window Group Configuration dialog box appears (Figure 16-2 on page 16-7).

5. Enter a unique name for the window group in the Window Group Name field to create a new window group. The space, the character combinations -> and <- and the characters ; \ , ” are invalid for window group names.

6. Enter a description of the window group in the Description field. The characters ; \ ” are invalid for window group descriptions.

7. Select **Capture** at the bottom of the Window Group Configuration dialog box, and select **Yes** in the confirmation dialog box.

8. After the capture, select **OK** to save your window group configuration and exit the configuration utility.

   The system saves the window group and adds it to the Window Group Directory. You can now access it from the Directory icon whenever the system is in the Operate state. (See page 16-10 for the procedure.)

Configuring a Window Group (continued)

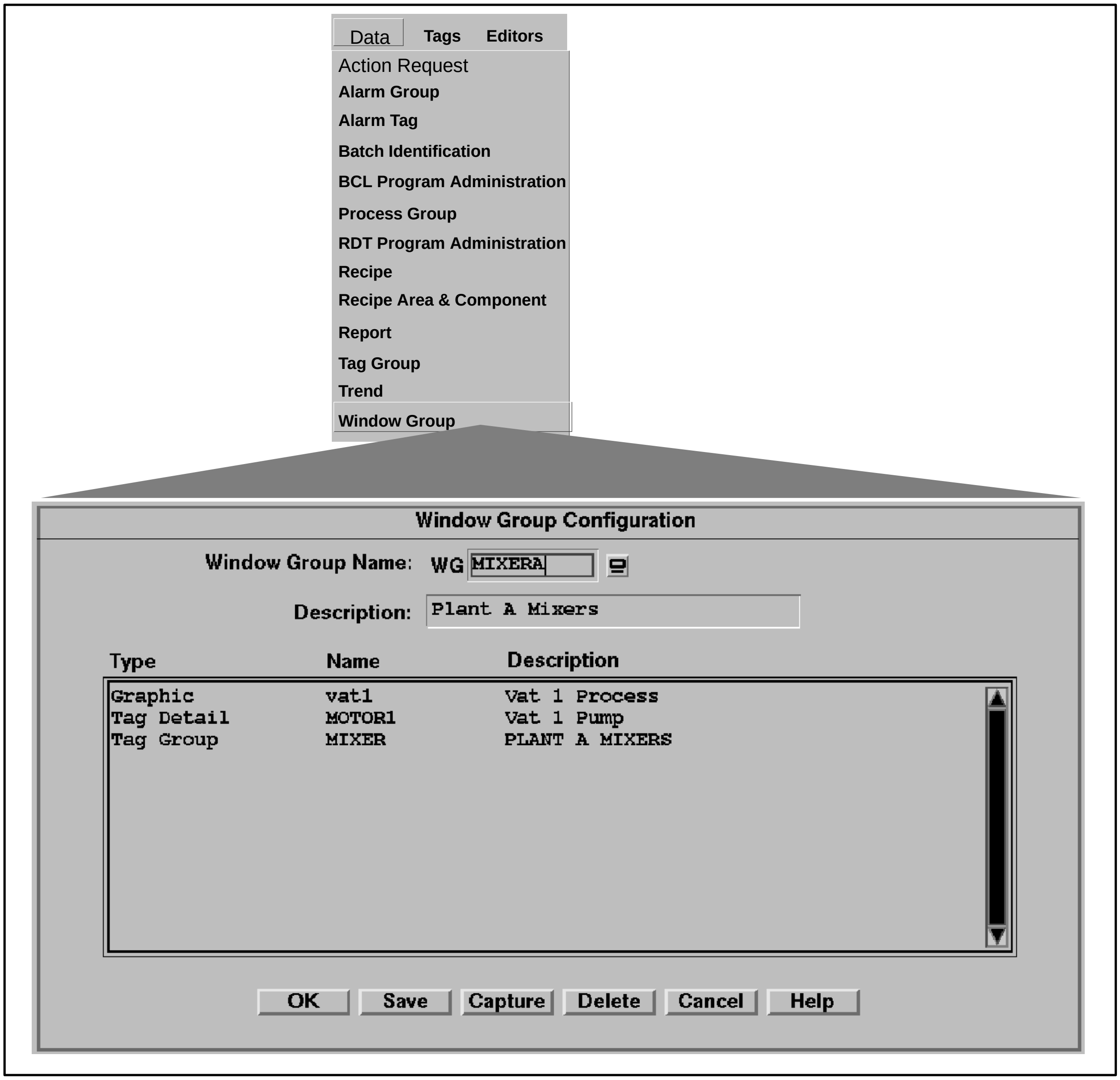


Figure 16-2 Window Group Configuration Dialog Box

Using Formats from Previous Versions of PCS/OSx

If you want to use a custom graphic, tag detail, historical trend or real-time trend format from a previous release of PCS/OSx, you must modify the format so that it works properly with window groups.

To modify the format, follow these steps:

1. On the OSx station where the format is located, open an XTerm window, and log in as `tistar`.

2. Type the following two commands, pressing **Enter** after each:

```
cd /usr/tistar/opr/data/valid/fmt

fmt_unload <name>.fmt -i. -o.
```

   where `<name>` is the name of the format that you want to transfer, including its prefix. Each file type has a prefix that precedes the name of the display:

   gr – Graphic
   td – Tag Detail
   ht – Historical Trend
   rt – Real Time Trend

3. Type the following two commands, pressing **Enter** after each:

```
cat <name>.asc | sed ’s/vb_lsvr = ””/vb_lsvr =
”@gs_serve”/’ > <name>.new

mv -f <name>.new <name>.asc
```

   where `<name>` is the name of the format, including its prefix.

4. To copy the format to an MO, insert the MO in the MO drive, type the following command and press **Enter**:

```
ls /usr/tistar/opr/data/valid/fmt/<name>.asc | cpio
 -ov -H odc > /dev/rmo0
```

5. Remove the MO from the MO drive.

6. On the OSx station where you want to place the format, open an XTerm window, and log in as `tistar`.

7. To place the format in the proper directory on the new OSx station, insert the MO in the MO drive of that station, type the following command, and press **Enter**.

```
cpio -iv -H odc < /dev/rmo0
```

8. Type the following two commands, pressing **Enter** after each:

```
cd /usr/tistar/opr/data/valid/fmt
```

```
fmt_load <name>.asc -i. -o.
```

where `<name>` is the name of the format, including its prefix.

16.3 Displaying a Window Group

To display a window group that you have configured, follow the steps below:

1. Be sure OSx is in the Operate state.

2. Click on the Directory icon in the navigation area at the bottom of the screen. A list of OSx display types appears.

3. Click on **Window Group** in the list of display types. A list of all the window groups that you have configured appears (Figure 16-3).

4. Click on the name of the window group that you want to display. The window group appears exactly as you configured it.

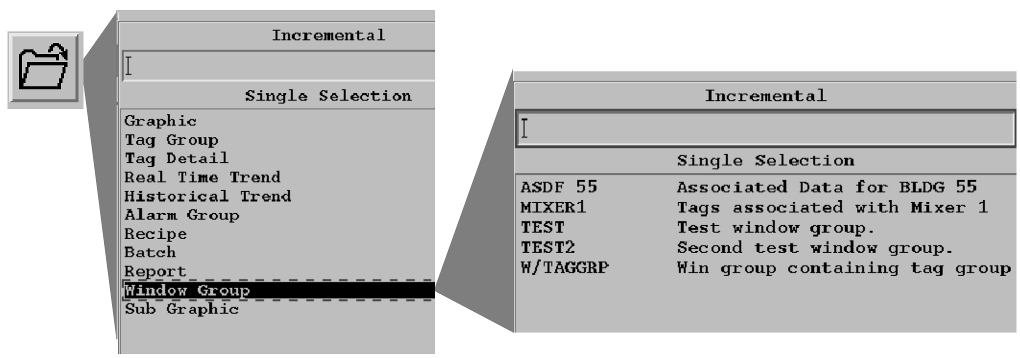


Figure 16-3 Displaying a Window Group

16.4 Modifying a Window Group

To modify the arrangement of a previously configured window group, follow the steps below:

1. Be sure OSx is in the Operate state.
2. Display the window group that you want to modify, using the procedure on page 16-10. Rearrange the graphic windows in the positions where you want them to appear.

