Tivoli

Tivoli for AS/400 Endpoints User’s Guide

Version 3.6.2

Tivoli for AS/400 Endpoints User's Guide (December 1999)

Copyright Notice

Trademarks

Notices

References in this publication to Tivoli Systems or IBM products, programs, or services do not imply that they will be available in all countries in which Tivoli Systems or IBM operates. Any reference to these products, programs, or services is not intended to imply that only Tivoli Systems or IBM products, programs, or services can be used. Subject to Tivoli System's or IBM's valid intellectual property or other legally protectable right, any functionally equivalent product, program, or service can be used instead of the referenced product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Tivoli Systems or IBM, are the responsibility of the user.

Tivoli Systems or IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, New York 10504-1785, U.S.A.

Contents

Chapter 1—Overview and Installation

Chapter 2—Tivoli Management Framework for AS/400 Endpoints

Chapter 3—Tivoli User Administration for AS/400 Endpoints

Chapter 4—Tivoli Security Management for AS/400 Endpoints

Chapter 5—Tivoli Distributed Monitoring for AS/400 Endpoints

Chapter 6—Tivoli Inventory for AS/400 Endpoints

Chapter 7—Tivoli Software Distribution for AS/400 Endpoints

Appendix A—Commands

Preface

Tivoli Management Environment for AS/400 provides support for AS/400 endpoints in the Tivoli Management Region (TMR). This guide explains the concepts and procedures necessary for you to effectively use Tivoli for AS/400 to manage AS/400 systems running the Tivoli endpoint service.

This guide is designed to supplement other Tivoli documentation about configuring and using the Tivoli Management Framework and applications. Tivoli has integrated support for AS/400 systems in the Tivoli application suite, and generally you will need to use procedures defined in Tivoli documentation. This guide provides all necessary AS/400-specific information and procedures to configure and use Tivoli on AS/400 endpoints.

Who Should Read This Guide

The target audience for this guide is Tivoli network administrators performing daily Tivoli operations on AS/400 systems. Users of the guide should have some knowledge of the following:

- AS/400 operating systems
- The Tivoli Management Framework and Tivoli applications that provide AS/400 support
- Operating systems running on the server to which the AS/400 endpoints are connected

Related Documents

You must understand the information in the following manuals before attempting to use this guide. Detailed information on how to install and use each specific Tivoli product is included in the user documentation that is provided with that product. See the 3.6 version of the following documentation for more product-specific information:

- *Tivoli Software Installation Service User’s Guide* and *Tivoli Software Installation Service Release Notes*

Provide installation information and describe how to add Tivoli products for multiple installations as well as how to configure large-scale installations.

- *Tivoli Management Framework Planning and Installation Guide*, *Tivoli Management Framework User's Guide*, *Tivoli Management Framework Reference Manual*, and *Tivoli Management Framework Release Notes*

  Provide detailed conceptual and procedural information about planning, configuring, maintaining, and managing Tivoli installations from the Tivoli desktop of the command line. Information about how to apply patches is also included in this manual.

- *Tivoli User and Group Management Guide* and *Tivoli User Group and Management Guide Release Notes*

  Provide information on user and group accounts, examples of setting up and using Tivoli and User Group Management, and procedures for using all the components of the user interface, including the graphical user interface (GUI) and examples using the command line interface (CLI).

- *Tivoli Security Management User's Guide* and *Tivoli Security Management Release Notes*

  Provide conceptual and procedural information about managing security profiles and records, and about using all components of the user interface from the Tivoli desktop and the command line interface (CLI).These manuals also provide information about installing the Tivoli Security Management product.

- *Tivoli Distributed Monitoring User's Guide* and *Tivoli Distributed Monitoring Collection Reference*

  Provide task-oriented information on how to use Tivoli Distributed Monitoring to manage system and application resources. These manuals also include background and procedural information on installing and using monitoring collections and monitoring sources, examples of setting up and using monitors, and procedures for using all components of the user interface, including the graphical user interface (GUI) and examples using the command line interface (CLI).

- *Tivoli Inventory User's Guide* and *Tivoli Inventory Release Notes*

  Provide information about installing and configuring Tivoli Inventory, as well as for setting up and gathering system information, and how to create and use SQL queries to populate the configuration repository.

- *Tivoli Software Distribution User's Guide*, *Tivoli Software Distribution Reference Manual*, and *Tivoli Software Distribution Release Notes*

  Provide concepts and procedures necessary to effectively install and use Tivoli Software Distribution from the Tivoli desktop and from the command line interface to distribute software over networks. These manuals also provide information about file package definitions, commands, policies, configuration programs, and about troubleshooting TME10 Software Distribution.

- *Tivoli Enterprise Console User's Guide*, *Tivoli Enterprise Console Adapters Guide*, *Tivoli Enterprise Console Rule Builder's Guide*, *Tivoli Enterprise Console Reference Manual*, and *Tivoli Event Integration Facility User's Guide*

  Provide detailed information and procedures for installing and maintaining Tivoli Enterprise Console software.

- AS/400 library of manuals available at **www.as400.ibm.com**. The AS/400 reference library provides searchable online guides, AS/400 manuals and Redbooks, and other information about configuring and using AS/400 systems.

What this Guide Contains

The *Tivoli for AS/400 Endpoints User's Guide* contains the following sections:

- Chapter 1, Overview and Installation

  Provides an introduction to Tivoli for AS/400 Endpoints by Tivoli application support for AS/400 endpoints. This chapter also provides installation information for installing and configuring

AS/400 endpoints in a Tivoli environment. describing file packages, distribution programs.

- Chapter 2, Tivoli Management Framework for AS/400 Endpoints

  Provides procedures and commands to configure and use AS/400 endpoints in the Tivoli environment. This chapter also describes how to use the AS/400 task library to automate tasks that are performed frequently by an AS/400 operator.

- Chapter 3, Tivoli User Administration for AS/400 Endpoints

  Describes how to configure and use Tivoli User Administration to create AS/400 user profiles and user accounts.

- Chapter 4, Tivoli Security Management for AS/400 Endpoints

  Describes how to configure and use Tivoli Security Management to protect a wide variety of network resources, including files, directories, terminals, OS/400 objects, and protected network processes.b

- Chapter 5, Tivoli Distributed Monitoring for AS/400 Endpoints

  Describes how to configure and use Tivoli Distributed Monitoring to manage the performance of AS/400 systems with AS/400 monitoring collections.

- Chapter 6, Tivoli Inventory for AS/400 Endpoints

  Describes how to configure and use Tivoli Inventory to scan inventory information on AS/400 machines and to use this data to manage AS/400 endpoints in the Tivoli environment.

- Chapter 7, Tivoli Software Distribution for AS/400 Endpoints

  Describes how to configure and use Tivoli Software Distribution to create and distribute file packages to AS/400 endpoints in the Tivoli environment.

- Appendix A, Commands

  Lists Tivoli commands that are specifically pertinent to managing operations on AS/400 endpoints. Syntax and descriptions of command functions are provided for each command.

Conventions Used in this Guide

The guide uses several typeface conventions for special terms and actions. These conventions have the following meaning:

Bold	Commands, keywords, file names, authorization roles, URLs, names of windows and dialogs, other controls, or other information that you must use literally are in **bold**.
Italics	Variables and values that you must provide as well as words and phrases that are emphasized are in *italics*.
Bold Italics	New terms that are defined in text are in ***bold italics***.
`Monospace`	Code examples, output, and system messages are in a `monospace` font.

Platform-specific Information

The following table identifies text used to indicate platform-specific information or procedures.

Platform	Supported Versions
AS/400	Endpoint: IBM AS/400 running OS/400, Versions V3R2, V3R7, V4R1, and V4R2

The AS/400 endpoint can connect to any TMR server or gateway whose platform is supported by Tivoli. See the *Tivoli Management Framework Planning and Installation Guide* for platform-specific information concerning the TMR server or gateways.

Revision History

The following table lists the revision history for the *Tivoli for AS/400 Endpoints User’s Guide*:

Form Number	Date	Changes to Documentation
GC32-0279-00	11/98	Original Version 3.6 release.
GC32-0279-00	4/99	Updated Version 3.6 release shipped with the 3.6.1 CD-ROM. This release included minor technical and formatting changes. This release was available only from the Tivoli Support Web site.
GC32-0279-01	12/99	Version 3.6.1 release. This release includes minor technical and formatting changes since the previous release.

Contacting Customer Support

If you encounter difficulties with any Tivoli products, you can enter **http://www.support.tivoli.com** to view the Tivoli Support home page. After you link to and submit the customer registration form, you will be able to access many customer support services on the Web.

Use the following phone numbers to contact customer support in the United States: the Tivoli number is 1-800-848-6548 (1-800-TIVOLI8) and the IBM number is 1-800-237-5511 (press or say 8 after you reach this number). Both of these numbers direct your call to the Tivoli Customer Support Call Center.

We are very interested in hearing from you about your experience with Tivoli products and documentation. We welcome your suggestions for improvements. If you have comments or suggestions about this documentation, please send e-mail to **pubs@tivoli.com**.

1

Overview and Installation

The Tivoli Enterprise software enables systems administrators to manage heterogeneous distributed computer systems. With the Version 3.6 release, the Tivoli Management Framework and select Tivoli products support AS/400 workstations as endpoints in a ***Tivoli Management Region*** (***TMR***). In this capacity, AS/400 systems can run the Tivoli endpoint service (or daemon) to receive profile distributions and execute tasks.

The Tivoli environment consists of one or more TMRs that are installed within a network. A TMR consists of a TMR server, endpoint gateways, and a set of directly associated endpoints. The TMR server manages information within the TMR and includes an endpoint manager, which maintains a list of all endpoints in the TMR and controls the assignment of endpoints to endpoint gateways. This documents assumes you have fully implemented the Tivoli environment. For more information about configuring and using Tivoli in the distributed network environment, refer to the *Tivoli Management Framework Planning and Installation Guide*.

Tivoli for AS/400 enables you to perform the following operations on an AS/400 endpoint:

- Scan all AS/400 systems in a policy region for a list of installed software.
- Identify systems running earlier releases of software.
- Automatically distribute, install, and configure upgraded software on the appropriate machines.

- Monitor system resources and performance of the AS/400 systems in the network.
- Manage AS/400 user profiles and AS/400 system security for all of your AS/400 system from a central source.
- Monitor events on AS/400 systems and forward selected events to a Tivoli Enterprise Console event server for processing.

The following Tivoli products can be used to manage AS/400 endpoints beginning with the Version 3.6 release:

- Tivoli Software Installation Service provides a graphical user interface to install AS/400 endpoints.
- Tivoli Management Framework enables AS/400 systems to be supported as an endpoint and creates an AS/400 Task Library. The task library contains common AS/400 Control Language (CL) commands that can be used by Tivoli network administrators to control daily system operations. These CL commands are helpful in automating tasks when used in conjunction with the Tivoli Enterprise Console.
- Tivoli Inventory can be used to collect information about the AS/400 hardware and software in your network. Tivoli Inventory provides a customized Management Information Format (MIF) to help manage your AS/400 Program Temporary Fixes (PTFs). To use the customized MIF, Inventory also provides an AS/400 PTF database schema that extends each database to accept the PTF data that is scanned.
- Tivoli Software Distribution can be used to distribute software to AS/400 systems in your network. In conjunction with Tivoli Inventory, you can use Tivoli Software Distribution to manage the deployment of new software products to all AS/400 systems. Using the Tivoli Object Packaging Utility for AS/400, you can create file packages for distribution to AS/400 endpoints.
- Tivoli Distributed Monitoring can be used to monitor critical system resources and performance of AS/400 systems in your network.
- Tivoli User Administration can be used to manage AS/400 user profiles across your entire network enterprise.

- Tivoli Security Management can be used to manage the security of the AS/400 machines in your network from a central source.
- Tivoli Enterprise Console (TEC) can be used to manage many different systems from one central console. Included with TEC are the AS/400 alert adapter and AS/400 message adapter that automate critical systems management functions in the TMR environment. These adapters monitor events on AS/400 systems and forward selected events to the TEC event server for processing. Depending on the outcome from event processing, automated responses can be issued using the task library feature.
- Tivoli Module Builder, Version 2.0, can be used to create modules for managing applications that run in a Tivoli environment. Using DTMF MIF standards, the Tivoli module builder helps application developers identify the information that is key to managing their applications in a complex network environment. The Tivoli Module Builder was updated to manage AS/400 applications. See the *Tivoli Module Builder User's Guide* for more information about using this product.

Installing AS/400 Endpoints

You can install AS/400 endpoints using either Tivoli Software Installation Service (SIS) or with the **w4inslcf.pl** script. Using either approach, you can specify support for languages other than English. Refer to "Installing Internationalization Support" on page 1-4 for additional details.

Installing the endpoint creates the **QTMELCF** library and the following user accounts:

- **QTIVROOT**
- **QTIVUSER**
- **QTIVOLI**

Installing with Tivoli Software Installation Service

Tivoli Software Installation Service (SIS) installs AS/400 endpoints using the FTP access method.

To install an AS/400 endpoint, select the correct interpreter type for the system on which you will install the AS/400 endpoint. Supported AS/400 interpreter types are as follows:

os400-v3r2	CISC models with OS/400 V3R2.
os400-v3r7	RISC models with OS/400 V3R7 or higher.

SIS provides a field for specifying additional languages. When specifying a language, use the AS/400 language code. For a list of AS/400 language codes, refer to “Installing Internationalization Support” on page 1-4.

Because SIS is designed to be used on managed nodes that support Java interface, you will need to use the **w4inslcf.pl** script to install AS/400 endpoint for operating systems currently not supported by SIS.

Installing with the w4inslcf.pl Script

The **w4inslcf.pl** script installs and optionally starts the AS/400 endpoint daemon on one or more AS/400 systems. The **w4inslcf.pl** script is installed on your endpoint gateway when you install the 3.6.1-TMF-0001 enablement patch. The location or path of this script is as follows:

```
$BINDIR/../lcf_bundle/generic/w4inslcf.pl
```

To install the AS/400 endpoint (AS/400 Licensed Program 1TMELCF) with this script, you *must* specify the path to the **/1tmelcf** directory.

Installing Internationalization Support

You can install the AS/400 endpoint code (AS/400 Licensed Program 1TMELCF) to use multiple languages. Although, Tivoli is translated into the following languages, the AS/400 system can be operating in other languages:

- Brazilian Portuguese (Code 2980)
- English Uppercase and Lowercase (Code 2924)
- English Uppercase DBCS (Code 2938)
- French (Code 2928)
- German (Code 2929)
- Italian (Code 2932)

- Japanese (Katakana) DBCS (Code 2962)
- Korean DBCS (Code 2986)
- Simplified Chinese DBCS (Code 2989)
- Spanish (Code 2931)

The default installation language is English. If you install the AS/400 endpoint code without specifying a language, English (2924) and uppercase English (2938) are installed.

When you install the AS/400 endpoint using Software Installation Service (SIS), you select the languages to install from a SIS dialog.

When you install the AS/400 endpoint using the installation script (**w4inslcf.pl**), you specify the languages using the **–N** argument. For example, to install French (2928) and German (2929), the command line would be as follows:

```
w4inslcf.pl -g gwlable[+port] -N ‘2928 2929’ endpoint
```

Where:

gwlable[+port]
The hostname or IP address and optionally the port number for the gateway to which the AS/400 endpoint will log in.

endpoint The host name or IP address of the AS/400 endpoint.