   You can also delete a graphic, by clicking the right mouse button and selecting **Close Display** from the menu, and replace it with another.

   A main graphic can only be replaced by another main graphic, and a subgraphic can only be replaced by another subgraphic.
3. When your window group is arranged the way you want it to appear, click **Data->Window Group** on the main menu bar. The Window Group Configuration dialog box appears (Figure 16-2 on page 16-7).
4. In the Window Group Name field, enter the name of the window group that you are modifying. The list of graphics for the previous arrangement of this window group appears in the large box.
5. Select **Capture** at the bottom of the Window Group Configuration dialog box, and select **Yes** in the confirmation dialog box.
6. Select **OK** to confirm your new window group configuration and exit the configuration utility.

Appendix A
Configuration Planning Sheets

A.1 Configuration Checklist

Use the planning sheets included in the appendix to ensure a more organized, efficient configuration.

- Configure OSx stations and control nodes (pages A-3 and A-4)
- Configure tag capacity (page A-5)
- Configure RBE tag memory requirements (page A-6)
- Configure tags (pages A-7 through A-21)
- Configure alarm groups (page A-22)
- Configure alarms (page A-23)
- Configure tag groups (page A-24)
- Configure trends (page A-25)
- Configure security (page A-26)

A.2 Operations Station Configuration Planning Sheet

Station Name

Station Description

Connection Type

Main Network Address

Alternate Network Address

Station Name

Station Description

Connection Type

Main Network Address

Alternate Network Address

Station Name

Station Description

Connection Type

Main Network Address

Alternate Network Address

A.3 Control Node Configuration Planning Sheet

Node Name	
Node Description	
Connection Type	
Controller Type	
Main Network Address	
Alternate Network Address	

Node Name	
Node Description	
Connection Type	
Controller Type	
Main Network Address	
Alternate Network Address	

Node Name	
Node Description	
Connection Type	
Controller Type	
Main Network Address	
Alternate Network Address	

A.4 Tag Capacity Configuration Planning Sheet

Tag Type	Maximum Records	Deadband Default	Comments
Analog Input (AI)	☐☐☐☐	☐☐☐☐	______
Analog Output (AO)	☐☐☐☐	☐☐☐☐	______
Area (AREA)	☐☐☐☐		______
Batch (BCH)	☐☐☐☐		______
Calculate Variable (CALC)	☐☐☐☐	☐☐☐☐	______
Counter (CTR)	☐☐☐☐		______
Digital Input (DI)	☐☐☐☐		______
Digital Output (DO)	☐☐☐☐		______
Digital Input – 10 Bit (DI10)	☐☐☐☐		______
Digital Output – 10 Bit (DO10)	☐☐☐☐		______
Integer Variable (IVAR)	☐☐☐☐		______
Loop (LOOP)	☐☐☐☐	☐☐☐☐	______
Motor (MTR1)	☐☐☐☐		______
Motor – Reversible (RMTR)	☐☐☐☐		______
Motor – Two Speed (MTR2)	☐☐☐☐		______
Text (TEXT)	☐☐☐☐		______
Timer (TMR)	☐☐☐☐	☐☐☐☐	______
Unit (UNIT)	☐☐☐☐		______
Valve – Single Feedback (VLV1)	☐☐☐☐		______
Valve – Dual Feedback (VLV2)	☐☐☐☐		______

A.5 Controller Memory Requirement for Tags Planning Sheet

You can set up this planning sheet in a spreadsheet format. After you enter the number of tags in cells D2–D17, the bytes per tag type (cells E2–E17) are summed in cell A2.

	A	B	C	D	E
1	**Total Memory All Tags**	**Tag Type**	**Bytes per Tag**	**Number of Tags**	**Bytes per Tag Type**
2	E2+E3+E4+E5+ E6+E7+E8+E9+ E10+E11+E12+E13+ E14+E15+E16+E17	AI	42		C2 × D2
3		AO	42		C3 × D3
4		AREA	42		C4 × D4
5		BCH	42		C5 × D5
6		CALC	36		C6 × D6
7		CTR	42		C7 × D7
8		Device	48		C8 × D8
9		DI	28		C9 × D9
10		DO	28		C10 × D10
11		DI10	28		C11 × D11
12		DO10	28		C12 × D12
13		IVAR	36		C13 × D13
14		LOOP	54		C14 × D14
15		TEXT	48		C15 × D15
16		TMR	42		C16 × D16
17		UNIT	48		C17 × D17

A.6 Analog Input (AI) Tag Configuration Planning Sheet

Sheet ___ of ___

CONTROL NODE: └┴┴┴┴┴┴┴┴┴┴┴┘

TAG NAME: └┴┴┴┴┴┴┴┴┴┴┴┘

DESCRIPTION: └┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┘

PROCESS GROUP: └┴┴┴┴┴┴┴┘

PARENT: └┴┴┴┴┴┴┴┴┴┴┴┘

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	20%	AUTOLOG	INITIAL VALUE
STATUS			YES NO		YES NO	
L_RANGE			YES NO		YES NO	
H_RANGE			YES NO		YES NO	
PV			YES NO	YES NO	YES NO	
TARGET			YES NO	YES NO	YES NO	
HH_ALARM			YES NO		YES NO	
H_ALARM			YES NO		YES NO	
L_ALARM			YES NO		YES NO	
LL_ALARM			YES NO		YES NO	
H_DEV (ORANGE)			YES NO		YES NO	
L_DEV (YELLOW)			YES NO		YES NO	
ROC_ALARM			YES NO		YES NO	
AL_DEADBAND*						

* This attribute cannot be networked

A.7 Analog Output (AO) Tag Configuration Planning Sheet

Sheet ___ of ___

CONTROL NODE:

TAG NAME:

DESCRIPTION:

PROCESS GROUP:

PARENT:

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	20%	AUTOLOG	INITIAL VALUE
STATUS			YES NO		YES NO	
L_RANGE			YES NO		YES NO	
H_RANGE			YES NO		YES NO	
OUT			YES NO	YES NO	YES NO	
MODE			YES NO		YES NO	

A.8 Area (AREA) Tag Configuration Planning Sheet

Sheet ___ of ___

CONTROL NODE:

TAG NAME:

DESCRIPTION:

PROCESS GROUP:

PARENT:

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	AUTOLOG	INITIAL VALUE
STATUS			YES NO	YES NO	
HOLD_REQ			YES NO	YES NO	
SCALE_FACTOR			YES NO	YES NO	
SCALE_HIGH*					
SCALE_LOW*					
PLC_REC_REQ			YES NO	YES NO	

* These attributes cannot be networked

The Area tag type is described in the *SIMATIC PCS 7 OSx Recipe Manual*.

A.9 Calculated Variable (CALC) Tag Configuration Planning Sheet

Sheet ___ of ___

CONTROL NODE: |_|_|_|_|_|_|_|_|_|_|_|_|

TAG NAME:

DESCRIPTION: |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

PROCESS GROUP: |_|_|_|_|_|_|_|_|

PARENT: |_|_|_|_|_|_|_|_|_|_|_|_|

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	20%	AUTOLOG	INITIAL VALUE
STATUS			YES NO		YES NO	
L_RANGE			YES NO		YES NO	
H_RANGE			YES NO		YES NO	
VALUE			YES NO	YES NO	YES NO	

A.10 Counter (CTR) Tag Configuration Planning Sheet

Sheet ___ of ___

CONTROL NODE: |_|_|_|_|_|_|_|_|_|_|_|_|

TAG NAME: |_|_|_|_|_|_|_|_|_|_|_|_|

DESCRIPTION: |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

PROCESS GROUP: |_|_|_|_|_|_|_|_|

PARENT: |_|_|_|_|_|_|_|_|_|_|_|_|

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	AUTOLOG	INITIAL VALUE
STATUS			YES NO	YES NO	
L_RANGE			YES NO		
PRESET			YES NO	YES NO	
CURRENT			YES NO	YES NO	

A.11 Device Tag Configuration Planning Sheet

PROCESS I/O: DEVICE
DEVICE CONFIGURATION PLANNING SHEET

Sheet ___ of ___

VLV1 (single-feedback valve), VLV2 (dual-feedback valve),
MTR1 (motor), MTR2 (two-speed motor), RMTR (reversible motor)

CONTROL NODE:

TAG NAME:

DESCRIPTION:

PROCESS GROUP:

PARENT:

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	AUTOLOG	INITIAL VALUE
STATUS			YES NO	YES NO	
OVERRIDE			YES NO	YES NO	
MODE_CMD			YES NO	YES NO	
SETPOINT			YES NO	YES NO	
TIMEOUT_1			YES NO	YES NO	
TIMEOUT_2			YES NO	YES NO	

A.12 Digital Input (DI) Tag Configuration Planning Sheet

Sheet ___ of ___

CONTROL NODE:

TAG NAME:

DESCRIPTION:

PROCESS GROUP:

PARENT:

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	AUTOLOG	INITIAL VALUE
STATUS			YES NO	YES NO	

A.13 Digital Input – 10-Bit (DI10) Tag Configuration Planning Sheet

Sheet ___ of ___

CONTROL NODE:

TAG NAME:

DESCRIPTION:

PROCESS GROUP:

PARENT:

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	AUTOLOG	INITIAL VALUE
STATUS			YES NO	YES NO	

A.14 Digital Output (DO) Tag Configuration Planning Sheet

Sheet ___ of ___

CONTROL NODE: ⌊_|_|_|_|_|_|_|_|_|_|_|_⌋

TAG NAME:

DESCRIPTION: ⌊_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_⌋

PROCESS GROUP: ⌊_|_|_|_|_|_|_|_⌋

PARENT: ⌊_|_|_|_|_|_|_|_|_|_|_|_⌋

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	AUTOLOG	INITIAL VALUE
STATUS			YES NO	YES NO	
COMMAND			YES NO	YES NO	

A.15 Digital Output – 10-Bit (DO10) Tag Configuration Planning Sheet

Sheet ___ of ___

CONTROL NODE: ____________

TAG NAME: ____________

DESCRIPTION: ______________________________

PROCESS GROUP: ________

PARENT: ____________

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	AUTOLOG	INITIAL VALUE
STATUS			YES NO	YES NO	
DATA_VAL1			YES NO	YES NO	
DATA_VAL2			YES NO	YES NO	
DATA_VAL3			YES NO	YES NO	
DATA_VAL4			YES NO	YES NO	
DATA_VAL5			YES NO	YES NO	
DATA_VAL6			YES NO	YES NO	
DATA_VAL7			YES NO	YES NO	
DATA_VAL8			YES NO	YES NO	
DATA_VAL9			YES NO	YES NO	
DATA_VAL10			YES NO	YES NO	

A.16 Integer Variable (IVAR) Tag Configuration Planning Sheet

Sheet ___ of ___

CONTROL NODE:

TAG NAME:

DESCRIPTION:

PROCESS GROUP:

PARENT:

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	AUTOLOG	INITIAL VALUE
STATUS			YES NO	YES NO	
L_RANGE			YES NO		
H_RANGE			YES NO		
VALUE			YES NO	YES NO	

A.17 Loop (LOOP) Tag Configuration Planning Sheet

CONTROL NODE: |_|_|_|_|_|_|_|_|_|_|_|_| Sheet ___ of ___

TAG NAME:

DESCRIPTION: |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

PROCESS GROUP: |_|_|_|_|_|_|_|_|

PARENT: |_|_|_|_|_|_|_|_|_|_|_|_|

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	20%	AUTOLOG	INITIAL VALUE
STATUS			YES NO		YES NO	
L_RANGE			YES NO		YES NO	
H_RANGE			YES NO		YES NO	
PV			YES NO	YES NO	YES NO	
SP			YES NO		YES NO	
HH_ALARM			YES NO		YES NO	
H_ALARM			YES NO		YES NO	
L_ALARM			YES NO		YES NO	
LL_ALARM			YES NO		YES NO	
H_DEV (ORANGE)			YES NO		YES NO	
L_DEV (YELLOW)			YES NO		YES NO	
ROC_ALARM			YES NO		YES NO	
AL_DEADBAND			YES NO		YES NO	
OUT			YES NO	YES NO	YES NO	
MODE			YES NO		YES NO	
GAIN			YES NO		YES NO	
RATE			YES NO		YES NO	
RESET			YES NO		YES NO	

A.18 Text (TEXT) Tag Configuration Planning Sheet

Sheet ___ of ___

CONTROL NODE: |_|_|_|_|_|_|_|_|_|_|_|_|

TAG NAME: |_|_|_|_|_|_|_|_|_|_|_|_|

DESCRIPTION: |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

PROCESS GROUP: |_|_|_|_|_|_|_|_|

PARENT: |_|_|_|_|_|_|_|_|_|_|_|_|

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	AUTOLOG	INITIAL VALUE
STATUS			YES NO	YES NO	
TEXT_1			YES NO	YES NO	
TEXT_2			YES NO	YES NO	
TEXT_3			YES NO	YES NO	

A.19 Timer (TMR) Tag Configuration Planning Sheet

Sheet ___ of ___

CONTROL NODE: ⌊_|_|_|_|_|_|_|_|_|_|_|_⌋

TAG NAME: ⌊_|_|_|_|_|_|_|_|_|_|_|_⌋

DESCRIPTION: ⌊_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_⌋

PROCESS GROUP: ⌊_|_|_|_|_|_|_|_⌋

PARENT: ⌊_|_|_|_|_|_|_|_|_|_|_|_⌋

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	AUTOLOG	INITIAL VALUE
STATUS			YES NO	YES NO	
L_RANGE			YES NO	YES NO	
H_RANGE			YES NO	YES NO	
PRESET			YES NO	YES NO	
CURRENT			YES NO	YES NO	

A.20 Unit (UNIT) Tag Configuration Planning Sheet

Sheet ___ of ___

CONTROL NODE:

TAG NAME:

DESCRIPTION:

PROCESS GROUP:

PARENT:

ATTRIBUTES	MEMORY TYPE/ADDRESS	NUMBER OF LOCATIONS	UPLOAD	AUTOLOG	INITIAL VALUE
STATUS			YES NO	YES NO	
COMMAND			YES NO	YES NO	
MODE_CMD			YES NO	YES NO	
BCH_REQ			YES NO	YES NO	
BCH_REQ_INFO			YES NO	YES NO	
BCH_REQ_RESP			YES NO	YES NO	

The Unit tag type is described in the *SIMATIC PCS 7 OSx Batch Programming Manual*.

A.21 Alarm Group Configuration Planning Sheet

Sheet ___ of ___

Group ID

Description

Associated Display

Alarm Group State Tag

Attribute

Default Alarm State

State	Description
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	

A.22 Alarm Tag Configuration Planning Sheet

Sheet ___ of ___

Tag Name

Tag Type

Alarm Group

Alarm State

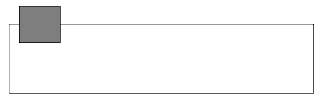
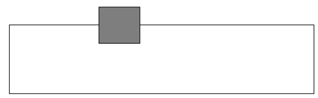
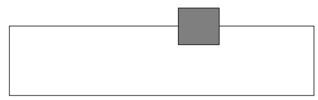
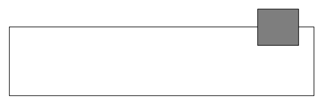
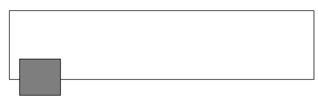
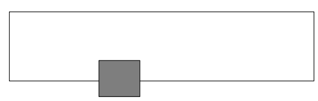
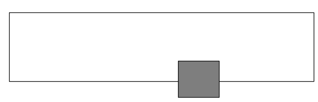
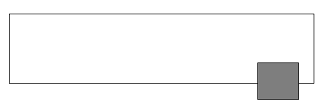
Alarm Type	Priority	Enabled Disabled	Alarm Suppression Information			
			Tag	Attribute	Bit Name	Suppress When
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0
	C W I M	EN DIS				1 0