To use the OS/400 CL commands from the command line in a language other than your primary language, add the appropriate Tivoli language library to your library list. The language library format is **QTME***xxxx*, where *xxxx* is the 4-digit language code. For French, the language library is **QTME2928**.

For additional information on installing the endpoint code on an AS/400 system, refer to “Installing AS/400 Endpoints” on page 1-3. For general internationalization information and instructions on setting up a non-English TMR, refer to the Internationalization section of the Tivoli Management Framework release notes.

Object Packaging Utility for AS/400

Using the Tivoli Software Distribution, you can distribute and deploy software to AS/400 endpoints throughout the network. The Object

Packaging Utility for AS/400 creates AS/400-specific file packages to be used with Software Distribution.

Before installing the Object Packaging Utility for AS/400, complete the following steps:

1. Ensure that Client Access/400 is installed and configured on the system to be used to prepare AS/400 file packages.
2. Ensure that this system is connected to the AS/400 Software Distribution source host.
3. Install the Software Distribution for AS/400, Version 3.6.1 patch on the endpoint gateway managing the AS/400 endpoints.
4. Run the **packinst.sh** script to install support for the Object Packaging Utility for AS/400. This script is located in the **$BINDIR\..\lcf_bundle\generic** directory. The only argument for this script is the name of the AS/400 endpoint.

Installing the Tivoli Object Packaging Utility for AS/400

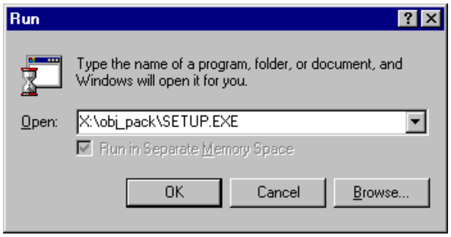
Using Tivoli Software Distribution, you can distribute and deploy software from a centralized point to AS/400 endpoints throughout the network. Prior to deploying this software, however, you must create a file package for the software you want to distribute. The Tivoli Object Packaging Utility for AS/400 creates AS/400-specific file packages to be used with Tivoli Software Distribution.

Note: Client Access/400 must be installed and configured on a Windows 95 or Windows NT system that you will be using to prepare AS/400 file packages. Ensure this system is connected to the AS/400 source system on which the software is stored.

To install the Tivoli Object Packaging Utility for AS/400 on any Windows 95 or Windows NT 4.0 system complete the following steps:

1. From the **File Manager** or **Program Manager** window, select **Run** from the **File** menu to display the **Run** dialog.

2. Enter the path to the **SETUP.EXE** file, which resides in the *x*:**\obj_pack** directory on the compact disc, in the **Open** field, as follows:



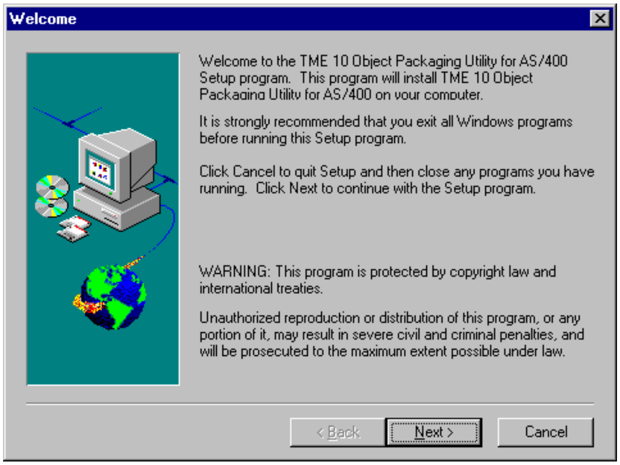
Type the name of a program, folder, or document, and Windows will open it for you.

Open: X:\obj_pack\SETUP.EXE

☑ Run in Separate Memory Space

where *x* is the CD-ROM drive where the Tivoli for AS/400 compact disc is inserted.

3. Click the **OK** button to run the setup program and display the **Welcome** dialog for the Tivoli Client Packaging Utility for AS/400.



Welcome to the TME 10 Object Packaging Utility for AS/400 Setup program. This program will install TME 10 Object Packaging Utility for AS/400 on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup program.

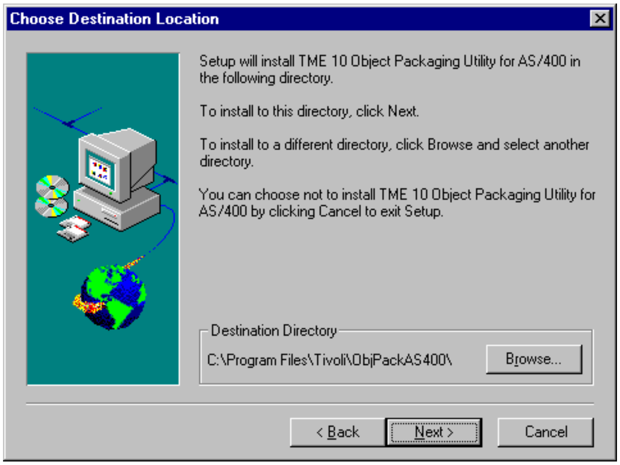
Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

The setup program leads you through the Tivoli Object Packaging Utility for AS/400 installation. Click the **Next** button to advance to the following page or click the **Back** button to return to the previous page. To stop the process at any time and return to your desktop session, click the **Cancel** button.

4. The following **Choose Destination Location** window is displayed:



Choose Destination Location

Setup will install TME 10 Object Packaging Utility for AS/400 in the following directory.

To install to this directory, click Next.

To install to a different directory, click Browse and select another directory.

You can choose not to install TME 10 Object Packaging Utility for AS/400 by clicking Cancel to exit Setup.

Destination Directory

C:\Program Files\Tivoli\ObjPackAS400\

The default location is automatically set to the **C:\Program Files\Tivoli\ObjPackAS400** directory. Click the **Browse** button to navigate through the directory structure and specify another directory if you want to set a different location.

Press the **Next** button to continue the installation.

5. To start the Object Packaging Utility, double-click on the **as4pack.exe** icon from the **ObjPackAS400** window or select the **Tivoli Object Packaging Utility for AS/400** option from the **Start** → **Programs** → **Tivoli** menu.

Uninstalling the Tivoli Object Packaging Utility

Perform the following steps to remove the Tivoli Object Packaging Utility for AS/400 from a Windows system:

1. From the Windows **Start** menu, select **Control Panel** from the **Settings** menu to display the **Control Panel** dialog.
2. Double-click on the **Add/Remove Programs** icon to display the **Add/Remove Programs Properties** dialog.

3. From this dialog, select TME 10 Object Packaging Utility for AS/400 from the scrolling list and click the **Add/Remove** button. The **Confirm File Deletion** dialog is displayed.
4. Confirm the removal operation by clicking the **Yes** button, or cancel the operation by clicking the **No** button. An uninstall shield removes the product from the system if you clicked **Yes**.
5. Click the **OK** button to return to the **Add/Remove Programs Properties** dialog.
6. Click the **OK** button on the **Add/Remove Programs Properties** dialog to complete the uninstall.

2

Tivoli Management Framework for AS/400 Endpoints

In the Tivoli environment, an ***endpoint*** is any system that runs an endpoint service (or daemon) and receives Tivoli operations. Typically, an endpoint is installed on a machine that is not used for daily management operations. Endpoints run a very small amount of software and do not maintain a database. Each endpoint is assigned to an ***endpoint gateway***, which controls all communications between a group of endpoints and the rest of the Tivoli environment. The gateway can launch methods on an endpoint or run methods on the endpoint's behalf.

With Tivoli, you can install and configure AS/400 systems as Tivoli endpoints using the Tivoli Management Framework. Tivoli for AS/400 Endpoints requires TCP/IP services to install and use AS/400 endpoints—for example, FTP is required for installing an AS/400 endpoint with the Software Installation Service (SIS). On AS/400 systems, use the **CFGTCP** command to configure table and server options. For more information on the Framework and endpoint architecture, refer to the *Tivoli Management Framework Planning and Installation Guide*.

Starting and Stopping AS/400 Endpoints

Tivoli provides the following Control Language (CL) commands to start and stop the endpoint daemon:

STRTMEEPT This command starts the AS/400 endpoint daemon. When installing an AS/400 endpoint, SIS or the **w4instlcf.pl** script starts the endpoint daemon job by executing the **STRTMEEPT** command. You must use this command to manually start the endpoint daemon job. You can also use the **STRTMEEPT** command in the auto-initialization program (QSYS/QSRTUP) to start the endpoint daemon job after an initial program load (IPL) or reboot.

ENDTMEEPT This command ends the AS/400 endpoint daemon, performs clean up operations on the AS/400, and disconnects from the gateway.

These commands are native AS/400 commands that you can execute from any AS/400 command line after the endpoint code is installed on the AS/400. You can also launch both commands from a remote machine. Both commands are shipped with authority set to **PUBLIC(*EXCLUDE)**. You must grant privileges to any users who will be using these commands. Additionally, you must also authorize these users to access the following associated command processing programs:

Library	Command Processing Program	Authority needed
QTMELCF	QLCFSTAR	*USE
QTMELCF	QLCFEND	*USE

Starting AS/400 Endpoints

The **STRTMEEPT** command starts the endpoint daemon process for a Tivoli endpoint. This endpoint daemon communicates with the endpoint gateway to receive and launch endpoint methods. The endpoint daemon job is always submitted to the QSYS/QSYSNOMAX job queue.

Note: **STRTMEEPT** is equivalent to the **$BINDIR/./lcf_bundle/generic/lcfd.sh** script, which is available for other platforms.

The **STRTMEEPT** command starts the QLCD job on the AS/400 with the appropriate configuration information. To enter endpoint configuration parameter from a prompt screen, press the **F4** key. The default value for all parameters is *NONE. If *NONE is used, the value of the parameter is set from the **last.cfg** file. If these parameters are not available in the **last.cfg** file or the **last.cfg** file is not available, the parameter values are assigned from internal default values. The following table lists and describes parameters that are available for the **STRTMEEPT** command:

Keyword and Description	Parameter	Value
LGNINTRFC Specifies the IP address, or host name and port number, of one or more gateways to which the endpoint will send its login packet. This option is required for the endpoint to log in to a gateway on a different subnet or to log in to a specific gateway when two or more gateways exist on a subnet.	Login interface	***NONE**
	Name	Hostname
	Port	Port number
GATEWAY Specifies the name of the Tivoli Management Gateway that will be used after the endpoint has successfully logged in. If the endpoint has not previously logged in, use the LGNINTRFC (or **lcs.login_interfaces** option in the configuration file) to provide one or more gateways through which the endpoint may login.	Gateway	***NONE**
	Name	Hostname
	Port	Port number
BCASTDSBL Disables the UDP broadcast. If you set this option to yes, you must use the **lcs.login_interfaces** option.	Broadcast disable	***NONE** ***YES** or **1** to disable ***NO** or **0** to enable

Keyword and Description	Parameter	Value
EPTNAME Specifies the name of this endpoint. This name is displayed when the **wep ls** command is executed in the gateway.	Endpoint name	***NONE** ***HOSTNAME** Use the host name as the name for this endpoint. *endpoint_name* Use a valid AS/400 endpoint name
PORT Specifies the port on which the endpoint daemon (LCFD) monitors gateway communications. The default value is 9494.	Local TCP/IP port	***NONE** *port_number*
MACHINEID Identifies the endpoint.	Machine unique ID	*machine_unique_id* Use a string that contains a unique identifier
THRESHOLD Defines the level of debug messages written to the **lcfd.log** file. **Note:** Level 3 and greater logging generates a large number of messages. For troubleshooting endpoints, Tivoli recommends using Level 2.	Log threshold	The following are valid entries: **0** No message logging **1** Minimal logging (default) **2** Tracing and moderate output **3** Data buffers and tight loops **4** Data
LOGSIZE Specifies the maximum size in bytes of the log file.	Log size	***NONE** *size* Use a value between 10240 and 10240000.
LOGQSIZE Specifies the maximum size in bytes of the log queue.	Log queue size	***NONE** *size* Use a value between 10240 and 10240000.

Keyword and Description	Parameter	Value
LOGFILE Specifies the name of the log file to be used for logging messages.	Log name	***NONE** *path_name* Where path name is the name of the integrated file system (IFS) file used for the log.
CACHESIZE Specifies the maximum size of the method cache. **Note:** When the maximum size is reached, the least recently used methods are deleted from the cache.	Cache size	***NONE** *size* and *integer*
UPINTRVL The number of seconds between endpoint broadcast calls.	UDP interval	***NONE** *seconds*
UDPATTMPT The number of times an endpoint will transmit a broadcast call.	UDP attempts	***NONE** *number of times* where *number* is an integer value
STRTMEOUT The amount of time in seconds before a communications timeout occurs during login.	Start timeout	***NONE** *seconds*
RUNTMEOUT The amount of time in seconds before a communications timeout occurs following a successful login.	Run timeout	***NONE** *seconds*
CFGFILE The name of the configuration file to be used to start the endpoint.	Configuration file name	***NONE** *config_file_name*

Keyword and Description	Parameter	Value
RUNDIR Specifies the name of the directory from which the endpoint code will run on the AS/400.	Running directory	***NONE** *directory_name*

Stopping AS/400 Endpoints

The **ENDTMEEPT** command stops the endpoint daemon process for a Tivoli endpoint. The job may be on a job queue, it may be active within a system, or it may have already completed running. Spooled files for an endpoint process that has been stopped remain in the output queue. Note, however, that the **ENDTMEEPT** command does not end any application jobs that have been started by the endpoint daemon—these jobs continue to run.

Use the **DELAY** option with the **ENDTMEEPT** command to specify if the endpoint is to end in a controlled manner with a time delay. The following syntax specifies valid parameters for the **ENDTMEEPT** command:

```
ENDTMEEPT [OPTION(*CNTRLD|*IMMED)[DELAY(30|1-9999999)]]
```

where:

***CTRLD** Specifies the default option that enables the endpoint daemon to finish any pending requests within the amount of time specified by **DELAY** (in seconds). The default delay is 30 seconds.

***IMMED** Specifies to stop the endpoint immediately. Use this option only when a ***CTRLD** end fails to end the job and a delay time is not used.

DELAY Specifies the timed delay in seconds after which the endpoint daemon is stopped.

Using the AS/400 Task Library for Common Tasks

Tivoli provides an AS/400 task library to automate tasks that are performed frequently by an AS/400 operator. These tasks are primarily designed to help automate Tivoli Enterprise Console (TEC) actions or to aid in remotely managing an AS/400 system. As with all AS/400 commands or jobs, these commands must be executed under the explicit authority of an AS/400 user profile that is specified in the task dialog box. If no profile is specified in the dialog box, the user ID running the desktop is used.

Tasks for AS/400 can be batch jobs, REXX scripts, or compiled programs. The following tasks are provided:

Command Enables you to enter any AS/400 CL command, valid in batch mode, for which all parameters are known. Prompting is not supported.

Send_Reply Invokes the AS/400 command **SNDRPLY** to send a reply to a message on a specific message queue.

Run_Backup Invokes the AS/400 command **RUNBKUP** to start saving information on your AS/400.

Start_cleanup Invokes the AS/400 command **STRCLNUP** to start cleaning up information on your AS/400, such as spool files or temporary libraries.

Power_Down_System
Invokes the AS/400 command **PWRDWNSYS** to end all subsystems, power down the system, and optionally re-IPL a remote AS/400 system.

Vary_Configuration
Invokes the AS/400 command **VRYCFG** to reset lines, controllers, and devices attached to an AS/400.