A.23 Tag Group Configuration Planning Sheet

Tag Group

Sheet ___ of ___

Description

Tag Name	Faceplate Position
	
	
	
	
	
	
	
	

A.24 Trend Configuration Planning Sheet

Trend Name | | | | | Sheet ___ of ___

Historical Viewing Window Time Base: Days Hours Minutes

Real Viewing Window Time Base: Hours Minutes

Analog Data

Tag	Attribute	Rate No:	Hours Minutes Sec
Range ... To ...		Duration No:	Days Hours Minutes
Tag	Attribute	Rate No:	Hours Minutes Sec
Range ... To ...		Duration No:	Days Hours Minutes
Tag	Attribute	Rate No:	Hours Minutes Sec
Range ... To ...		Duration No:	Days Hours Minutes
Tag	Attribute	Rate No:	Hours Minutes Sec
Range ... To ...		Duration No:	Days Hours Minutes

Digital Data

Tag	Attribute	Rate No:	Hours Minutes Sec
Bit Name		Duration No:	Days Hours Minutes
Tag	Attribute	Rate No:	Hours Minutes Sec
Bit Name		Duration No:	Days Hours Minutes
Tag	Attribute	Rate No:	Hours Minutes Sec
Bit Name		Duration No:	Days Hours Minutes
Tag	Attribute	Rate No:	Hours Minutes Sec
Bit Name		Duration No:	Days Hours Minutes

A.25 Security Configuration Planning Sheets

User ID [| | | | | | |]

Password [| | | | | | | | | | | |]

Sheet ___ of ___

Privilege	Privilege Status	
Primary Control	On	Off
Secondary Control	On	Off
Tertiary Control	On	Off
Control Tuning	On	Off
Process Cyclic Function	On	Off
Real-Time Trend	On	Off
Historical Trend	On	Off
Report Configuration	On	Off
Operator Level Report Request	On	Off
System Configuration	On	Off
Database Administration	On	Off
System Startup/Shutdown	On	Off
System Maintenance	On	Off
Management Level Report Request	On	Off
Control Application	On	Off
Action Request Configuration	On	Off
Batch Control	On	Off
Role Switch	On	Off
Operation	On	Off
Graphical Editor	On	Off
Re-Sync Control	On	Off

User ID | | | | | | | | |

Password | | | | | | | | | | | | | |

Sheet ___ of ___

Privilege	Privilege Status	
Primary Control	On	Off
Secondary Control	On	Off
Tertiary Control	On	Off
Control Tuning	On	Off
Process Cyclic Function	On	Off
Real-Time Trend	On	Off
Historical Trend	On	Off
Report Configuration	On	Off
Operator Level Report Request	On	Off
System Configuration	On	Off
Database Administration	On	Off
System Startup/Shutdown	On	Off
System Maintenance	On	Off
Management Level Report Request	On	Off
Control Application	On	Off
Action Request Configuration	On	Off
Batch Control	On	Off
Role Switch	On	Off
Operation	On	Off
Graphical Editor	On	Off
Re-Sync Control	On	Off

Appendix B

Navigating without a Mouse

B.1 Introduction

The following tables define keystrokes that allow you to navigate without a mouse. Use these key combinations to simulate mouse commands when a mouse is inoperative or unavailable. For example, you can silence an alarm by pressing **Ctrl** and **Shift** and then **S,** instead of clicking on the alarm with the mouse button. To use the above combination, press and hold down the **Ctrl** and **Shift** keys, then press **S**.

B.2 Navigating Icons

An icon is a small graphical representation of a window. When opened, it becomes a window with all of its displayed contents. Table B-1 shows the keystrokes for icon navigation, and their associated menus. Use these keys to select an icon, display a window from an icon, iconifiy an opened window, move an icon on the screen, or delete an icon.

Table B-1 Icon Key Definitions

Icon Functions	Key Combinations
Displays the icon window menu.	**Shift ESC**
Raises icon to front.	**Alt F3**
Dismisses icon and corresponding client.	**Alt F4**
Restores icon to a window.	**Alt F5**
Lowers icon to back.	**Alt F6**
Moves an icon. Use arrows to move in small increments, **Ctrl** plus arrows to move in large increments. To end movement, press **Enter**.	**Alt F7**
Changes the icon to a full screen window.	**Alt F10**

B.3 OSx Key Navigation

Table B-2 shows the keystroke combinations used for the OSx Windows environment. These keys are configured to simulate window button presses (such as **F3** for **SAVE**), erase fields, move within and between fields, and open pop-up lists.

Table B-2 OSx Key Definitions

OSx Window Functions	Key Combinations
Simulates pressing the **HELP** button.	**F1**
Simulates pressing the **OK** button.	**F2**
Simulates pressing the **SAVE** button.	**F3**
Simulates pressing the **ADD** button.	**F5**
Simulates pressing the **DELETE** button.	**F6**
Simulates pressing the **COPY** button.	**F7**
Simulates pressing the **MODIFY** button.	**F8**
Simulates pressing the **START** button.	**F9**
Calls up the menu-bar in an active window, pointing on the leftmost cell.	**F10***
Simulates pressing the **PRINT** button.	**F11**
Simulates pressing the **CANCEL** button.	**F12**
Cancels selection in a menu-bar.	**ESC**
Selects the button under the cursor. (Cannot use in a select-list, when cursor is a cross: **X**.)	**Spacebar**
Deletes the character to the left of the cursor.	**Backspace**
Deletes the character to the left of the cursor.	**Delete**
Erases from current position to the end of the line.	***Keypad /***
Erases a field.	***Keypad ****
Tabs right to the next field or menu selection.	**Tab**
Tabs left to the next field or menu selection.	**Shift Tab**
Enters typed data or menu selection.	***Keypad* Enter**
Enters typed data or menu selection.	**Return**
Moves the cursor one menu selection in the direction of the arrow or one character in a field. The down arrow (↓) also pops up selection list and moves selection down in a menu-bar.	← → ↑ ↓
*To move between items in the menu bar of an active window, press **F10** to select the left-most cell, then **Ctrl→** or **Ctrl Shift→** to move to the right through the other selections.	

B.4 Navigating Windows

You can use keystrokes to manipulate windows on the screen without a mouse. Table B-3 shows keys you can use with XTerm Root windows and OSx application windows to access window menus, open and close windows, select windows, size windows, and move the cursor on the screen. Note that some windows do not support all the key combinations listed in Table B-3.

Table B-3 Window Key Definitions

Window Functions	Key Combinations
Displays the root window menu.	**Ctrl ESC**
Displays the window menu.	**Shift ESC**
Circles focus downward through window/icon stack.	**Alt** ↓
Circles focus upward through window/icon stack.	**Alt** ↑
Raises window to front of screen.	**Alt** →
Lowers window to back.	**Alt** ←
Refreshes window.	**Alt R**
Lowers window to back.	**Alt F3**
Closes the window and its corresponding client.	**Alt F4**
Raises window to front.	**Alt F6**
Moves a window. Use arrows to move in small increments, **Ctrl** plus arrows to move in large increments. End movement by hitting **Return**.	**Alt F7**
Resizes a window. Use arrows to size in small increments, **Ctrl** plus arrows to size in large increments. End resizing by hitting **Return**.	**Alt F8**
Minimizes the window to an icon.	**Alt F9**
Maximizes the window to fill the screen.	**Alt F10**
Moves the cursor one pixel in the direction of the arrow.	**Ctrl** ← **Ctrl** → **Ctrl** ↑ **Ctrl** ↓
Moves the cursor 64 pixels in the direction of the arrow.	**Ctrl Shift** ← **Ctrl Shift** → **Ctrl Shift** ↑ **Ctrl Shift** ↓
Adds one to the pixel cursor increment.	**Ctrl +**
Subtracts one from the pixel cursor movement. You cannot decrement below ten pixels.	**Ctrl -**
Draws a large crosshair to show cursor's current position.	**Ctrl** ***Keypad*** **+**
Magnifies cursor in an OSx window for the duration of the window. If you close the window and reopen it, the cursor reverts to its normal size.	**Ctrl** ***Keypad*** **-**

B.5 Engineering Keyboard Navigation

Table B-4 depicts key combinations for the Engineering Keyboard which correspond to the Operator Keyboard functions. The keys below are in addition to those key functions that directly correspond to the Operator Keyboard. For example, pressing the **A** key on the Engineering Keyboard produces the same result as pressing the **A** key on the Operator Keyboard.