Note: Ensure that the maximum length of each line in any source script file does not exceed 92 characters.

For more information on these and other CL commands, refer to the *IBM AS/400 Control Language Reference Manual*. For more information on Tivoli tasks, refer to the *Tivoli Management Framework User's Guide*.

Tivoli Commands for AS/400 Systems

You can also use Tivoli commands to perform operations on AS/400 endpoints. However, the **lcfd** command is replaced by the **STRTMEEPT** command on the AS/400.

AS/400 Unique Features

Tivoli supports networks that utilize dynamic IP addresses. However, Dynamic Host Configuration Protocol (DHCP) is not currently supported on AS/400 systems so this Tivoli feature is not currently available.

You may need to specify longer timeout values if you perform Tivoli operations that time out frequently, such as inventory queries, task library tasks, or software distributions. Use the **wgateway** command to change the timeout value. Refer to the *Tivoli Management Framework Reference Manual* for more information about the **wgateway** command.

AS/400 User Profiles

When you configure an AS/400 system with the endpoint code, Tivoli creates three user profiles on the endpoint. These profiles are shipped without passwords and are used by the endpoint code. The following table describes the generated profiles:

User Profile	Description	Authority needed
QTIVOLI	LCF job runs under QTIVOLI. This user profile is required to start the endpoint and owns all AS/400 unique Tivoli objects. This profile must have *UPDATE authority to all the directories in the path to the configuration files.	***JOBCTL** ***SAVSYS** ***SPLCTL**

User Profile	Description	Authority needed
QTIVROOT	Used when the management server requires root authority.	***ALLOBJ** ***AUDIT** ***IOSYSCFG** ***JOBCTL** ***SAVSYS** ***SECADM** ***SERVICE** ***SPLCTL**
QTIVUSR	Equivalent to the UNIX **nobody** user ID.	No special authority

Note: If any of these profiles is accidentally deleted or corrupted, you must delete and reinstall the AS/400 endpoint code to recreate the user profiles and the objects that belong to them.

3

Tivoli User Administration for AS/400 Endpoints

System administrators can spend a considerable amount of time managing user and group accounts in a distributed enterprise. User accounts are constantly added, removed, and updated as the enterprise and user needs change.

The AS/400 operating system, or OS/400, provides numerous user management functions, and stores specific user information in specially architected objects called AS/400 User Profiles.

Tivoli User Administration for AS/400 provides the tools needed to manage user accounts on OS/400 systems, and integrates this support with other operating system types in the network. This product includes the following user and group management features:

- Single graphical interface used to manage user accounts on AS/400 systems along with UNIX, Windows NT, and NetWare machines
- Templates that enable you to enter a minimal amount of information to create a complete record
- Central location for quickly locating a specific user record
- Single sign-on support
- Central administrator control over accounts coupled with limited user control

- Management of general user information as well as AS/400, UNIX, Windows NT, and NetWare account information in one user record

Refer to the following background topics in the *Tivoli User Administration User and Group Management Guide* before using the AS/400 support:

- How user and group management works
- User and group account components
- User and group profiles
- User and group profile policy
- OnePassword utility
- Password control
- The User Locator
- Record locking
- Security management
- Examples of managing a large number of user

Using User Administration for AS/400

Tivoli User Administration for AS/400 is a profile-based application that runs on your AS/400 system to manage AS/400 user profiles. With this product, you can populate a user profile with records from any managed node and, after you update the profile records, you can distribute the profile to any other AS/400 endpoint.

Additionally, you can manage and organize AS/400 user profiles on different systems from one central location by using the underlying framework and tools provided by the Tivoli Management Framework and Tivoli User Administration.

AS/400 Attributes

The AS/400 attributes that can be managed with Tivoli User Administration for AS/400 are directly related to OS/400 commands that are used to change, audit, and delete AS/400 user profiles. The

Tivoli desktop is organized to help manage the AS/400 attributes by the following new subcategories under the new OS/400 category:

- OS/400 Auditing
- OS/400 Directory
- OS/400 Display Session
- OS/400 Group Membership
- OS/400 International
- OS/400 Login
- OS/400 Output
- OS/400 Password
- OS/400 Privileges
- OS/400 Session Start-up

AS/400-Specific Functions

The User Administration functions have been extended to AS/400 systems using custom code that translates information between Tivoli and OS/400 user profile format. The following functions are provided:

- Populate/Discover
- Distribute/Update
- Passwords

Populate/Discover

You can use the Tivoli desktop to populate all AS/400 user profiles from an AS/400 endpoint. Tivoli recommends that you logically organize your profiles into manageable groups that are tailored to your business. You can also use the **wpopusrs** command to populate selected user profiles to an AS/400 endpoint. Refer to the *Tivoli User Administration User and Group Management Guide* for information about creating and populating AS/400 user profiles. This document also provides command line syntax and examples for the **wpopusrs** command.

Distribute/Update

You can use the graphical user interface or the **wdistrib** command to distribute modified records. The AS/400 endpoint receives the data and makes the modifications to the AS/400 user profiles.

Passwords

Tivoli manages AS/400 user passwords as follows:

- By changing the password natively on the AS/400 system. If a password is changed with OS/400 CL commands, it is not overwritten by the Tivoli user profile password until the Tivoli user profile password is explicitly changed.
- By changing the **OS400_PassW** attribute using the Tivoli desktop.
- By running the **wpasswd** command from the command line interface.
- By changing the common password with the **OnePassword** web tool, which will change the OS400 password. This is also specified with the **–L** option to the **wpasswd** command.

When populating a user profile from an AS/400 endpoint, Tivoli User Administration sets the **OS400_PassW** attribute to the value of the **OS400_Login** attribute. Remember this setting convention when distributing user profiles to systems that do not allow passwords to match login names.

Unique Features for AS/400

Tivoli provides the following Tivoli User Administration features for AS/400 systems:

- Customizing deletion of AS/400 User Profiles. Use the **OS400_DLT_Action** and the **OS400_DTL_NewOwn** attributes to specify what actions to take with any objects owned by the AS/400 user profiles to be deleted.

- Controlling Group Membership on AS/400 endpoint. Use the **OS400_Prim_GRP** and **OS400_Supp_GRPs** attributes to control group membership in AS/400 endpoints.
- A script to help manage your user profiles in smaller blocks.

Use the **w4getusers.pl** script to create files to use with the **wpopusrs -f** command. The script is located in the **/usr/Tivoli/bin/lcf_bundle/generic** directory. The following example is the syntax for the **w4getusr.pl** command and its parameters:

```
root@ibmtmp1 [161]./w4getusr.pl

  Usage:./w4getusr.pl <option> host-name
  -d <dir> Destination directory where to write output files
  -n <Number> Maximum of number of entries to put in each file.

  root@ibmtmp1 [164]mkdir /tmp/dddd

  root@ibmtmp1 [166]./w4getusr.pl -d /tmp/dddd -n 10 rocket

  /tmp/dddd

  Machine 'rocket' NOT found in.netrc

  /tmp/dddd

  We need a user account, please type or just hit enter to use
  [default]

  login for rocket ([QSECOFR]): nlloyd
  Type new password for 'nlloyd':
  Total Received : 117250 root@ibmtmp1 [167]ls /tmp/dddd
  usr0 usr1  usr2  usr3  usr4  usr5  usr6  usr7
```

This sequence of commands creates an ordered list of files in the destination directory. You can use these files with the **wpopusrs** command to populate a user profile from the specified files.

AS/400-Specific Account Information

The AS/400 attributes and their associated values, which are being mapped from **CHGUSRPRF**, **CHGUSRAUD**, and **DLTUSRPRF** commands, are listed in the following table. If you are using the Tivoli desktop, these values are shown with the AS/400 attributes. If you are using the command line, you must type the AS/400 attributes with associated values.

For more information about creating new user accounts from the Tivoli desktop or from the command line, refer to the *Tivoli User Administration User and Group Management Guide*.

AS/400 Attribute and Description	Keyword Values
OS400_Accounting_Code Specifies the accounting code, if any, associated with this user profile.	***BLANK** *endpoint_name* An endpoint name (up to 15 characters)
OS400_Assist_Level Specifies the level of help information available.	***SYSVAL** ***BASIC** ***INTERMED** ***ADVANCED**
OS400_Attention_PGM Specifies the attention program associated with this user.	***NONE** ***SYSVAL** ***ASSIST** *program* A valid program name in the form LIB/PROGRAM
OS400_CCSID Specifies the coded character set ID to use for this profile. Used for national language purposes.	***SYSVAL** ***HEX** *ccsid* An AS/400 CCSID
OS400_Change_Rollkey Specifies whether the opposite of the system default actions are performed when the roll key is pressed.	***YES** ***NO**

AS/400 Attribute and Description	Keyword Values
OS400_Class Specifies to which class the user profile belongs.	***USER** ***SYSOPR** ***PGMR** ***SECADM** ***SECOFR**
OS400_Command_Audit Specifies whether to log CL command strings and procedures for the user profile.	***YES** ***NO**
OS400_Country Specifies the country identifier to use with this profile. Used for national language purposes.	***SYSVAL** *country_id* An AS/400 country ID
OS400_Current_Lib Specifies the name of the library associated with the currently running job.	***CRTDFT** *name* A valid AS/400 library name (10 characters or less)
OS400_Directory Specifies the path name for the home directory of the user profile.	***USRPRF** *path* A “/” delimited path to the home directory
OS400_Display_Help Specifies whether online help information is displayed as full screen instead of in a window.	***YES** ***NO**

AS/400 Attribute and Description	Keyword Values
OS400_DLT_Action Specifies the action to take when deleting the user profile. Before deleting a user profile, all objects owned by this user profile must be deleted or transferred to another owner. **Note:** This attribute is an interface to the **DLTUSRPRF** command.	***NODLT** ***DLT** ***CHGOWN** *name*
OS400_DLT_NewOwn Specifies the owner to transfer objects to when the user profile is deleted on an AS/400 system.	***CHGOWN** *name*
OS400_Initial_Menu Specifies the name of the initial menu to display when this user signs on.	***SIGNOFF** *initial_menu* A valid menu name in the format LIB/MENU
OS400_Initial_PGM Specifies the name of the initial program to call when this user signs on.	***NONE** *initial_program* A program name in the form LIB/PROGRAM
OS400_JOBD Specifies the name of the job description used for jobs that start through subsystem workstation entries. If the job description does not exist when the user profile is created, a library qualifier must be specified (because the job description name is kept in the user profile). **Note:** For the keyword values except **QDFTJOB**, you must specify the path to this name.	***QDFTJOB** ***LIBL** ***CURLIB** *library_name* The name of the library where the job description is located
OS400_Job_Audit Specifies whether all job start and stop data is audited for this user.	***YES** ***NO**

AS/400 Attribute and Description	Keyword Values
OS400_KBD_Buffering Specifies the type of keyboard buffering, if any, to be used or allows the buffering to default to a system value.	***SYSVAL** ***TYPEAHEAD** ***YES** ***NO**
OS400_Language Specifies the language identifier used for this user profile.	***SYSVAL** *language_id* An AS/400 language ID
OS400_Limit_Cap Specifies whether to limit the capabilities for this user profile or to set them to a system value.	***PARTIAL** ***YES** ***NO**
OS400_Limit_Dev_Sess Specifies whether to limit the device sessions for this user or to set them to a system value.	***SYSVAL** ***YES** ***NO**
OS400_Locale Specifies the path name of the locale assigned to the **LANG** environment variable. **Note:** Supported in releases later than V3R2.	***SYSVAL** ***NONE** ***C** ***POSIX** *path* A “/” delimited path to the locale
OS400_Loc_CCSID Specifies whether to use the coded character set ID from the locale. **Note:** Supported in releases later than V3R2.	***YES** ***NO**

AS/400 Attribute and Description	Keyword Values
OS400_Loc_DATFMT Specifies whether to use the date format from the locale. **Note:** Supported in releases later than V3R2.	***YES** ***NO**
OS400_Loc_DATSEP Specifies whether to use the date separator from the locale. **Note:** Supported in releases later than V3R2.	***YES** ***NO**
OS400_Loc_NONE Specifies whether **NONE** is set for job attributes associated with locale. **Note:** Supported in releases later than V3R2.	***YES** ***NO**
OS400_Loc_SRTSEQ Specifies whether to use the sort sequence from the locale. **Note:** Supported in releases later than V3R2.	***YES** ***NO**
OS400_Loc_SYSVAL Specifies whether **SYSVAL** is set for job attributes associated with the locale. **Note:** Supported in releases later than V3R2.	***YES** ***NO**
OS400_Loc_TIMSEP Specifies whether to use the time separator from the locale. **Note:** Supported in releases later than V3R2.	***YES** ***NO**
OS400_Login Specifies the user login.	*name* A 10 characters or less string
OS400_Login_Enablement Specifies whether the user can log in.	***ENABLED** ***DISABLED**

AS/400 Attribute and Description	Keyword Values
OS400_Login_Text Specifies the textual description of the user profile.	*description* A 50-character or less string
OS400_Max_Storage Specifies the maximum amount of auxiliary storage allocated for permanent objects.	***NOMAX** *max_kbytes* The number of kilobytes allowed
OS400_MSGQ Specifies the message queue associated with this user.	***USRPRF** *queue_name* A queue name in the form LIB/QUEUE
OS400_MSG_Delivery Specifies how the messages are delivered. These messages are sent to the message queue associated with the user profile.	***NOTIFY** ***HOLD** ***BREAK** ***DFT**
OS400_MSG_Sev_Level Specifies the lowest severity a message can have and still be delivered to a user in **BREAK** or **NOTIFY** delivery mode.	*severity_code* An integer from 0 to 99
OS400_Obj_Audit Specifies the level of object auditing for this user. **Note:** This attribute is an interface to the **CHGUSRPRF** command.	***NONE** ***CHANGE** ***ALL**
OS400_Obj_Create_Audit Specifies whether auditing entries are sent when this user creates objects.	***YES** ***NO**

AS/400 Attribute and Description	Keyword Values
OS400_Obj_Delete_Audit Specifies whether audit entries are sent when this user deletes objects.	***YES** ***NO**
OS400_Obj_Mgmt_Audit Specifies whether audit entries are sent when this user makes object management changes, such as object move or rename.	***YES** ***NO**
OS400_Office_Audit Specifies whether audit entries are sent when this user performs any office services change.	***YES** ***NO**
OS400_Optical_Audit Specifies whether audit entries are sent for optical device changes performed. **Note:** This attribute is not supported for V3R2; it is ignored when populating or distributing to V3R2.	***YES** ***NO**
OS400_OUTQ Specifies the output queue associated with this user.	***WRKSTN** ***DEV** *queue_name* A queue name in the form LIB/QUEUE
OS400_PassW Specifies the password. The password must be changed explicitly in the GUI or with the Tivoli common user password. During populate, passwords are set to the user profile name.	*password* A 10 characters or less string
OS400_PassW_Expired Specifies whether the password has expired.	***YES** ***NO**

AS/400 Attribute and Description	Keyword Values
OS400_PassW_ExpInv Specified the password expiration interval.	***SYSVAL** ***NOMAX** *interval* A integer from 1 to 366 indicating the number of days between when a password is changed to when it expires
OS400_PGM_Adopt_Audit Specifies whether authority obtained through program adoption is audited.	***YES** ***NO**
OS400_Prim_GRP Specifies primary group membership.	***NONE** *group_name* A GROUP on an AS/400 endpoint. The GROUP name must be known, it cannot be queried.
OS400_Printer_Device Specifies the name of the default printer.	***WRKSTN** ***SYSVAL** *name* A printer defined on an AS/400 endpoint
OS400_Priority Specifies the highest priority level for job submission.	*number* An integer from 0 to 9
OS400_Priv_ALLOBJ Specifies whether to grant all object authority for accessing system resources.	***YES** ***NO**