Table B-4 Engineering Keyboard Definitions

Engineering Keyboard Functions	Key Combinations
Accesses online help for window currently in focus.	**F1**
Silences alarm.	**Ctrl Shift s**
Acknowledges alarm.	**Ctrl Shift a**
Accesses alarm group.	**Ctrl Shift i**
Accesses action request.	**Ctrl Shift r**
Accesses real time trend.	**Ctrl Shift F1**
Accesses graphic.	**Ctrl Shift F2**
Accesses tag group.	**Ctrl Shift F3**
Accesses batch.	**Ctrl Shift F4**
Accesses reports.	**Ctrl Shift F5**
Accesses historical trend.	**Ctrl Shift F6**
Accesses tag detail.	**Ctrl Shift F7**
Accesses alarm group.	**Ctrl Shift F8**
Accesses recipe.	**Ctrl Shift F9**
Returns to previous display.	**Ctrl Shift F12**
Overview.	**Ctrl Shift *Keypad* End**
Page home.	**Ctrl Shift *Keypad* Home**
Page up.	**Ctrl Shift *Keypad*** ↑
Page down.	**Ctrl Shift *Keypad*** ↓
Page left.	**Ctrl Shift *Keypad*** ←
Page right.	**Ctrl Shift *Keypad*** →

B.6 Using Function Keys to Display Graphics

You can configure each of the keyboard function keys **F1** through **F12** to display a different graphic when that function key is pressed. Only graphics can be associated with function keys; trends, tag details, reports, and alarm screens do not support function key association.

Since function keys are also required by OSx configuration utilities, such as the Report Editor, you can enable or disable this feature as needed. OSx is shipped with this feature disabled.

You can enable or disable function key associations on a per terminal basis. If X terminals are connected to the OSx station, each X terminal, as well as the OSx station, can individually enable or disable the feature. This feature is not supported for remote terminals.

To enable function key associations, follow the steps below:

1. Place the cursor in the unoccupied gray area below the alarm icon in the lower right corner of the OSx screen, and click the left mouse button to access the root menu.
2. Click **Enable F Keys** on the root menu.
3. Once the feature is enabled, you must restart the Window Manager for the change to be recognized by OSx. Bring up the root menu again (as in step 1) and click **Restart**.
4. Click **OK** to the Restart prompt. You can now associate function keys with individual graphics and bring up these graphics in the Operate state with a single keystroke.

You can disable function key associations through a similar procedure. Once you have enabled the function keys, the root menu displays the option **Disable F Keys**. Select this option and restart the Windows Manager for the change to take effect.

Using Function Keys to Display Graphics (continued)

To associate a function key with a particular graphic, enable the function key associations as described on page B-7, and then follow these steps:

1. Set OSx to the Operate state, and display the graphic that you want to associate with a function key.
2. Press **Ctrl** and the function key simultaneously. For example, press **Ctrl F1** to assign the **F1** key to the currently open graphic.
3. Restart the Window Manager by selecting **Restart** from the root menu, and clicking **OK** to the Restart prompt.

Now when you press the function key in the Operate state, the graphic associated with that function key is displayed.

When you enable function key associations, the auto repeat function is disabled to prevent problems with multiple keystrokes, such as accidentally defining multiple associations or displaying a graphic multiple times if you hold down a function key for too long.

When function key associations are enabled, you can also display the previous screen by clicking the middle mouse or track ball button. This feature requires no additional configuration. However, when the function key associations are enabled, you cannot zoom within a trend display by pressing the middle mouse button.

Using Reflection X

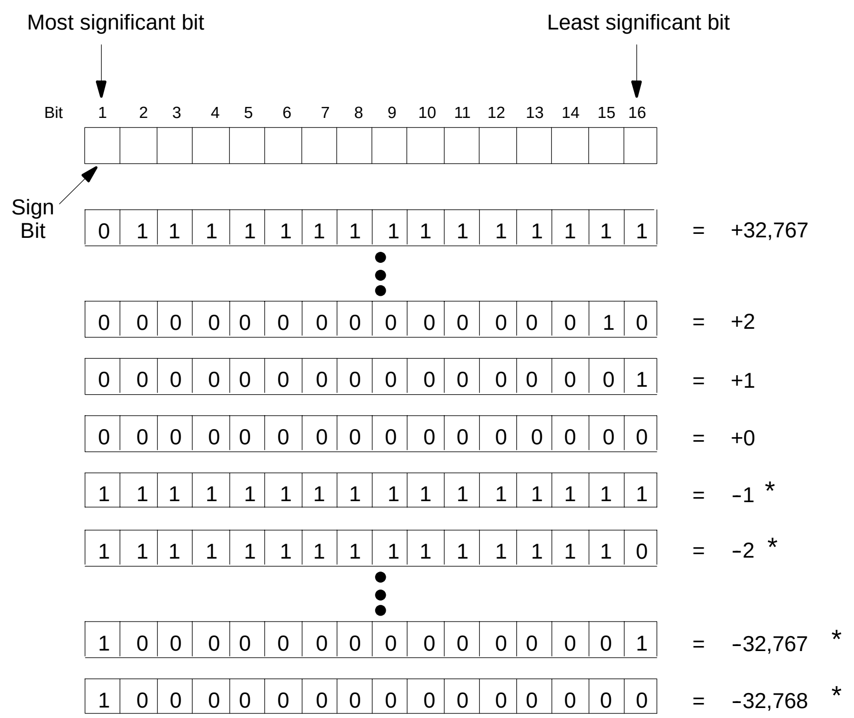
When you use Reflection X, press the **Shift** key instead of the **Ctrl** key to associate a function key with a particular graphic. Press **Shift** and the function key simultaneously. For example, press **Shift-F1** to assign the **F1** key to the currently displayed graphic. Press **Ctrl-Alt-Esc** and click **OK** to restart the Window Manager.

Appendix C

Number Base Conversions

C.1 Integer Format

The controller stores signed integers as 16-bit words in the format shown in Figure C-1. The 16-bit format allows you to store values ranging from –32,768 to +32,767 (decimal integer values). When the most significant bit contains a one, the value is a negative number. A zero in the most significant bit indicates a positive number. Negative numbers are stored in two's complement format.



* By the two's complement, if the sign bit is a 1, the remaining bits are added and then 1 is added.

Figure C-1 Format of Signed Integers

C.2 Binary Form of a Negative Number

To determine the binary equivalent of a negative number, follow these steps. See Figure C-2 for an example.

1. Convert the number to its positive binary equivalent.
2. Take the one's complement of the binary form of the number. (Invert the value of every bit: 1 for 0, and 0 for 1.)
3. Add one.

To enter −2, follow these steps.

1. Convert +2 to its binary equivalent.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	=	+2

2. Take the 1's complement of the result.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	=	1's complement of +2

3. Add 1.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	=	−2

Figure C-2 Determining Binary Form of a Negative Number

C.3 Floating Point Number Format

OSx uses the I.E.E.E. floating point standard when converting controller values into floating point numbers. Figure C-3 shows the binary format.

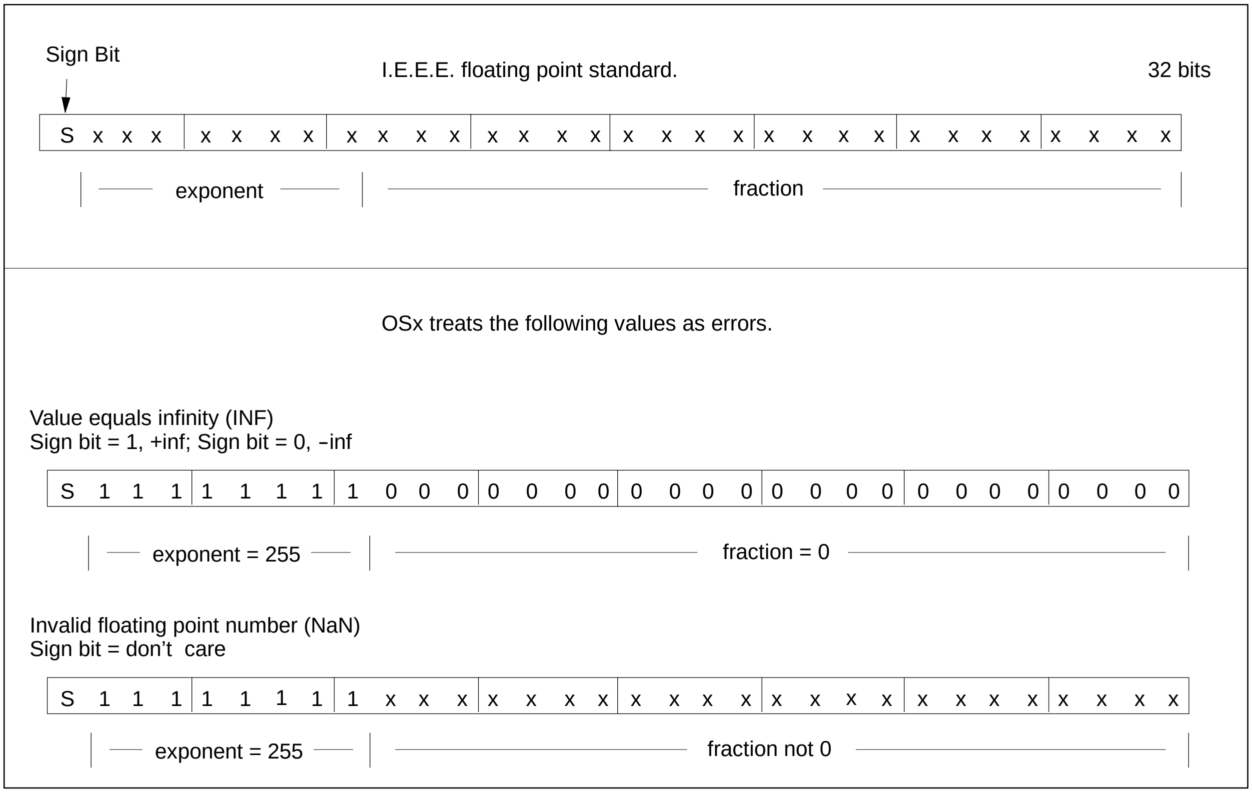


Figure C-3 Binary Format for Converting Floating Point Numbers

---

NOTE: When OSx encounters one of the error values shown at the bottom of Figure C-3, an error message is generated. The error message identifies the tag that contains the invalid value, and states, **INF | NAN, DB_FLOAT_NULL placed in database.** The invalid value is overwritten with the value of DB_FLOAT_NULL, which is –1e+38. Typically, this error occurs when the tags you have installed (for example, by compiling APT) do not match the memory locations that were expected by the program in the controller.

C.4 Hex to Decimal Conversion

To determine the decimal equivalent of a four-digit signed hexadecimal number, follow the steps listed in Figure C-4. The digits A, B, C, D, E, and F in hexadecimal are 10, 11, 12, 13, 14, and 15 in decimal. You can also use the bc utility, described in Section C.5.

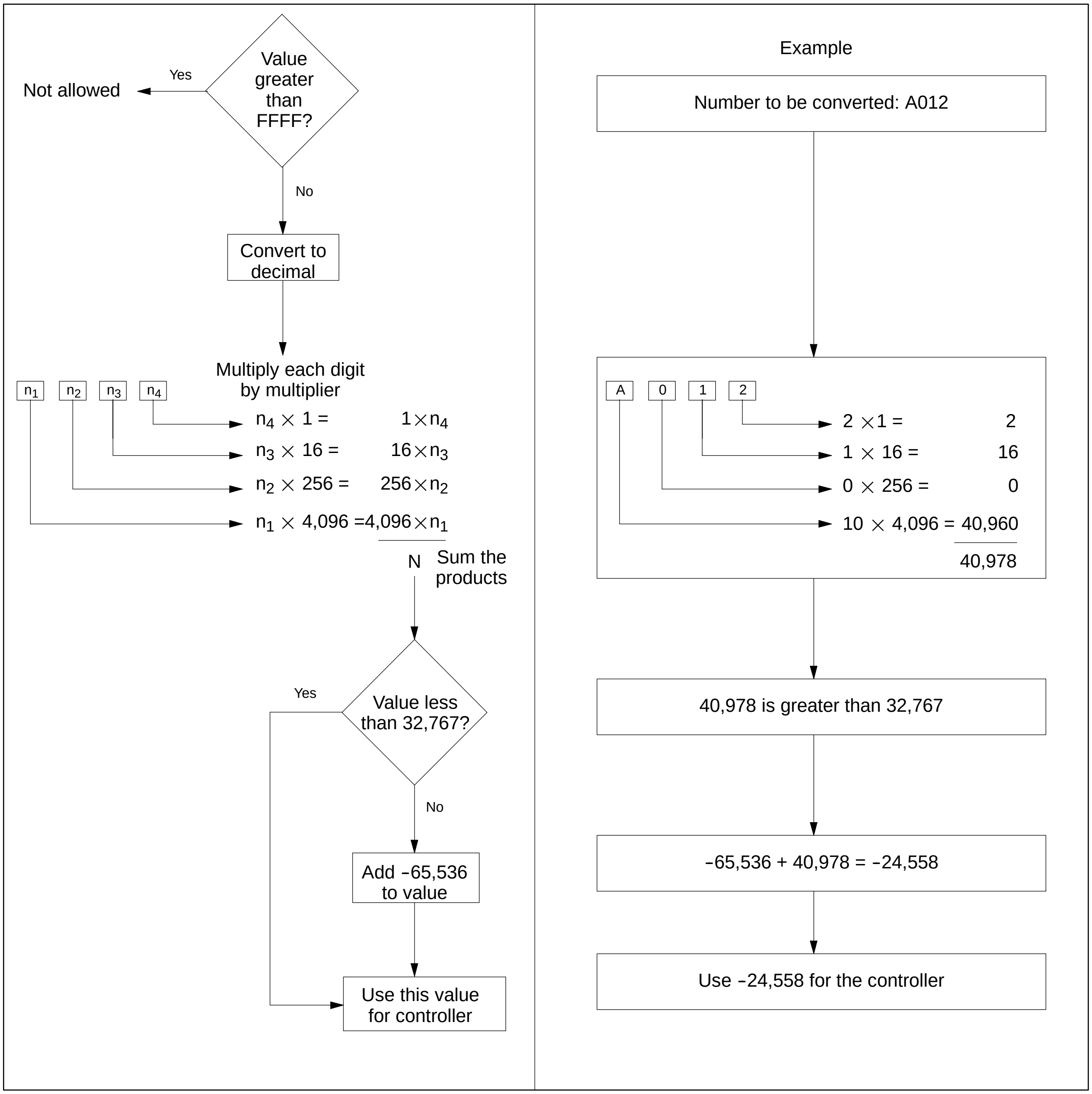


Figure C-4 Hexadecimal to Decimal Conversion

C.5 Decimal to Hex Conversion

You can use the bc conversion utility that comes with your system to convert numbers from decimal to hexadecimal. Follow the steps below.

1. Open an Xterm window and log in as `tistar`.
2. Type `bc` and press **Enter**.
3. Type `obase=16` and press **Enter**. This sets the number base for the output to hexadecimal.
4. Type in the numbers that you want converted, separated by semicolons. Each decimal number is returned as hexadecimal.

   Example: `22;14;43` is returned as **16**, **E**, and **2B**.
5. To end the program, type `quit` and press **Enter**.