AS/400 Attribute and Description	Keyword Values
OS400_Priv_AUDIT Specifies whether to grant audit authority.	***YES** ***NO**
OS400_Priv_IOSYSCFG Specifies whether to grant I/O configuration authority.	***YES** ***NO**
OS400_Priv_JOBCTL Specifies whether to grant job control authority to change, display, hold, release, cancel, or clear any job that is running.	***YES** ***NO**
OS400_Priv_SAVSYS Specifies whether to grant the authority to save, restore, or free storage for all objects on the system.	***YES** ***NO**
OS400_Priv_SECADM Specifies whether to grant security administration authority to create or change user profiles.	***YES** ***NO**
OS400_Priv_SPLCTL Specifies whether to grant spool control authority.	***YES** ***NO**
OS400_Priv_Service Specifies whether to grant service authority.	***YES** ***NO**
OS400_SAVRST_Audit Specifies whether audit entries are sent for save and restore actions.	***YES** ***NO**
OS400_Security_Audit Specifies whether audit entries are sent for security changes.	***YES** ***NO**

AS/400 Attribute and Description	Keyword Values
OS400_Send_Message Specifies whether a message is sent to the owner of a spooled file indicating that printing was completed or held.	***YES** ***NO**
OS400_Service_Audit Specifies whether audit entries are sent for system service tools.	***YES** ***NO**
OS400_Show_Details Specifies whether additional detailed options are shown when defining or changing the system.	***YES** ***NO**
OS400_Show_Status_MSG Specifies whether status messages are sent.	***YES** ***NO**
OS400_Show_Parm_Keys Specifies whether parameter keywords are shown instead of possible keyword values when a command is displayed.	***YES** ***NO**
OS400_Signon_Info Specifies whether to display sign on information or to set it to a system value.	***SYSVAL** ***YES** ***NO**
OS400_Sort_Sequence Specifies which sequence to use for string comparisons.	***SYSVAL** ***HEX** ***LANGIDUNK** ***LANGIDSHR** *object* A object in the form LIB/NAME

AS/400 Attribute and Description	Keyword Values
OS400_Spool_Audit Specifies whether audit entries are sent for spooled file operations made by this user.	***YES** ***NO**
OS400_Supp_GRPs Specifies supplemental group membership. **Note:** When specifying multiple supplemental groups from the command line, the groups must be specified as a comma-separated item.	***NONE** *grp1 … grpn* A list of GROUP names. Up to 15 groups can be specified
OS400_Sys_Mgmt_Audit Specifies whether audit entries are sent for grant system management actions made by this user.	***YES** ***NO**
OS400_UID Specified the AS/400 user ID.	***GEN** A system-generated ID *number* An integer from 1 to 4294967294

Modifying OS/400 User Accounts

Use the following commands provided with Tivoli User Administration to modify OS/400 attributes for user accounts.

- The **wcrtusr** command to create a new user.
- The **wgetuser** command to list user information in an existing profile.
- The **wsetusr** command to change attributes in an existing user profile.

The following examples in this section describe how to create a Tivoli profile for an AS/400 user, query information about this profile, and set additional attributes for this user.

The **wcrtusr** command creates a new user in the specified user profile. For example, to create a user profile for user **nlloyd**, an employee in Austin, Texas, enter the following **wcrtusr** command to initialize the user record with employee name, telephone number, and profile name attributes.

```
wcrtusr -e 512-555-1212 -x OS400_Login NICK
        @UserProfile:Austin nlloyd
```

where:

–e 512-555-1212
Specifies a phone extension.

–x OS400_Login NICK
Sets the value of the OS/400 attribute name to NICK.

@UserProfile:Austin
Specifies the name of the profile in which to create the user record.

nlloyd Specifies the name of the profile in which to create the user record.

The **wgetusr** command lists information about an existing user from the specified profile. For example, to query the **OS400_DLT_Action** attribute in the AS/400 user profile for **nlloyd**, enter the following command from the command line:

```
wgetusr -x OS400_DLT_Action @UserProfile:Austin nlloyd
```

where:

–x OS400_DLT_Action
Returns the value of the **OS400_DLT_Action** attribute.

@UserProfile:Austin
Specifies the name of the user profile to query.

nlloyd Specifies the login name.

The **wsetusr** command modifies the attributes of an existing user in the specified user profile. For example, to set the value of the

OS400_DLT_Action attribute for users **nlloyd** and **nmcginnis**, enter the following command from the command line:

```
wsetusrs -x OS400_DLT_Action *DLT @UserProfile:Austin
         nlloyd nmcginnis
```

where:

–x OS400_DLT_Action
Sets the value of the **OS400_DLT_Action** attribute.

@UserProfile:Austin
Specifies the name of the user profile in which the new information is set.

nlloyd nmcginnis
Specifies the login names.

Refer to the *Tivoli User Administration User and Group Management Guide* for more information about creating and modifying user records in user profiles, as well as for command syntax and usage.

4

Tivoli Security Management for AS/400 Endpoints

One of the goals of managing and protecting resources in a distributed network is avoiding breaches in security. Security violations can occur if an outside intruder attempts to enter the computer network to compromise confidential files or other network resources. Employees within a company can also jeopardize network security. Security administrators must implement and maintain security across all AS/400 systems that enables users to have access to the tools, files, and systems they need to do their jobs while restricting access to all other resources.

The OS/400 operating system is designed with inherent security measures. Objects are secured at the object level, user authorities are controlled through user profiles, and system values help manage the security of each AS/400 machine at the system level. System level security can be set to enable lesser or greater degrees of security. Additionally, the OS/400 operating system is C2 certified.

Tivoli Security Management Features

Tivoli Security Management provides the following features:

- Enables you to implement a comprehensive, consistent security policy across your enterprise using a role-based security model. ***Roles*** correspond to job functions.

- Increases the availability and integrity of systems by enforcing centrally managed security.
- Limits the days and times users can log in or access system resources.
- Produces a complete security audit trail that enables you to audit any combination of login attempts or failures and resource access attempts or failures.
- Provides a set of tasks to query which users have access to resources and to retrieve audit log information from OS/400 endpoints.

Tivoli Security Management protects a wide variety of network resources, including files, directories, terminals, OS/400 objects, and protected processes. To manage these resources, Security Management provides ***security profiles*** that contain the definitions of all groups, resources, roles, and system-wide policies, each of which is stored in its own type of record. Records are stored in a system-independent format that allows profile records to be distributed across an enterprise that contains many different system types.

For more information about configuring Tivoli Security Management to ensure a secure network environment, see the *Tivoli Security Management User’s Guide*.

OS/400 Native Security

OS/400 native security is built into the operating system itself and shipped on every AS/400. OS/400 security involves three parts:

- System values as described in “AS/400 System Values for Security” on page 4-3.
- User profiles for individuals and groups as described in “AS/400 User Profiles” on page 4-3.
- Object authorities as described in “AS/400 Object Authorities” on page 4-4.

AS/400 System Values for Security

The first step in securing your AS/400 system is to set an overall security level for the entire operating system using the IBM supplied system value, **QSECURITY**.

The system values range from no security to enhanced security. To run Tivoli Security Management, **QSECURITY** must be set to “30” resource level security, or higher. Use the AS/400 Control Language command, **WRKSYSVAL**, to change the value to 30 or higher before you plan to use this product.

You can use other AS/400 system values to manage the security of AS/400 user profile passwords, system sessions, access from remote systems, and system auditing functions. For more information about these system values, refer to the *AS/400 Security Reference Manual*, which can be accessed at the **www.as400.ibm.com** web site.

AS/400 User Profiles

AS/400 user profiles identify each user to the native security system and contain parameters that define the following:

- Security attributes like passwords and special authorities
- Group profile information
- Signon attributes
- Job settings
- National Language settings

When user profiles are created, users can be associated with user classes, such as *USER, *SYSOPR, or *PGMR, and they can be granted special authorities, such as *ALLOBJ or *AUDIT. These attributes help determine which OS/400 CL command the user can execute. However, when used by themselves these attributes do not provide a secure system from an application perspective.

AS/400 Permissions

AS/400 profiles can have the following special permissions assigned for objects in the QSYS library:

Permission	Description
Change	Allows all operations except those limited to the owner to be controlled by object existence and object management authority.
All	Allows all operations except those limited to the owner.
Use	Allows read and execution for objects.
Exclude	Prevents the user from using this particular object.

AS/400 Group Profiles

AS/400 group profiles allows users with similar jobs to share authorities without having to share the same password. AS/400 user profiles are associated with one or more group profile after the group was created. Group profiles should not passwords and should not own objects that help enforce system security. All group profiles created by Tivoli Security Management are created without passwords to help ensure the integrity of your AS/400 security.

AS/400 Object Authorities

OS/400 is an object based operating system that requires users to have appropriate authority before accessing or using objects. This authority can come from the following sources:

- *PUBLIC authority — All objects have *PUBLIC authority until revoked by the owner of the object.
- *ALLOBJ special authority — You have special authority to all objects on the system.

- A “private” authority — The owner of the object gives you certain authority to a specific object.
- An authorization list associated with the object — An ***authorization list*** is way to group similar objects, such as all the files associated with an application, and give users that same authority to all of the objects on the list.
- Primary group authority — A group profile that you are associated with was authorized for the object.
- Adopted authority — Under certain circumstances, OS/400 user profiles adopt additional authority from another specific user profile. Through this process, you can be authorized to objects that you would not otherwise be able to access. This type of authority is temporary and is disabled when the adoption terminates.

Most users will implement AS/400 security through a combination of securing libraries and objects, and by using both AS/400 user profiles and AS/400 group profiles.

AS/400 Object and Data Authorities

Objects can be secured with specific authorities or permissions as defined in the following tables. AS/400 associates authorities with both the object and the data it can contain.

Object Permission	Description
Object Alter	Allows changes to object attributes. Objects can be stored in the QSYS library or the integrated files system (IFS).
Object Management	Allows specification of object security. The object can be renamed, moved, or members can be added to a file. Objects can be stored in the QSYS library or the IFS.

Object Permission	Description
Object Exist	Controls the existence and ownership of the object. The object can be deleted or transferred to a different owner. Objects can be stored in the QSYS library or the IFS.
Object Reference	Allows one object to reference another object. Used only by data base files. Objects can be stored in the QSYS library or the IFS.
Object Operational	Allows users to view the description of an object and use the object as defined by the data authority (refer to the following table). This applies only to objects stored in the QSYS library.

Data Authorities apply to the data portion or contents of an AS/400 object as follows:

Data Authorities	Description
Add	More entries or data can be added to an object. This applies only to objects stored in the QSYS library. It does not apply to the IFS.
Delete	Entries or data can be deleted from the object. This applies only to objects stored in the QSYS library. It does not apply to the Integrated File System.
Execute	The object can be used (for example, executed). Objects can be stored in the QSYS library or the IFS.
Read	The data can be read but not changed. Objects can be stored in the QSYS library or the IFS.

Data Authorities	Description
Update	The data can be read, changed, and saved. This applies only to objects stored in QSYS library. It does not apply to the IFS.
Write	The data can be read, changed, and saved. This applies only to objects stored in the IFS.
Exclude	The data can not be read or changed. This applies only to objects stored in the IFS.

AS/400 Auditing

OS/400 also uses a native audit function in addition to the security function. You can audit by using the following actions:

- System-wide actions
- User-level actions
- Object access
- Specific user access enabled for a given object

For more information about auditing, refer to the *AS/400 Security Reference Manual*, which can be accessed at the **www.as400.ibm.com** Web site.

Authorization Roles and Notification Groups

AS/400 supports the same authorization roles and notification groups defined by Tivoli Security Management. For a complete list of authorization roles and operations written to the Tivoli Security Management notice group, see the *Tivoli Security Management User's Guide*.

Populating an AS/400 Security Profile

Populating a security profile is similar to populating other Tivoli profiles. Security Management discovers security-related information from a managed system and places each element into a record. On AS/400 endpoints, the program uses native OS/400 security functions.

The unique feature in populating security profiles is the ability to restrict the type of data populated. Tivoli Security Management enables you to limit the populate to a specific record type. For example, you can populate a set of group records without simultaneously populating role records. This feature not only helps save time, but also allows you to quickly configure which resources and groups need to be assigned to a role.

The security profile distribution process also has the following unique features:

- Populating all records into a security profile, as described in “Populating All Records into a Security Profile” on page 4-8.
- Populating specific records into a security profile as described in “Populating Specific Records into a Security Profile” on page 4-9.

Populating All Records into a Security Profile

When you populate a security profile, Tivoli Security Management reads data from the OS/400 native system security and places this data in the appropriate group, role, resource, and system policy records.

Note: OS/400 resource records must be populated at the record level rather than at the profile level. If profile level actions were permitted, every file, directory, and registry entry on the endpoint would be its own resource record.

For a complete description of how each record type is populated, see the *Tivoli Security Management User’s Guide.*

In most cases, you will need to review and edit the contents of each record before you distribute the profile. The settings defined within a security profile do not take effect until the profile is distributed.

You can perform this task from the Tivoli desktop or the command line. For instructions on using the desktop interface, see *Tivoli Security Management User’s Guide.*

The following command line example populates an entire security profile:

```
wpopulate @ManagedNode:olympus @SecurityProfile:firestorm
```

where:

olympus	Specifies the name of the system from which to populate.
firestorm	Specifies the name of the profile to populate.

For more information, refer to the **wpopulate** command in the *Tivoli Management Framework Reference Manual*.

Populating Specific Records into a Security Profile

This section describes how Tivoli Security Management populates a profile at the record level.

OS/400 Group Records

Tivoli Security Management populates a security profile by importing system-level groups, user membership, and audit control information from the specified endpoints into security group records in the current security profile.

During a populate operation, Security Management also creates a new group record for each unique system-level group name that is imported from the specified endpoints. If the group name already exists within the security profile, the product adds or updates the user membership list using the associated system-level group’s user membership list. Tivoli Security Management for AS/400 maps these security group records to OS/400 group profiles.

Role Records

Roles are used to organize resource access needs based on job function need. With the exception of NT, most system-specific security services do not directly support this concept. As a result, OS/400-oriented roles are not created when you populate a security profile from OS/400 systems (unless roles existed from previous profile distributions). You can begin the process of defining roles by performing a selective populate of a role from the resource/access permission list of an existing group.

Resource Records

Tivoli Security Management populates a security profile by importing system-level resources and their associated access control information into resource records. Each resource defined in the system-level database is mapped to a unique resource record with the same name. The resource type assigned to the record depends on the resource class in the system-level database. The resource's security attributes are assigned to the equivalent Security Management resource attributes.

For OS/400, only one record can be populated at a time. Each record will be associated with a single OS/400 resource or object.

System Policy Records

Tivoli Security Management populates a security profile by importing system-level global security policy information from the specified endpoints into a single security system policy record in the current security profile. If a system policy does not already exist, Security Management creates a new system policy record for each specified endpoint and its associated security policy attributes during a populate operation.

Procedures to Populate Records

You can populate security records from the Tivoli desktop or from the command line. Although you can simultaneously populate several managed nodes from the Tivoli desktop, you can only populate records for one managed node from the command line interface. Refer to the *Tivoli Security Management User's Guide* for a detailed description of how to use the desktop interface.

The following command line example populates resource records from host **olympus** into a security profile:

```
wpopsec Resource olympus secprof
```

The **wpopsec** command imports group, resource, role, or system policy information from a specified source profile and allows you to populate single records or all types of records simultaneously.

Additional restrictions for OS/400 endpoints include the following:

- You cannot populate values for login times for groups or access times for resources.
- OS/400 populates only one resource at a time.
- Use of wildcards to get several resources simultaneously is not supported.
- Directories or stream files stored in IFS cannot be populated on an AS/400 at V3R2.

Setting Profile Distribution Defaults

Before distributing a Tivoli Security Management profile, you can control the default profile distribution settings. When you distribute this profile, Security Management uses the default options you have defined from the **Distribute Defaults** dialog. These settings also determine default options selected when you use the **Distribute Profile** dialog from the **Security Profile** pop-up menu or the **Distribute** option of the **Security Profile Properties** window. The defaults are set on a per-profile basis. Each profile's default options can be different from those of other profiles.

For OS/400 endpoint, the **Exact Copy** option only affects Tivoli controlled groups and resources. Groups and resources that are not controlled by Tivoli are not affected by an **Exact Copy** distribution.

Also, access control information is not defined in any corresponding resource or role record for new QSYS or IFS objects. The access control information is refreshed on any subsequent distribution regardless of whether the associated resource or role record was changed.

Note: During a profile distribution, access control information is reset by a resource or role record for each defined QSYS or IFS object whether or not it was changed. This update might temporarily affect access to these objects while the distribution is in progress.

Since each object must be reset, the number of objects and the system load affects the time to complete a distribution. Tivoli recommends that you distribute large numbers of objects when the system load is light. For more information about setting profile distribution defaults, see *Tivoli Security Management User's Guide.*

Distributing Security Profiles

Distributing a security profile places the information defined in the profile on each subscribing managed node. You must distribute a profile if you or another administrator have performed any of the following tasks:

- Populated a profile
- Added a record to the profile
- Deleted a record from the profile
- Edited an existing record

The **Distribute Profile** dialog and the **wdistrib** command provide an option for distributing an exact copy of the profile to all subscribers. The purpose of this feature is to overwrite all changes made on copies of the profiles distributed to subscribing endpoints or other security profiles.

The security profile distribution process differs from other Tivoli profiles. On an AS/400 endpoint, many records are not affected by the **Make Exact Copy** attribute. These records include:

- Group records. Only groups known to Tivoli Security Management are controlled. OS/400 group profiles that are not known to Security Management are not affected.
- Resource records. OS/400 objects that are not known to Tivoli Security Management are not affected.
- Resource Types. Only objects in OS/400 user libraries are affected. Objects in system-supplied libraries are not affected.

You can perform this task from the Tivoli desktop or the command line. For more information about using the Tivoli desktop or the **wdistrib** command, refer to the *Tivoli Security Management User's Guide.*

Adding Security Records

This section explains how to add new security records for an AS/400 system. Tivoli Security Management contains four types of records: system policy, group, resource, and role records. The following sections briefly describes how to define system policy records as well as how to set login and password policies.

Adding security records has the following features:

- Adding system policy records as described in “Adding System Policy Records” on page 4-13.
- Setting access policy for resource types as described in “Setting Resource Type Access Policy” on page 4-16.
- Assigning members to a group record as described in “Assigning Members to a Group Record” on page 4-16.

For detailed background information and procedures about creating systems records and defining policy attributes, refer to *Tivoli Security Management User’s Guide.*

Adding System Policy Records

A system policy record contains system-level security information for a collection of system- and application-level user IDs and resources. The specified system policy, when distributed, applies to all user IDs and resources, unless specifically overridden by settings in a resource record or in a Tivoli user profile.

Each system policy record is stored in a system-independent format that allows the same profile records to be distributed across an enterprise that contains heterogeneous system types. The information stored in each system policy record is mapped onto the system-level equivalent entities and associated attributes when the profile is distributed to profile endpoints.

Defining a System Policy Record

Tivoli Security Management does not automatically define a system policy record when you create a security profile. Normally, only one system policy record exists on each endpoint. If you have multiple records, Tivoli Security Management applies the system policy characteristics in the last record it encounters.

Setting Login Policy

Login policy defines how to handle inactive user accounts, and what to do if a user attempts to login and repeatedly specifies the wrong password. Login policy also states how many grace and concurrent logins a user can have. These two attributes can be overridden if the user is defined in a user profile.

The procedure for setting login policy globally and for endpoint-specific machines is the same. However, not all login policy options are available on all endpoints. The following table lists the valid audit control attributes for OS/400:

Attribute	OS/400
Suspend Inactive Accounts	Yes
Lock Account upon Multiple Login Failures	Yes[1 3]
Limit Grace Logins	No
Limit Concurrent Logins	Yes[2]

1. Accounts can be locked but they will not automatically be unlocked.
2. This limits users to having a single connection to the host.
3. For an AS/400 endpoint, the **Time span** field is not supported.

Setting Password Policy

Password policy defines the attributes of a user’s password. It can define how many and what type of characters a password must contain. For example, if you want to require each user’s password to contain a minimum of six characters, one of which must be a capital letter, then you set these attributes within the password policy. Several password policy attributes can be overridden if the user is defined in a user profile.

The procedure for setting password policy globally and for endpoint-specific machines is the same. However, not all password policy options are available on all endpoints. The following table lists the valid audit control attributes for OS/400:

Attribute	OS/400
Min Days between Password Changes	No
Max Days between Password Changes	Yes
Min Password Length	Yes
Min Alphabetic Characters	No
Min Alphanumeric Characters	No
Min Numeric Characters	No
Min Uppercase Characters	No
Min Lowercase Characters	No[1]
Max Repeated Characters	Yes[2]
Min Special Characters	No
Password History Depth	Yes

1. OS/400 user profile passwords are folded to upper case characters.
2. OS/400 allows no repeated characters, no adjacent repeated characters or unlimited repeated characters.

Notes::

- **Min Password Length** indicates the minimum number of characters that must appear in a password. The default is 5. For OS/400, the maximum length is 10 characters so **Min Password Length** cannot exceed 10.

- **Max Repeated Characters** indicates the maximum number of times a character can appear consecutively within a password. OS/400 does not allow repeated characters, adjacent repeated characters, or unlimited repeated characters.
- **Password History Depth** indicates the number of previous passwords that are stored. When a user supplies a new password, he or she cannot specify any of the passwords stored in the password history list. Valid values for OS/400 are 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, and 32.

Setting Resource Type Access Policy

Tivoli Security Management can be used to support objects that reside in ***ALLUSR** libraries. This support is for existing objects only and does not apply to new objects unless the security profiles are redistributed.

Assigning Members to a Group Record

You can install Tivoli User Administration to simplify the process of adding users to a group record. In environments where Tivoli User Administration is installed, the users defined in user profiles are available as potential members of a group record’s member list. To add a member to a group record from the desktop, you simply select the user profile in which the potential member is defined and double-click on the user’s name.

You can also apply the same techniques to add endpoint-specific user IDs to a group record. The Security Management graphical user interface allows the administrator to select a managed node from the list of subscribing managed nodes. Selecting a managed node displays the current list of user IDs defined on that managed node. These users can be added to the group record.

Security Management does not allow you to create new system-level user IDs. The program returns an error message if you specify a user that is not already defined on at least one of the subscribed endpoints.

The procedure for setting Tivoli users and endpoint-specific users is the same. However, values set on the **TME 10 User List** panel cannot be applied to the endpoint-specific panels.

Notes:

- You should plan who will be defined in a security group carefully. Users who belong to multiple groups might have conflicting access permissions, depending on how you define your roles and resources.
- When users are added to a user profile, the order in which the user and security profiles are distributed becomes important. A user profile must be distributed before the security profile.

Creating a Group Record

A group record defines member users. All members of a group record are assigned the same roles and have common login restrictions. Members can be populated from Tivoli User Administration user and group profiles, or from endpoint-specific files and databases. For additional information on the following topics, refer to the *Tivoli Security Management User's Guide*:

- Assigning members to a group record
- Assigning roles to a group record

AS/400-specific information for group records is described in the following sections.

Specifying an Endpoint-specific Group Name

The value entered in the **Group Name** field is the default group name for all endpoint types. However, if you want to ensure that OS/400, NT and UNIX groups have different names, you can do so by specifying endpoint-specific names. If you specify an endpoint-specific group name, the value always overrides the generic value. Group names on OS/400 must be 10 characters or less. For detailed procedures on specifying an endpoint group name refer to the *Tivoli Security Management User's Guide.*

Setting Audit Control

The audit control settings determine whether to log instances in which a user defined in the record attempts to login to any workstation or access any resource. Tivoli Security Management can write both successful and failed actions to the audit log.

Note: Use caution when auditing of resources. Each time a user accesses any resource, Security Management writes an entry to the audit log. If you audit commonly accessed resources, such as home directories, the audit log can fill up quickly.

The procedure for setting audit control globally and for endpoint-specific machines in a group record is the same. However, not all auditing options are available on all endpoints. The following table lists the valid audit control attributes for OS/400:

Attribute	OS/400
Login Attempt Successes	Yes
Login Attempt Failures	Yes
Resource Access Successes	No
Resource Access Failures	Yes

Setting Login Time Restrictions

Use Tivoli Security Management to determine when members of a group can log in to the network. You can allow users to log in on any day and at any time, or you can restrict the log in times to specific days and time spans. Login times must be specified in 24-hour periods.

The procedure for setting global login time restrictions and endpoint-specific restrictions is the same for OS/400 endpoints as for other platforms. In addition, all available fields are valid on the OS/400 endpoint.

Creating Resource Records

A resource record defines one or more system-level resources and assigns access permissions to these resources. You can also set access time restrictions, audit control, and network service access permissions from within a resource record. For more information and procedures about creating resource records, see the *Tivoli Security Management User's Guide*. AS/400-specific information for resource records is described in the following sections.

Creating resource records has the following unique features:

- Defining a resource record, as described in "Defining a Resource Record" on page 4-19.
- Setting default access permissions as described in "Setting Default Access Permissions" on page 4-20.
- Setting access time restrictions as described in "Setting Access Time Restrictions" on page 4-21.
- Setting audit control as described in "Setting Audit Control" on page 4-21.

Defining a Resource Record

Tivoli Security Management defines the following OS/400 resource types to protect confidential and critical resources for AS/400 endpoints.

OS/400 Supported Resource Types

The following table describes AS/400 resource types:

Resource Type	Description
DATA AREA	Specially named areas of OS/400 storage used to share data between OS/400 jobs.
DEVICE DESCRIPTION	OS/400 objects that describe an attached device or communications protocol connection.

Resource Type	Description
FILE	Any OS/400 file including its members or records.
JOB DESCRIPTION	An OS/400 object that describes other OS/400 resources that will be used by a specific job.
JOB QUEUE	A queue for jobs that are waiting to be executed.
LIBRARY	A special OS/400 object used to organize a collection of related objects. For example, an OS/400 application can be created with all of its programs and files in one library.
OUTPUTQ	The OS/400 queue where the job output is sent.
PROGRAM	An executable module that will run on the OS/400.
SERVICE PROGRAM	A bound set of modules used by a program. The service program is similar to a DLL and is often called by APIs from other applications.
DIRECTORY	An integrated file system (IFS) path.
STREAM FILE	An IFS file that contains the bytes of data or byte stream.

Note that Tivoli Security Management does not create resources on endpoints. For example, if you create a FILE resource record with a name of **/tmp/output**, and a file with this name does not exist, Security Management will not attempt to create the file upon distribution. For OS/400, the record will be saved and changes will be reapplied on subsequent distributions.

Setting Default Access Permissions

If a user attempts to access a resource, but does not have access permissions, the default access permissions defined in the resource record are applied. The set of default access permissions supported depends on the resource type selected on the **Resource Record Properties** dialog.

Setting Access Time Restrictions

Tivoli Security Management allows you to define when resources can be accessed. You can allow users to access the resource on any day and at any time, or you can restrict access times to specific days and time spans. Access times must be specified in 24-hour periods.

The procedure for setting global access time restrictions is the same for OS/400-specific restrictions. In addition, all fields are valid on OS/400. For OS/400 resources, a specific resource name must be specified. Wildcard characters, such as a ***** wildcard character, are allowed in OS/400 resource names.

Setting Audit Control

Tivoli Security Management allows you to audit any user attempts to access the resources defined in the resource record. The procedure for setting audit control globally and for endpoint-specific machines in a group record is the same. However, not all auditing options are available on all endpoints. The following table lists the valid audit control attributes for OS/400:

Attribute	OS/400
Warning Mode	No
Resource Access Successes	Yes
Resource Access Failures	Yes

Setting TCP Service Access Permissions

OS/400 does not support setting Transmission Control Protocol (TCP) service access permissions.

Creating Role Records

One of the primary functions of a role record is to assign resources, as defined by Tivoli Security Management resource records, required by the job function represented by the role. There is no limit on the number of resource record members a role can contain. A role can contain member resources of multiple resource types, such as **FILE**, **COMMAND**, and **PROGRAM**.

Resource access rights are managed as a set of capability entries for each role. Each entry consists of the following:

- Resource record name
- Access permissions to the resources defined in the resource record
- A list of groups that are role members

In many cases, a user can have several access permissions to a resource. This is because permissions can be assigned from resource records as well as role records. For more information and procedures to perform the following operations for role records, refer to the *Tivoli Security Management User's Guide*:

- Defining role records
- Defining resource lists
- Assigning roles to a group record
- Setting resource type access rights
- Assigning parent resource roles
- Assigning resource roles to group records

AS/400-specific information for role records is described in the following sections.

Setting Resource Access Rights

You can set default permissions for a specified resource by setting resource access rights. Tivoli Security Management applies these permissions when a user assigned to the current role accesses a resource that is defined in the same role record. For OS/400 subscribers, you can set standard access permissions or OS/400 authorities for all AS/400 resources in ***ALLUSR** libraries.

Setting OS/400 Security Audit Report Tasks

Tivoli Security Management provides task libraries that enable you to generate audit reports. You can use these reports to track individual login requests or patterns of login attempts and resources accesses.

Setting OS/400 Endpoint Audit Log Report

You can use the OS/400 Endpoint Audit Log Report to retrieve selected OS/400 audit journal entries from the OS/400 Security audit journal, **QAUDJRN**. As data is stored in the OS/400 Security audit journal, you can retrieve information about users who successfully logged in and out of selected AS/400 systems, and about users who encountered failures when attempting to login. You can also retrieve information about which users are successfully, or unsuccessfully, accessing particular resources on AS/400 systems.

When you request resource accesses to be returned, information on both the data access and object management of the resource is returned, including the creation, deletion, move, restore, auditing changes, and authority changes of the object.

The output of the task specifies the following data:

- Date and time of the event
- Endpoint system where the event occurred
- What type of event the audit journal entry represents
- Name of the resource accessed or the display device from where the login was attempted
- Information about the user associated with the entry

The following are the supported audit event types for resources:

- LOGIN
- LOGOUT
- READ

- CHANGE
- MOVE
- DELETE
- RESTORE
- CREATE
- UPDATE

Additionally, the following codes are displayed, depending on whether the event type was a success or failure:

P Specifies that a resource was successfully accessed or that a user successfully logged into the OS/400 endpoint system.