You can also use the bc utility to convert from hex to decimal. In step 3, type `ibase=16.` All numbers that you input are converted to decimal. You must use upper case for the hexadecimal numbers A, B, C, D, E, and F. For more information, type `man bc` from the command line.

Appendix D

Controller Memory Types

D.1 Memory Types for Series 505 Controllers

Table D-1 and Table D-2 list and describe the digital and analog memory types available in the Series 505 controllers, the class of information each memory type contains, and how the data is stored.

Table D-1 Digital Memory Types (Series 505 Controllers)

Controller Memory	Operation	Number of Locations	Controller Storage	Database Storage U=used for data storage x= don't care
X *	Read	1 ⋮ 13	1 bit ⋮ 13 seq. bits	U x x x x x x x x x x x x m k a ⋮ U U U U U U U U U U U U U m k a
Y *	Read	1 ⋮ 13	1 bit ⋮ 13 seq. bits	U x x x x x x x x x x x x m k a ⋮ U U U U U U U U U U U U U m k a
	Write	1 ⋮ 16	1 bit ⋮ 16 seq. bits	U x x x x x x x x x x x x x x x ⋮ U U U U U U U U U U U U U U U U
C	Read	1 ⋮ 13	1 bit ⋮ 13 seq. bits	U x x x x x x x x x x x x m k a ⋮ U U U U U U U U U U U U U m k a
	Write	1	1 bit	U x x x x x x x x x x x x x x x
	Write	2	2 seq. bits	U U x x x x x x x x x x x x x x
	Write	1 ⋮ 16	1 bit ⋮ 16 seq. bits	U x x x x x x x x x x x x x x x ⋮ U U U U U U U U U U U U U U U U

* For the Series 505 controllers, X and Y memory have been combined to X/Y. The X/Y memory represents the same memory, with the only distinction being made by the actual module in the I/O base.

OSx reserves the three least significant bits for the following data:

- **m**—Manual Set — 1 = can manually trend tag data
0 = cannot manually trend tag data
- **k**—Alarm UnAcknowledge — 1 = alarm not acknowledged *and* alarm condition still exists
0 = alarm acknowledged
- **a**—Active — 1 = point being scanned
0 = point not being scanned

Table D-2 Analog Memory Types (Series 505 Controllers)

Memory Type	Information Class	Data Storage Format in Controller
WX Word Input	Location mapped to an analog input (AI), word input (WI), or special function module Read only Integer	16 bits
WY Word Output	Location mapped to an analog output (AO), word output (WO), or special function module Read or Write Integer	16 bits
V-Memory (Variable Memory)	General purpose memory Read or Write Integer	16 bits
K-Memory (Constant Memory)	General purpose memory Read or Write Integer	16 bits
G-Memory* (Global Memory)	General purpose memory Read or Write Integer	16 bits
V.-Memory (Real Number format)	Read or Write Floating Point Number	32 bits
K.-Memory (Real Number format)	Read or Write Floating Point Number	32 bits
G.-Memory* (Real Number format)	Read or Write Floating Point Number	32 bits

* SIMATIC 575 controllers only.

The SIMATIC 575 controller has global memory available for tag assignments. When you assign tag attributes to memory locations in the range G1–G32768, then the assignment is made to local G-Memory, the global memory in the 575 CPU in which the CP1434 is located. Assignment to one of the specific 32 Kword global memory areas, such as GA, GB, GC, etc., is also possible. If you install tags through APT, the G-Memory address is automatically provided. If you use a spreadsheet or edit the **install.tag** file to assign tags, you need to add 32768 for each global memory area. For example, install a tag for memory location GA1 as GA32769 (32768 + 1); install a tag for memory location GB1 as GB65537 (32768 + 32768 + 1).

Table D-3, Table D-4, and Table D-5 list the variables used by Series 505 controllers for loops, analog alarms, timers and counters.

Table D-3 Loop Memory Types (Series 505 Controllers)

Controller Memory	Database Attribute	Data Storage Format in Controller
LSTATUS	STATUS	16 bits
LPVL	L_RANGE	32 bits
LPVH	H_RANGE	32 bits
LPV	PV	32 bits
LMN	OUT	32 bits
LSP	SP	32 bits
LMODE	MODE	32 bits
LKC	GAIN	32 bits
LTD	RATE	32 bits
LTI	RESET	32 bits
LHHA	HH_ALARM	32 bits
LHA	H_ALARM	32 bits
LLA	L_ALARM	32 bits
LLLA	LL_ALARM	32 bits
LODA	H_DEV (ORANGE)	32 bits
LYDA	L_DEV (YELLOW)	32 bits
LRCA	ROC_ALARM	32 bits

Table D-4 Analog Alarm Memory Types (Series 505 Controllers)

Controller Memory	Database Attribute	Data Storage Format in Controller
AVF	STATUS	16 bits
APVL	L_RANGE	32 bits
APVH	H_RANGE	32 bits
APV	PV	32 bits
ASP	TARGET	32 bits
AHHA	HH_ALARM	32 bits
AHA	H_ALARM	32 bits
ALA	L_ALARM	32 bits
ALLA	LL_ALARM	32 bits
AODA	H_DEV (ORANGE)	32 bits
AYDA	L_DEV (YELLOW)	32 bits
ARCA	ROC_ALARM	32 bits

Table D-5 Timer/Counter Memory Types (Series 505 Controllers)

Controller Memory	Database Attribute	Data Storage Format in Controller
TCP	PRESET	16 bits
TCC	CURRENT	16 bits

D.2 Memory Types for S5 Controllers

Table D-6 and Table D-7 list and describe the memory types available in the S5 controllers, the class of information each memory type may contain, and how the data is stored.

Table D-6 Digital Memory Types (S5 Controllers)

Controller Memory	Operation	Number of Locations	Controller Storage	Database Storage U=used for data storage x= don't care
Q Output image	Read Only	1 . . . 13	1 bit . . . 13 seq. bits	U x x x x x x x x x x x x m k a . . . U U U U U U U U U U U U U m k a
I Input image	Read Only	1 . . . 13	1 bit . . . 13 seq. bits	U x x x x x x x x x x x x m k a . . . U U U U U U U U U U U U U m k a
F Flag area	Read Only	1 . . . 13	1 bit . . . 13 seq. bits	U x x x x x x x x x x x x m k a . . . U U U U U U U U U U U U U m k a

Table D-7 Other Memory Types (S5 Controllers)

Controller Memory	Database Attribute	Operation	Controller Storage
DBDW	Data Block Data Word	General Purpose Read/Write Integer	16 bits
DBDD	Data Block Double Data Word	General Purpose Read/Write Real	32 bits
DXDW	Extended Data Block Data Word	General Purpose Read/Write Integer	16 bits
DXDD	Extended Data Block Double Data Word	General Purpose Read/Write Real	32 bits
CC	Counter Current	Read Only Integer	16 bits
TC	Timer Current	Read Only Integer	16 bits

D.3 Memory Types for Tag Attributes

In the **install.tag** file, which contains all tag information (Chapter 5), the memory locations are typically specified by a numeric memory location appended to a memory name. For example, **V100** defines V-memory word 100. However, for the case of V, K, or G memory used as a floating point value, the memory name is **V.**, **K.**, or **G.**. For example, **V.100** defines a floating point value located in V-memory words 100 and 101.

For the case of S5 memory names DBDW, DBDD, DXDW, and DXDD, the memory name defines two values: a data block number, and a word number within the data block. The required syntax is **DBx:DWy**, where x is the data block number and y is the word number. For example, **DB100:DW10** defines data word 10 in data block 100. The S5 memory names Q, I, and F, require specification in the format **x.y** where x is byte and y is a bit, e.g., F0.7.

Table D-8 lists the memory type assignments for Series 505 and S5 controllers by tag attribute.