A Specifies that a user unsuccessfully attempted to gain access to a resource or that the user unsuccessfully attempted to login to the OS/400 endpoint system.

O Specifies that a user logged out from an OS/400 endpoint system.

For more information about running an endpoint audit report job, see the *Tivoli Security Management User's Guide*.

Creating Audit Log Report Filters

The audit report filters enable you to restrict which OS/400 audit journal entries are returned. The following options are available to specify the type of audit events that Security Management will display:

Report All Radio Button

Specifies that login, logout, and resource accesses for all objects of all supported types is returned.

Report Successes Radio Button

Specifies that successful login or successful resource accesses, or both are returned.

Report Failures Radio Button

Specifies that failed login or failed resource accesses, or both are returned.

Who and What to Report Button

Limits the audit report to the following specific users or events:

User Name
Specifies the name of the user that information is to be returned for. The name must be a valid OS/400 user profile name. If you wish to retrieve information corresponding to all users, specify a * wildcard.

Logins
Specifies that user login information is returned. The audit journal entries that are returned are for the user specified in the **User Name** field.

Report Logouts
Specifies that information about user logouts returned. The audit journal entries that are returned are for the user specified in the **User Name** field. Logout information is only returned when the **Report successes** check box is checked.

Resource Name
Specifies the name of an OS/400 resource for which an OS/400 audit journal entry is created. You can retrieve all or a specific resource.

If all resources are retrieved, audit journal entries associated with any OS/400 resource that is of a supported resource type are returned. Specify all resources with a * wildcard. If a specific resource is retrieved, then You must enter a resource name in OS/400 IFS format.

For example, to retrieve audit journal entries for program MYPGM in library MYLIB, enter the following path:
QSYS.LIB/MYLIB.LIB/MYPGM.PGM

If you want to retrieve information about a system value or a user profile, the name of the system value or user profile can be specified. There is no need to specify it in IFS format.

The resource type, as described below, is ignored when a specific resource is specified.

Resource Type

Specifies an OS/400 resource type. The resource types available correspond to those that the Tivoli Security Management product allows you to manage. This value is ignored when a value other than a * wildcard is entered for the **Resource Name**.

Date and Time Button

Specifies the date and time range that you wish to have audit journal entries returned from. You can specify absolute dates, such as January 1, 1998 though January 31, 1998, or you can specify relative dates, such as 5 days ago through 2 days ago. If no date or time is specified, then all audit journal entries in the current chain of OS/400 journal receivers are returned.

Where to Run Report List

Specifies an OS/400 resource type. The resource types available correspond to those that the Tivoli Security Management product allows you to manage. This value is ignored when a value other than a wildcard (*) is provided for the **Resource Name**.

Endpoint Timeout

Specifies the amount of time that the endpoint task should be allowed to run on the target AS/400s. Because the target AS/400s can contain many audit journal records, you might want to increase this value to allow for complete processing of all OS/400 journal records.

Executing Audit Report Endpoint Tasks

The Audit Report Endpoint task is executed by the OS/400 Endpoint Audit Log Report task and must not run interactively.

Specifying Security Database Report

When specifying OS/400 resource names, they must be in the following form:

```
OS4:resource_type:resource_name
```

where *resource_type* is any resource type that Tivoli Security Management can manage for the AS/400 and *resource_name* specifies the name of the resource.

Modifying OS/400 Endpoint Security

You can use a number of Tivoli command line commands to create and modify records in a security profile. In particular, the **wcrtsec** and **wmodsec** commands enable you to create and update security profiles. Although commands provided by Tivoli Security Management work similarly on AS/400 systems, there are some differences that you must consider for AS/400 endpoints. These differences are summarized in the following table.

CLI	OS/400 Changes
wcpsec	Endpoint resource type argument must be specified as **OS4**.
wcrtsec	See description for the **wcrtsec** command that follows this table.
wdelsec	This command will not delete any IBM supplied user profiles. These profiles include **QSECOFR, QPGMR, QUSER,** and so on.
wmodsec	See description for the **wmodsec** command that follows this table.
wmvsec	Endpoint resource type argument must be specified as **OS4**.

CLI	OS/400 Changes
wpopsec	For OS/400, Tivoli Security Management populates all existing groups or populates requested group name. Some values cannot be populated. These values include login times and access times. On OS/400, resources are populated one at a time. Wildcards and populate "all" are not supported.

The following examples in this section describe how to use the **wcrtsec** and **wmodsec** commands to create a security profile and modify security profile attributes after it is created. For additional information on these commands, refer to Appendix A, "Commands" on page A-1.

Refer to the *Tivoli Security Management User's Guide* for more information about Tivoli Security Management and commands.

To create the **AS400** system policy record in the **SecProf** security profile, enter the following command at the command line:

```
wcrtsec SystemPolicy -s LoginAudit=Failure -s PwChk=True
        -s PwHistory=5 SecProf AS400
```

where:

SystemPolicy Indicates the record is a system policy record.

–s LoginAudit=Failure
Indicates that login failures are audited.

–s PwChk=True
Indicates password rule checking is enabled. Password history is one type of password rule.

–s PwHistory=5
Indicates the user cannot re-use a password until 5 other passwords have been specified.

SecProf Specifies the security profile to which the record is added.

AS400 Specifies the name of the record.

To add members of the **Marketing** group to the security profile, enter the following command from the command line:

```
wmodsec Group –a OS4UserMembers='hhenry, jgigot'
        SecProf Marketing
```

where:

Group Indicates the record is a group record.

-a OS4UserMembers='jhuey, psmith'
Adds values for the **OS4UserMembers** attribute.

SecProf Specifies the security profile to which the record is added.

Marketing Specifies the name of the group record.

5

Tivoli Distributed Monitoring for AS/400 Endpoints

Tivoli Distributed Monitoring provides availability management, monitors system and application resources, and generates events and alarms on a network-wide basis (based on the monitoring data). Tivoli Distributed Monitoring functions across a variety of operating systems and platforms within a distributed environment, including AS/400 endpoints on OS/400 systems.

To manage the performance of AS/400 systems in the TMR, Tivoli provides a number of AS/400 monitoring collections. The AS/400 monitoring collections provide the same monitoring capabilities as the AS/400 Performance Monitor application. They also provide the same functions as the AS/400 Work/Management APIs that are integrated with the OS/400 operating system.

Each AS/400 monitoring collection contains several monitoring sources that allow you to manage different aspects of your network, such as system resources and network protocols.

Using Tivoli Distributed Monitoring with the AS/400 monitoring collections enables you to manage many AS/400 endpoints from a central location.

Note: The communication probes for OS/400 Distributed Monitoring return statistic about only the SNA traffic over the requested line.

You can distribute a Tivoli Distributed Monitoring profile so that all AS/400 endpoints monitor the same internal resources and make the same corrective actions if a specified threshold is reached. In addition, updating the monitors as your network changes is simpler than individually updating the AS/400 Performance Monitor on each AS/400 system. After adding a new monitor to a Tivoli Distributed Monitoring profile or editing an existing monitor, you can distribute the profile.

The AS/400 monitoring collections can be run simultaneously with the AS/400 Performance Monitor. It is also possible to simultaneously monitor a resource with an AS/400 monitoring collection and the AS/400 Performance Monitor.

The AS/400 monitoring collections comprise two categories:

- Core AS/400 monitoring collections to monitor AS/400 system resources and objects
- Network protocol monitoring collections to monitor the availability and use of standard network protocols

The core AS/400 monitoring collections are as follows:

- OS/400 System
- OS/400 Subsystem
- OS/400 Job
- OS/400 Pool
- OS/400 Disk
- OS/400 Object
- OS/400 IOP

The network protocol monitoring collections are as follows:

- OS/400 Async
- OS/400 Bisync
- OS/400 Token Ring
- OS/400 Ethernet
- OS/400 IDLC
- OS/400 ISDN

- OS/400 SDLC
- OS/400 X25

The following sections describe each AS/400 monitoring collection, the resources monitored by each collection, and any required input for each monitor. To add a monitor from the Tivoli Distributed Monitoring CLI, use the **waddmon** command and specify the monitor name.

See the *Tivoli Distributed Monitoring User's Guide* and the *Tivoli Distributed Monitoring Collection Reference* for information about installing and using monitoring collections to manage system and application resources.

OS/400 System Monitoring Collection

The following table lists the monitoring sources for the OS/400 System monitoring collection:

Resource Monitored	Description	Name
Number of jobs	Number of jobs running on the AS/400. Includes all jobs regardless of status.	NumberofJobs
Percent CPU time	Percentage of CPU time used by all jobs.	CPUTimePct
Numeric script		ncustom
String script		scustom

OS/400 Subsystem Monitoring Collection

The following table lists the monitoring sources for the OS/400 Subsystem monitoring collection:

Resource Monitored	Description	Input	Name
Number of jobs	Number of jobs active in this subsystem.	Subsystem name (char 10) Library name (char 10)	NumberofJobs
Subsystem status	Indicates whether or not the subsystem is active.	Subsystem name (char 10) Library name (char 10)	Status

OS/400 Job Monitoring Collection

The following table lists the monitoring sources for the OS/400 Job monitoring collection. All OS/400 job monitoring resources require the following input:

- Job name (char 10)
- Job user (char 10)
- Job number (char 6)

Resource Monitored	Description	Return Values	Name
Job existence	Indicates whether or not the job exists.	**Yes** or **No**	Existence
Job status	Status of an active job.	**HLD**, **DEQW**, **MSGW**, and so on.	ActiveStatus

Resource Monitored	Description	Return Values	Name
Job type	Type of job or task.	**A** autostart **B** batch **H** horizontal licensed internal code **I** interactive **M** subsystem monitor **R** spooled reader **S** system **V** licensed internal code **W** spooled writer **X** start control program function system	Type

Resource Monitored	Description	Return Values	Name
Job subtype	Subtype of the job.	**N** no special subtype **D** immediate **E** evoke job or batch communicate **J** prestart job **P** print driver job **T** multiple requests terminal (MRT) job (System/36 environment only) **U** alternate spool user	Subtype
Job priority	Priority over other jobs.	Priority number	Priority
Job pool	Pool in which the job is running.	Pool name	Pool
Time slice	Time slice value.	Seconds	TimeSlice
CPU time	Processing unit used by this job.	Milliseconds	CPUTime
Function	Function currently performed by this job.	Function name as shown in the function column of **WRKACTJOB** (for example, **PGM** for pgmname and **CMD** for command name)	Function

Resource Monitored	Description	Return Values	Name
Function type	Type of function performed by this job.	**U** not logged **C** command **D** delayed job **G** transfer group job **I** index path rebuild **L** logging to **QHST** **M MRT** job **N** menu **O** I/O **R** procedure ***** special function (you must also request the function name to receive additional information)	Function Type
Transaction count	Number of transactions performed by this job.	Number of transactions	TransactionCount
Transaction time	Total transaction time.	Seconds	TransactionTime
Synchronous database reads	Total number of physical synchronous reads for database operations.	Number of reads	SyncDBReads
Synchronous database writes	Total number of physical synchronous writes for database operations.	Number of writes	SyncDBWrites

Resource Monitored	Description	Return Values	Name
Synchronous non-database reads	Total number of physical synchronous reads for non-database operations.	Number of reads	SyncnonDBReads
Synchronous non-database writes	Total number of physical synchronous writes for non-database operations.	Number of writes	SyncnonDBWrites
Asynchronous database reads	Total number of physical asynchronous reads for database operations.	Number of reads	AsyncDBReads
Asynchronous database writes	Total number of physical asynchronous writes for database operations.	Number of writes	AsyncDBWrites
Asynchronous non-database reads	Total number of physical asynchronous reads for non-database operations.	Number of reads	AsyncnonDBReads
Asynchronous non-database writes	Total number of physical asynchronous writes for non-database operations.	Number of writes	AsyncnonDBWrite
Communication puts	Number of communication writes. Includes the activity related to intersystem communication function (ICF) files when the I/O is for an ICF device. Does not include remote workstation activity.	Number of writes	CommPuts

Resource Monitored	Description	Return Values	Name
Communication gets	Number of communication writes. Includes activity related to intersystem communication function (ICF) files when the I/O is for an ICF device. Does not include remote workstation activity.	Number of reads	CommGets
Binary overflows	Number of binary overflows.	Number of overflows	BinOverflows
Decimal overflows	Number of decimal overflows.	Number of overflows	DecOverflows
Floating point overflows	Number of floating point overflows.	Number of overflows	FPOverflows
Logical database reads	Count of calls to the internal database read functions. Does not include I/O operations to readers or writers or operations caused by the **Copy Spool File (CPYSPLF)** or **Display Spooled File (DSPSPLF)** command. If **SEQONLY(*YES)** is specified on the **Override with Database File (OVRDBF)** command, this number is the blocks of records read, not the number of records read.	Number of records or blocks read	LogicalDBReads

Resource Monitored	Description	Return Values	Name
Logical database writes	Count of calls to the internal database write functions. Does not include I/O operations to readers or writers or operations caused by the **Copy Spool File (CPYSPLF)** or **Display Spooled File (DSPSPLF)** command. If **SEQONLY(*YES)** is specified on the **Override with Database File (OVRDBF)** command, this number is the blocks of records read, not the number of records read.	Number of records or blocks written	LogicalDBWrites
Miscellaneous database operations	Number of update, delete, forced-end of data, and release operations.	Number of operations	MiscDBOpts
Permanent writes	Number of permanent writes.	Number of writes	PermanentWrites
PAG faults	Number of times the program access group (PAG) was referred to while it was not in main storage.	Number of times	PAGFaults
Number of print lines	Number of lines written by the job. This does not reflect what was actually printed. Spool files can be ended or printed with multiple copies.	Number of lines	NumPrintLines

Resource Monitored	Description	Return Values	Name
Number of print pages	Number of print pages written by the job. This does not reflect what was actually printed. Spooled files can be ended or printed with multiple copies.	Number of pages	NumPrintPages
Active to wait transactions	Number of transitions changed from active to wait state.	Number of transitions	ActiveToWait
Wait to ineligible transitions	Number of transitions changed from wait to ineligible status.	Number of transitions	WaitToIneligible
Active to ineligible transitions	Number of transitions changed from active to ineligible status.	Number of transitions	ActiveToIneligible

OS/400 Pool Monitoring Collection

The following table lists the monitoring sources for the OS/400 Pool monitoring collection:

Resource Monitored	Description	Input	Name
Activity Level	Maximum number of processes that can be active in the machine at the same time.	Pool number	ActivityLevel
Pool size	Amount of main storage (in KB) that is dedicated to machine functions.	Pool number	Size

Resource Monitored	Description	Input	Name
Machine-reserved portion	Amount of storage (in KB) from the pool that is dedicated to machine operations	Pool number	MachinePortion
Database faults	Total number of interruptions to processes (not necessarily assigned to this pool) that were required to transfer data into the pool to permit the machine interface (or MI) instructions to process this database function	Pool number	DBFaults
Non-database faults	Total number of interruptions to processes (not necessarily assigned to this pool) that were required to transfer data into the pool to permit the machine interface (or MI) instructions to process this non-database function.	Pool number	NonDBFaults
Database pages	Total number of pages of database data transferred from auxiliary storage to the pool to permit the instruction to run as a consequence of set access state, implicit access group movement, and internal machine actions.	Pool number	DBPages
Non-database pages	Total number of pages of non-database data transferred from auxiliary storage to the pool to permit the instruction to run as a consequence of set access state, implicit access group movement, and internal machine actions.	Pool number	NonDBPages

Resource Monitored	Description	Input	Name
Active-to-wait transitions	Total number of transitions by processes assigned to this pool from active state to wait state.	Pool number	ActiveToWait
Wait-to-ineligible transitions	Total number of transitions by processes assigned to this pool from wait state to ineligible state.	Pool number	WaitToIneligible
Active-to-ineligible transitions	Total number of transitions by processes assigned to this pool from active to ineligible state.	Pool number	ActiveToIneligibl e

OS/400 Disk Monitoring Collection

The following table lists the monitoring sources for the OS/400 Disk monitoring collection:

Resource Monitored	Description	Input	Name
System ASP used	Storage capacity (in MB) of the system auxiliary storage pool (ASP1).	N/A	SystemASPUsed
Percent system ASP used	Percentage (in ten-thousandths) of the system storage pool currently in use. (For example, 41123 would be reported as 4.1123.)	N/A	SystemASPUsedPct
Unprotected storage used	Amount of storage in use (in MB) for temporary objects.	N/A	MaxUnprotectedUsed

Resource Monitored	Description	Input	Name
Maximum unprotected storage used	Largest amount of storage used (in MB) for temporary objects at any time since the last IPL.	N/A	MaxUnprotectedUsed
Mirror status	Status of disk mirroring. Possible values are: **1** Active **2** Resuming **3** Suspended	Disk arm number (char 4)	MirrorStatus
Drive capacity	Number of KB of auxiliary storage provided on unit for the storage of objects and internal machine functions. The unit reserved is subtracted from the unit capacity to calculate this value.	KB of storage	Capacity
Percent drive available	Percentage of auxiliary storage space not currently assigned to objects or internal machine functions.	Percentage of storage	DriveAvailablePct
Blocks read per second	Number of blocks read. The block length is 520 bytes, which includes 8 bytes of system control information.	Disk arm number (char 4)	BlocksReadPerSec
Blocks written per second	Number of blocks written. The block length is 520 bytes, which includes 8 bytes of system control information.	Disk arm number (char 4)	BlocksWrittenPerSec
Writes per second	Number of writes per second.	Disk arm number (char 4)	WritesPerSec

Resource Monitored	Description	Input	Name
Reads per second	Number of reads per second.	Disk arm number (char 4)	ReadsPerSec
Processor idle loop count	Number of times the disk controller passes through the idle loop. The count increases differently for the 9332 and 9335 disk drives. For the 9332, this counter is increased only if the disk controller is totally idle (no I/O operations are active). FOr the 9335, event though the disk controller may be idle and the counter is increased, an I/O operation can be active (for example, seek is being performed). This field in non-zero for drive types that support a dedicated disk processor and is zero for other drive types.	Disk arm number (char 4)	IdleLoopCount
Processor idle loop time	Time (in hundredths of microseconds) to complete one pass through the idle loop. This field is non-zero for drive types that support a dedicated disk processor, and is zero for other drive types.	Disk arm number (char 4)	IdleLoopTime
Percent seeks great than 2/3	Percentage the disk arm traveled more than two-thirds of the disk on a seek request.	Disk arm number (char 4)	PctSeeksOverTwoThirds

Resource Monitored	Description	Input	Name
Percent seeks 1/3 to 2/3	Percentage the disk arm traveled more than one-third of the disk but less than two-thirds of the disk on a seek request.	Disk arm number (char 4)	PctSeeksThirdToTwoThirds
Percent seeks 1/6 to 1/3	Percentage the disk arm traveled more than one-sixth of the disk but less than one-third of the disk on a seek request.	Disk arm number (char 4)	PctSeeksSixthToThird
Percent seeks 1/12 to 1/6	Percentage the disk arm traveled more than one-twelfth of the disk but less than one-sixth of the disk on a seek request.	Disk arm number (char 4)	PctSeeksTwelfthtoSixth
Percent seeks less than 1/12	Percentage the disk arm traveled less than one-twelfth of the disk on a seek request.	Disk arm number (char 4)	PctSeeksUnderTwelfth
Percent zero seeks	Percentage the disk arm did not move on a seek request.	Disk arm number (char 4)	PctSeeksZero
Buffer overruns	Number of times data was available to be read into the disk controller buffer from the disk, but the disk controller buffer contained data that was not retrieved by the storage device controller. The disk had to make an additional revolution until the buffer was available to accept data.	Disk arm number (char 4)	BufferOverruns

Resource Monitored	Description	Input	Name
Buffer underruns	Number of times the disk controller was ready to transfer data to the disk on a write operation, but the disk controller buffer was empty. The data was not transferred in time by the disk IOP to this buffer. The disk had to make an additional revolution awaiting the data.	Disk arm number (char 4)	BufferUnderruns
Average queue length	Average number of I/O operation waiting for service. This value includes the I/O operation currently in progress.	Disk arm number (char 4)	AvgQueueLength
ASP number	Auxiliary storage pool to which this unit is currently allocated. Possible values are: **0** not allocated **1** allocated to system ASP **2** to **16** allocated to a user ASP	Disk arm number (char 4)	ASPNumber

OS/400 Object Monitoring Collection

The following table lists the monitoring sources for the OS/400 Object monitoring collection:

Resource Monitored	Description	Input	Name
Object Size	Size (in bytes) of the object.	Qualified object name: ■ object name (char 10) ■ object lib (char 10) ■ object type (char 10)	ObjectSize
Object Owner	AS/400 user profile that owns the object	Qualified object name: ■ object name (char 10) ■ object lib (char 10) ■ object type (char 10)	ObjectOwner

OS/400 IOP Monitoring Collection

The following table lists the required input parameters for the OS/400 IOP monitoring collection:

AS/400 Hardware Release or Type	Required Input Parameters
V3R2 - CISC systems	Bus number (integer), IOP address (integer)
V3R7 - RISC systems	IOP resource name (char 10)

The following table lists the monitoring sources for the OS/400 IOP monitoring collection:

Resource Monitored	Description	Name
Idle loop count	Number of times the IOP ran an idle loop. This occurs when the IOP has no work to perform.	IdleLoopCount
Idle loop time	Time (in hundredths of a millisecond) to run the idle loop once.	IdleLoopTime
RAM utilization	Number of bytes of free local storage in the IOP. The free local storage will probably be noncontiguous due to fragmentation. This monitors applies only to communications and multifunction IOPs; a value of **0** is returned for other types of IOPs.	RAMUtlization
IOP System Function Time	Total time (in milliseconds) used by the IOP for basic system functions. This monitor applies only to communications and multifunction IOPs; a value of **0** is returned for other types of IOPs.	SystemFunctionTime

Resource Monitored	Description	Name
Communications time (all protocols)	Total processing time (in milliseconds) used by all of the communications protocols. This monitor applies only to communications and multifunction IOPs; a value of **0** is returned for other types of IOPs.	CommTime
Percent SDLC time	Percentage of time used by SDLC tasks. This monitor applies only to communications and multifunction IOPs; a value of **0** is returned for other types of IOPs.	SDLCPct
Percent Asynchronous time	Percentage of time used by asynchronous tasks. This monitor applies only to communications and multifunction IOPs; a value of **0** is returned for other types of IOPs.	AsyncPct
Percent Bisynchronous time	Percentage of time used by bisynchronous tasks. This monitor applies only to communications and multifunction IOPs; a value of **0** is returned for other types of IOPs.	BisyncPct

Resource Monitored	Description	Name
Percent X25 LLC time	Percentage of time used by X.25 logical link control tasks. This monitor applies only to communications and multifunction IOPs; a value of **0** is returned for other types of IOPs.	XTwoFiveLLCPct
Percent X25 PLC time	Percentage of time used by X.25 packet layer tasks. This monitor applies only to communications and multifunction IOPs; a value of **0** is returned for other types of IOPs.	XTwoFivePLCPct
Percent X25 DLC time	Percentage of time used by X.25 data link control tasks. This monitor applies only to communications and multifunction IOPs; a value of **0** is returned for other types of IOPs.	XTwoFiveDLCPct
Percent LAN time	Percentage of time used by Token Ring and Ethernet tasks. This monitor applies only to communications and multifunction IOPs; a value of **0** is returned for other types of IOPs.	LANPct

Resource Monitored	Description	Name
Percent SDLC SMH time	Percentage of time used by SDLC short-hold mode tasks. This monitor applies only to communications and multifunction IOPs; a value of **0** is returned for other types of IOPs.	SDLCSHMPct
Percent ISDN LAPE/LAPD time	Percentage of time used by integrated services digital network tasks. This monitor applies only to communications and multi-function IOPs; a value of **0** is returned for other types of IOPs. The ISDN communications tasks are **LAPD** (Link access procedure D-channel) and **LAPE** (enhanced version of **LAPD**).	ISDNLAPPct
Percent ISDN Q931 time	Percentage of time used by ISDN Q.931 tasks. This monitor applies only to communications and multifunction IOPs; a value of **0** is returned for other types of IOPs.	ISDNQNineThreeOnePct
Disk Time	Total processing time (in milliseconds) used by disk tasks. This monitor applies only to communications and multifunction IOPs; a value of **0** is returned for other types of IOPs.	DiskTime

OS/400 Async Monitoring Collection

The following table lists the monitoring sources for the OS/400 Async monitoring collection.

Note: Line description (char 10) is the input for each resource monitored.

Resource Monitored	Description	Name
Line Status	Status of the line. The possible values are: **0** The line is not active **1** The line is active	LineStatus
Active Time	Amount of time (in seconds) that the line was active.	ActiveTime

OS/400 Bisync Monitoring Collection

The following table lists the monitoring sources for the OS/400 Bisync monitoring collection.

Note: Line description (char 10) is the input for each line status.

Line Status	Description	Name
Active Time	Status of the line. The possible values are: **0** The line is not active **1** The line is active	LineStatus
Line Status	Amount of time (in seconds) that the line was active (varied on).	ActiveTime

OS/400 Ethernet Monitoring Collection

The following table lists the monitoring sources for the OS/400 Ethernet monitoring collection.

Note: Line description (char 10) is the input for each resource monitored.

Resource Monitored	Description	Name
Line Status	Status of the line. The possible values are: **0** The line is not active **1** The line is active	LineStatus
Active Time	Amount of time (in seconds) that the line was active.	ActiveTime
SABME frames sent	Number of set asynchronous balanced mode extended frames sent.	SABMEFramesSent
SABME frames received	Number of set asynchronous balanced mode extended frames received.	SABMEFramesRcv
Mail Frames Discarded	Number of store and forward mail products dropped.	MailFramesDiscarded

OS/400 Token Ring Monitoring Collection

The following table lists the monitoring sources for the OS/400 Token Ring monitoring collection.

Note: Line description (char 10) is the input for each resource monitored.

Resource Monitored	Description	Name
Line Status	Status of the line. The possible values are: **0** The line is not active **1** The line is active	LineStatus
Active Time	Amount of time (in seconds) that the line was active.	ActiveTime
SABME frames sent	Number of set asynchronous balanced mode extended frames sent.	SABMEFramesSent
SABME frames received	Number of set asynchronous balanced mode extended frames received.	SABMEFrameRcv
Percent routing I-frames sent	Percentage of frames (logical ling control and medium access control) with routing information sent.	RoutingIframesSentPct
Percent routing I-frames received	Percentage of frames (logical link control and medium access control) with routing information received.	RoutingIFramesRcvPct
Percent ring utilization	Percentage of time the ring was in use. For half-duplex lines, this is line utilization. For full-duplex lines, is in-line and out-line utilization.	RingUtilizationPct

OS/400 IDLC Monitoring Collection

The following table lists the monitoring sources for the OS/400 IDLC monitoring collection.

Note: Line description (char 10) is the input for each resource monitored.

Resource Monitored	Description	Name
Line Status	Status of the line. The possible values are: **0** The line is not active **1** The line is active	LineStatus
Active Time	Amount of time (in seconds) that the line was active.	ActiveTime

OS/400 ISDN Monitoring Collection

The following table lists the monitoring sources for the OS/400 ISDN monitoring collection.

Note: Line description (char 10) is the input for each resource monitored.

Resource Monitored	Description	Name
Line Status	Status of the line. The possible values are: **0** The line is not active **1** The line is active	LineStatus
Active Time	Amount of time (in seconds) that the line was active.	ActiveTime

Resource Monitored	Description	Name
Errored seconds	Number of seconds that had at least one DTSE-in or DTSE-out error. DTSE is a maintenance channel that measures the quality of the line.	ErrorSeconds
Severely errored seconds	Number of seconds that had more than three DTSE-in and DTSE-out errors. DTSE is a maintenance channel that measures the quality of the line.	SeveErroredSecond s

OS/400 SDLC Monitoring Collection

The following table lists the monitoring sources for the OS/400 SDLC monitoring collection.

Note: Line description (char 10) is the input for each resource monitored.

Resource Monitored	Description	Name
Line Status	Status of the line. The possible values are: **0** The line is not active **1** The line is active	LineStatus
Active Time	Amount of time (in seconds) that the line was active.	ActiveTime
Polling Wait Time	Length of time (in tenths of a second) that the system waited for a response to a poll while in normal disconnect mode before polling the next station.	PollWaitTime

OS/400 X25 Monitoring Collection

The following table lists the monitoring sources for the OS/400 X25 monitoring collection.

Note: Line description (char 10) is the input for each resource monitored.

Resource Monitored	Description	Name
Line Status	Status of the line. The possible values are: **0** The line is not active **1** The line is active	LineStatus
Active Time	Amount of time (in seconds) that the line was active (varied on).	ActiveTime

Application Response Management for AS/400

Tivoli Distributed Monitoring Application Response Management (ARM) agent for AS/400 endpoints provides the ARM API to monitor availability and performance transactions on OS/400 systems. To use the ARM API, you must create the application and call this API as described in the *Tivoli Distributed Monitoring ARM Agents User’s Guide, Version 3.6*.

The following procedure briefly describes how to create and use this application:

1. Install the ARM patch on your endpoint gateway using Tivoli Software Installation Service (SIS), the Tivoli desktop, or the **wpatch** command.

2. From the ARM directory, install the binary files to the AS/400 endpoint from the gateway. Run the following command from the TMR server command line:

   ```
   cd $BINDIR/../arm_bundle/bin/generic/TME/ARM

   arminst.ep as400endpoint.mydom.com
   $BINDIR/../arm_bundle
   ```

3. Start the ARM agent with the following command:

   ```
   warmcmd srvstart -f /tmp/myarm.log
   ```

4. Start the ARM client on the AS/400 endpoint with either the following command:

   ```
   CALL QTMELCFDM/WARMCMD (‘clntstart’ ‘-h’ ‘server_URL’)
   ```

   or from the task library provided with ARM agent.

5. Check the status of ARM agent by running the following command on the endpoint:

   ```
   CALL QTMELCFDM/WARMCMD (‘clntstatus’)
   ```

6. Bind the AS/400 application with the ARM service program. The *Tivoli Distributed Monitoring ARM Agents User’s Guide* describes the location of the header file in the source directory. You can use the arm.h file in the application program when using the ILE/C compiler in AS/400. When using this header file, you must move it to your development AS/400. You can also use the VisualAge C++ cross-compiler for AS/400.