Table D-8 Tag Attribute Memory Types

Tag Type	Attribute	Series 505 Memory Type Assignment	S5 Memory Type Assignment
AI	STATUS	K, V, G, X/Y, C, AVF	DXDW, DBDW
	L_RANGE	K., V., G., APVL	DXDD, DBDD
	H_RANGE	K., V., G., APVH	DXDD, DBDD
	PV	K, V, G, WX/WY, K., V., G., APV	DXDD, DBDD
	TARGET	K, V, G, K., V., G., ASP	DXDD, DBDD
	HH_ALARM	K, V, G, K., V., G., AHHA	DXDD, DBDD
	H_ALARM	K, V, G, K., V., G., AHA	DXDD, DBDD
	L_ALARM	K, V, G, K., V., G., ALA	DXDD, DBDD
	LL_ALARM	K, V, G, K., V., G., ALLA	DXDD, DBDD
	H_DEV (ORANGE)	K, V, G, K., V., G., AODA	DXDD, DBDD
	L_DEV (YELLOW)	K, V, G, K., V., G., AYDA	DXDD, DBDD
	ROC_ALARM	K, V, G, K., V., G., ARCA	DXDD, DBDD
AO	STATUS	K, V, G, X/Y, C	DXDW, DBDW
	L_RANGE	K., V., G.	DXDD, DBDD
	H_RANGE	K., V., G.	DXDD, DBDD
	OUT	K, V, G, WX/WY, K., V., G.	DXDD, DBDD
	MODE	K, V, G, X/Y, C	DXDW, DBDW
DI	STATUS	K, V, G, X/Y, C	Q, I, F, DXDW, DBDW

Table continued on next page.

Memory Types for Tag Attributes (continued)

Table D-8 Tag Attribute Memory Types (continued)

Tag Type	Attribute	Series 505 Memory Type Assignment	S5 Memory Type Assignment
DO	STATUS	K, V, G, X/Y, C	Q, I, F, DXDW, DBDW
	COMMAND	X/Y, C	DXDW, DBDW
LOOP	STATUS	K, V, G, X/Y, C, LSTATUS	DXDW, DBDW
	L_RANGE	K., V., G., LPVL	DXDD, DBDD
	H_RANGE	K., V., G., LPVH	DXDD, DBDD
	PV	V., G., LPV	DXDD, DBDD
	SP	V., G., LSP	DXDD, DBDD
	OUT	V., G., LMN	DXDD, DBDD
	MODE	V, G, X/Y, C, LMODE	DXDW, DBDW
	GAIN	V., G., LKC	DXDD, DBDD
	RATE	V., G., LTD	DXDD, DBDD
	RESET	V., G., LTI	DXDD, DBDD
	HH_ALARM	V., G., LHHA	DXDD, DBDD
	H_ALARM	V., G., LHA	DXDD, DBDD
	L_ALARM	V., G., LLA	DXDD, DBDD
	LL_ALARM	V., G., LLLA	DXDD, DBDD
	H_DEV (ORANGE)	V., G., LODA	DXDD, DBDD
	L_DEV (YELLOW)	V., G., LYDA	DXDD, DBDD
	ROC_ALARM	V., G., LRCA	DXDD, DBDD
TMR	STATUS	X/Y, C	DXDW, DBDW
	L_RANGE	K., V., G.	DXDD, DBDD
	H_RANGE	K., V., G.	DXDD, DBDD
	PRESET	K, V, G, K., V., G., TCP	DXDW, DBDW, DXDD, DBDD
	CURRENT	K, V, G, K., V., G., TCC	DXDW, DBDW, DXDD, DBDD, TC
CTR	STATUS	X/Y, C	DXDW, DBDW
	PRESET	K, V, G, TCP	DXDW, DBDW
	CURRENT	K, V, G, TCC	DXDW, DBDW, CC

Table continued on next page.

Table D-8   Tag Attribute Memory Types (continued)

Tag Type	Attribute	Series 505 Memory Type Assignment	S5 Memory Type Assignment
CALC	STATUS	K, V, G, X/Y, C	DXDW, DBDW
	L_RANGE	K., V., G.	DXDD, DBDD
	H_RANGE	K., V., G.	DXDD, DBDD
	VALUE	K, V, G, WX/WY, K., V., G.	DXDD, DBDD
IVAR	STATUS	K, V, G, X/Y, C, STW	DXDW, DBDW
	VALUE	STW, K, V, G, WX/WY	DXDW, DBDW
DEVICE	STATUS	V, G, X/Y, C	DXDW, DBDW
	OVERRIDE	X/Y, C	DXDW, DBDW
	MODE_CMD	X/Y, C	DXDW, DBDW
	SETPOINT	X/Y, C	DXDW, DBDW
	TIMEOUT_1	TCP	DXDW, DBDW
	TIMEOUT_2	TCP	DXDW, DBDW
DI10	STATUS	K, V, G, X/Y, C	DXDW, DBDW
DO10	STATUS	K, V, G, X/Y, C	DXDW, DBDW
	DATA_VAL_1	X/Y, C	DXDW, DBDW
	DATA_VAL_2	X/Y, C	DXDW, DBDW
	DATA_VAL_3	X/Y, C	DXDW, DBDW
	DATA_VAL_4	X/Y, C	DXDW, DBDW
	DATA_VAL_5	X/Y, C	DXDW, DBDW
	DATA_VAL_6	X/Y, C	DXDW, DBDW
	DATA_VAL_7	X/Y, C	DXDW, DBDW
	DATA_VAL_8	X/Y, C	DXDW, DBDW
	DATA_VAL_9	X/Y, C	DXDW, DBDW
	DATA_VAL_10	X/Y, C	DXDW, DBDW
TEXT	STATUS	V, G, X/Y, C	DXDW, DBDW
	TEXT_1	V, G	DXDW, DBDW
	TEXT_2	V, G	DXDW, DBDW
	TEXT_3	V, G	DXDW, DBDW

Table continued on next page.

Memory Types for Tag Attributes (continued)

Table D-8 Tag Attribute Memory Types (continued)

Tag Type	Attribute	Series 505 Memory Type Assignment	S5 Memory Type Assignment
UNIT	STATUS	V, G, X/Y, C	DBDW
	COMMAND	V, G, X/Y, C	DBDW
	MODE_CMD	V, G, X/Y, C	DBDW
	OPERATION	V, G	DBDW
	BCH_REQ	V, G, X/Y, C	DBDW
	BCH_REQ_INFO	V, G	DBDW
	BCH_REQ_RESP	V, G, X/Y, C	DBDW
RECIPE_ AREA	SCALE_FACTOR	K., V., G., ASP	DXDD, DBDD
	PLC_RCP_REQ	K, V, G, TCP	DXDW, DBDW
	HOLD_REQ	K, V, G, X/Y, C	DXDW, DBDW

Index

B

E

F

S

T

U

V

W

X

Z

Customer Response

We would like to know what you think about our user manuals so that we can serve you better. How would you rate the quality of our manuals?

	Excellent	Good	Fair	Poor
Accuracy	_______	_______	_______	_______
Organization	_______	_______	_______	_______
Clarity	_______	_______	_______	_______
Completeness	_______	_______	_______	_______
Graphics	_______	_______	_______	_______
Examples	_______	_______	_______	_______
Overall design	_______	_______	_______	_______
Size	_______	_______	_______	_______
Index	_______	_______	_______	_______

Would you be interested in giving us more detailed comments about our manuals?

☐ **Yes!** Please send me a questionnaire.

☐ **No.** Thanks anyway.

Your Name: ______________________________

Title: ______________________________

Telephone Number: (_____) ______________________________

Company Name: ______________________________

Company Address: ______________________________

____

____

Manual Name: SIMATIC PCS 7 OSx 4.1.2 Process Configuration Manual **Edition:** Original
Manual Assembly Number: 2810913-0001 **Date:** 7/02
Order Number: 6ES7 6550XX058BC2

FOLD

SIEMENS ENERGY & AUTOMATION INC
3000 BILL GARLAND ROAD
P O BOX 1255
JOHNSON CITY TN 37605–1255

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.3 JOHNSON CITY, TN

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN TECHNICAL COMMUNICATIONS M/S 519
SIEMENS ENERGY & AUTOMATION INC
P O BOX 1255
JOHNSON CITY TN 37605–1255

FOLD