   To bind the program, you must bind it to the ARM service program, as in the following example:

   ```
   CRTPGM PGM(MYLIB/MYPGM)BNDSRVPGM(QTMELCFDM/LIBARM)
   ```

7. Start the collections. Tivoli recommends that you start the collection using the callable program from the AS/400, for example:

   ```
   CALL QTMELCFDM/WARMMD (‘startcoll’ ‘-a’ ‘ARM_sample’
   ’-t’ ‘Simple_transaction_1’ ‘-u’ ‘*’ ‘-i’ ‘240’ -’-f’)
   ```

8. Run the application that uses the ARM APIs, for example:

   ```
   CALL MYLIB/MYPGM
   ```

To stop the AS/400 ARM client, enter the following command from the client:

```
CALL QTMELCFDM/WARMCMD (‘clntstop’)
```

To stop the AS/400 ARM server, enter the following command from the TMR server:

```
warmcmd srvstop
```

6

Tivoli Inventory for AS/400 Endpoints

Tivoli Inventory is a hardware and software inventory-gathering application designed to help system administrators and accounting personnel manage the complexity of a distributed client/server enterprise. Specifically, Tivoli Inventory enables users to do the following:

- Regularly maintain and upgrade hardware and software
- Monitor and record changes in software and hardware configurations
- Manage all of the enterprise's systems from a central point
- Access inventory information to perform system auditing functions

Tivoli Inventory for AS/400 enables you to scan inventory information on AS/400 machines and to use this data to manage your AS/400 systems. All information is stored in the ***configuration repository***, which is a relational database management system (***RDBMS***) that holds information about system configurations in your Tivoli environment. Tivoli Inventory provides the scripts necessary to create the configuration repository as well as to install the configuration repository schema.

All procedures to scan and monitor software and hardware information on machines in the TMR are outlined in the *Tivoli Inventory User’s Guide*. Using these procedures, you can gather information about installed system hardware, installed AS/400 software products, and about AS/400 Program Temporary Fixes (PTFs) or patches.

Scanning for AS/400 Inventory

Tivoli Inventory for AS/400 stores and updates hardware and software information in the following tables in the configuration repository.

Scanning Hardware Inventory

Tivoli Inventory for AS/400 collects the following information about AS/400 hardware:

Information	Configuration Repository Tables
Mass storage	CDROM_DRIVE, HARDDISK INSTALLED_CDROM_DRIVE INSTALLED_HARDDISK INSTALLED_PRINTER INSTALLED_TAPE_DRIVE INSTALLED_FLOPPYDRIVE FLOPPYDRIVE TAPE_DRIVE PRINTER
Memory	COMPUTER_SYSTEM_MEMORY
Network interface	NETWORK_NODE
Processor	INSTALLED_PROCESSOR INSTALLED_COPROCESSOR
System	COMPUTER_SYSTEM MANUFACTURER

Scanning Software Inventory

Inventory collects the following information about AS/400 software products:

Information	Configuration Repository Tables
Software	INSTALLED_OS_DETAILS

Extending the Configuration Repository Schema for AS/400 PTFs

All procedures for installing the configuration repository and database scripts are documented in the *Tivoli Inventory User's Guide*.

After installing the configuration repository schema, complete the procedure to install AS/400 PTF extensions as provided in the *Tivoli Inventory User's Guide*. Select the correct schema for your database from the following table when completing the installation procedure:

Database Type	Database Schema
DB/2	tivoli_db2_schema_as400_ptf.sql
Informix	tivoli_inf_schema_as400_ptf.sql
Microsoft SQL Server	tivoli_ms_sql_schema_as400_ptf.sql
Oracle	tivoli_ora_schema_as400_ptf.sql
Sybase	tivoli_syb_schema_as400_ptf.sql

Note: You must copy the appropriate script from the **$BINDIR/TME/INVENTORY/SCRIPTS/RDBMS** directory on the TMR server to a temporary directory on the RDBMS server.

Scanning for AS/400 PTFs

AS/400 Program Temporary Fixes (PTFs) are supported through a user-defined MIF file in Tivoli Inventory. To scan for AS/400 PTF Inventory information, you must first extend your configuration repository by running the scripts supplied. You must also create and customize an Inventory Profile that will run a program from a REXX script or a batch job and read the results of the custom MIF file named **ptf.mif**.

To customize an Inventory profile for AS/400 PTFs, complete the following steps from the Tivoli desktop:

1. Extend your configuration repository by running the **tivoli_***dbtype***_schema_as400_ptf.sql** script located in **$BINDIR/TME/INVENTORY/SCRIPTS/RDBMS** directory on the TMR server.
2. Create an Inventory profile as described in the *Tivoli Inventory User's Guide*.
3. Right click on the this profile to select the **Customize** option.
4. Add a script to the profile, as follows:
    a. Under the **Execute at Target** window, select the **Script** option.
    b. Click the **…** button to open the **Edit Target Script** dialog.
    c. Enter the following:

    ```
    "CALL PGM(QLCFCACHE/OS4_PTFDIS)
          PARM (prodid1 prodid2 … prodidn)"
    ```

    where:

    prodid1prodid2 … prodidn
    Specifies the AS/400 product identifier numbers, such as 5763CM1 or 5763DB1

    Note: You must include the **CALL** command in double quotes ("").

    You can display your AS/400 product identifier numbers via the **GO LICPGM** menu on the AS/400. If the **PARM** argument is omitted, all software products installed will be scanned for PTFs.

For example, to scan for all PTFs for products 5763DB1 and 5763CM1, enter the following:

```
"CALL PGM(QLCFCACHE/OS4_PTFDIS) PARM(5763DB1 5763CM1)"
```

To scan for all PTFs enter:

```
"CALL PGM(QLCFCACHE/OS4_PTFDIS)"
```

5. To read the results of the script, choose **Custom MIF file** in the Read Results Panel.

6. Add **ptf.mif** to the **Custom MIF Files to be Read**.

For more detailed information about customizing Inventory profiles, refer to the *Tivoli Inventory User's Guide*.

When you customize Tivoli Inventory to scan for AS/400 PTF information, the following information is collected and stored in the specified tables:

Information	PTF Information
AS/400 PTF	PRODUCT_ID PTF_ID PTF_STATUS STATUS_DATE STATUS_TIME TYPE UNATTN_IPL_ACTION LANG_FEATURE IPL_SOURCE SYS_NAME ON_ORDER PTF_SAVE_FILE OPTIONAL_PART SUPERSEDING_PTF RELEASE TGT_OS400_RELEASE ACTION_PENDING ACTION_REQUIRED

To detect errors while running REXX scripts, add the following lines before and after your script call. Before your script call, add:

```
CALL ON ERROR
CALL ON FAILURE
```

After your script call, add:

```
exit 0
error:
exit 1
```

A successful script will return PTF MIF data and exit with a return code of 0. A script that encounters an error will return with a return code of 1.

For more information on errors encountered when reading or parsing PTF MIF files, refer to the endpoint error log.

Viewing and Querying OS/400 Information

In the Tivoli environment, a query library facility enables you to create, manage, and logically organize Tivoli queries. For example, you can create a query library to hold queries that selectively retrieve information about AS/400 software information. Tivoli provides the following customized **INVENTORY_OS400_SOFTWARE** query and the **INVENTORY_OS400_PTF** query to retrieve information about AS/400 inventory. It also provides the **PTF_VIEW** table in which updated PTF information is stored.

OS/400 Software Information

The **INVENTORY_OS400_SOFTWARE** query enables you to retrieve information about software that is installed on AS/400 endpoints. To install this query, run the **inventory_queries.sh** script that is provided with Tivoli Inventory. This script automatically installs a query library and all pre-defined queries. Refer to the *Tivoli Inventory User's Guide* for information about installing queries with the **inventory_queries.sh** script.

OS/400 PTF Information

Tivoli Inventory provides a customized view, **PTF_VIEW**, to view OS/400 PTF information. This view displays all of the data in the

PTF_INFORMATION table, and provides fields from the **COMPUTER_SYSTEM** table (or **COMPUTER_SYS** table for DB2 database), from **TME_OBJECT_LABEL**, and from **TME_OBJECT_ID**.

The **PTF_VIEW** is installed when you run the **tivoli_***dbtype***_schema_as400_ptf.sql** script to extend the configuration repository for all the PTF tables. Additionally, Tivoli provides the **INVENTORY_OS400_PTF** query to retrieve information from **PTF_VIEW**. To set up this query, you must install the **inventory_as400_queries.sh** script, as follows:

1. Copy the **inventory_as400_queries.sh** script from the **$BINDIR/TME/INVENTORY/SCRIPTS/QUERIES** directory on the TMR server to a temporary directory.
2. Run the **inventory_as400_queries.sh** script by entering the following command:

   ```
   inventory_as400_queries.sh region
   ```

   where *region* specifies the name of the policy region in which the query library will be created.

For more information about installing query scripts or about viewing and querying AS/400 information, see the *Tivoli Inventory Users's Guide* for more information about viewing inventory information.

Inventory Features Not Supported by AS/400

Tivoli Inventory for AS/400 does not support the following features:

- Only hardware, software, PTF, or script file customizations are supported by Tivoli Inventory for AS/400 (all GUI options). The following Tivoli commands are not supported for the AS/400 endpoint:
  - **wsetipdirs**
  - **wsetipext**

  In addition, users cannot specify which file directories or which file extensions to scan in the **Scan Directories** dialog.
- Collecting and saving **.CFG** files is not available on AS/400 systems, because AS/400 does not use **.CFG** files. Therefore, the

wsetipcf command is not supported for the AS/400. Users cannot specify the configuration files to save or how to save them in a **Save Configuration Files** window of the **Customize Inventory Profile** dialog.

- AS/400 software inventory does not include or support software signatures. As a result, the **wfilesig** command is not supported for the AS/400.
- AS/400 only supports REXX and batch job scripts for endpoint scripts.

7

Tivoli Software Distribution for AS/400 Endpoints

Tivoli Software Distribution provides an efficient method for distributing, installing, and controlling software across a network. Typically, an organization maintains many software applications and tools for its employees to use. Software Distribution is a centralized, software management system through which the administrator can install and configure new applications, update existing software, and synchronize software on distributed systems.

Tivoli Software Distribution for AS/400 enables you to create and distribute file packages to AS/400 endpoints. This chapter describes how Software Distribution uses configuration programs to perform a variety of operations during a file package distribution and provides procedures to define and set platform-specific file package options for OS/400 systems.

You must be familiar with the AS/400 convention of using *SAVF or save files to move programs from one system to another. *SAVF are similar to ZIP files in that they are packages of code and AS/400 objects. Prior to being used, *SAVF files must be restored on the systems to which they are distributed.

Configuration Programs

Tivoli Software Distribution uses configuration programs during a distribution to perform preprocessing or postprocessing operations, such as file manipulations or dependency checks, when a file package is installed. The format of available configuration programs depends on the platform type of the subscribing system. Available configuration programs for AS/400 subscribers are any type of executable programs including C programs, CL programs, REXX exec or batch job scripts. The CL or C programs are distributed in *SAVFs (AS/400 Save files). REXX executables and batch job scripts are in ASCII text format.

The following list specifies different types of configuration programs that are available for subscribers across platforms:

- Before programs
- After programs
- Removal programs
- After Removal programs
- Commit programs
- On Error programs

The before, after, (upon) removal, and commit programs run on the subscriber and are available using the Tivoli desktop, the command line, and the export and import capabilities. The source before and source after program options are only available using export and import or from the command line to set the appropriate keywords (**src_before**_*xxx* and **src_after**_*xxx*). Additionally, you cannot specify after removal or on error programs from the desktop. These configuration programs are available by setting the *yyy***_after_removal_***xxx* and *yyy***_on_error_***xxx* keywords, where *yyy* is a platform-specific prefix. Refer to the *Tivoli Software Distribution Reference Manual* for detailed information on configuration program keywords, file package definition files, and example programs.

Before Program

The before program runs (as part of a file package distribution operation) before the files in the file package are placed on the subscriber. Software Distribution supports the following before programs.

Source Before Program

Runs once on the source host before the files in the file package are sent to any subscriber. This is useful if you need to perform a special operation on the source host or if you need to prepare the source files before they are sent to any subscriber.

For example, you have a file package containing files that reside on a CD-ROM and the drive is not mounted. Use a source before program to mount the CD-ROM before Software Distribution begins to package the files for distribution.

The source before program is an advanced feature that is only available through the export and import capabilities or from the command line. Refer to the *Tivoli Software Distribution Reference Manual* for information about the **src_before_***xxx* keyword in the file package definition.

Note: Because endpoints cannot serve as a source host of a distribution, you cannot specify a source before configuration program to run on an AS/400 endpoint.

Target Before Program

Runs on every subscriber of the distribution before any files in the file package are distributed. This is useful if you need to prepare the subscribers before file package is distributed, or to perform dependency checks prior to a distribution. You can specify a target before configuration program to stop distribution to a subscriber if this program does not exit successfully.

For example, suppose you want to distribute a file package only if the subscriber has a certain amount of available disk space. Use a target before program to check the disk space availability on the subscriber. If sufficient disk space is available, the program exits successfully. Otherwise, the target before program exits with an error.

The target before program is a basic feature available through the platform-specific options dialogs on the **File Package Properties** window, through export and import capabilities, and with the command line.

After Program

An after program runs (as part of a file package distribution operation) after the files in the file package are placed on the subscriber. Software Distribution supports the following after programs.

Target After Program

Runs after the files in the file package are successfully placed on a subscriber. This is useful if you need to configure each subscriber after the files are placed there.

For example, use a target after program to restart a database server after distributing updated database files or to run the AS/400 Control Language command, **RSTLICPGM**, after distributing a program product.

The target after program is a basic feature that is available through the platform-specific options dialogs on the **File Package Properties** window from the Tivoli desktop, as well as through the export and import capabilities and from the command line.

Source After Program

Runs once on the source host after the files in the file package are distributed and after the target after programs have run on all subscribers. The source after program runs regardless of whether the distribution to all subscribers was successful or not. This is useful if you need to perform a special operation on the source host or if you need to configure the source files after they are sent to the subscribers.

Note: Because endpoints cannot serve as the source host of a distribution, you cannot specify a source before configuration program to run on an AS/400 endpoint.

For example, if your source before program mounted a CD-ROM drive, use a source after program to unmount the CD-ROM drive after the application files are distributed to the subscribers.

The source after program is an advanced feature that is only available using the export and import capabilities or from the command line. Refer to the *Tivoli Software Distribution Reference Manual* for information about the **src_after**_*xxx* keyword in the file package definition.

Removal Program

The removal program runs before the files in the file package are removed from the subscriber.

For example, suppose you have installed an application on each AS/400 system that you now want to remove. Your removal program can issue the AS/400 Control Language Command, **DLTLICPGM**, which will delete the licensed program and its associated libraries from the AS/400.

The removal program is a basic feature that is available through the platform-specific options dialogs on the **File Package Properties** window, as well as through the export and import capabilities and from the command line.

After Removal Program

The after removal program runs after the files in the file package are removed from the subscriber.

For example, suppose you want to remove an application that you previously distributed using Software Distribution. After the removal operation, run an after removal program to send a message to indicate that the application is no longer available.

The after removal program is an advanced feature that is only available through the export and import capabilities and from the command line. Refer to the *Tivoli Software Distribution Reference Manual* for information about the *yyy*_**after_removal_***xxx* keywords in the file package definition, where *yyy* is the platform-specific prefix.

Commit Program

A commit program runs when you perform a commit operation or after distribution when you perform a distribute and commit operation. You can use this configuration program to perform an operation on all subscribers at the time of or some time after distribution. This is useful if your subscribers are connected to networks of varying speed, yet you want the distributed information to be available on all subscribers at the same time.

For example, suppose your application is configured to obtain the pricing information from the files in the **/product/pricing** directory. This directory is symbolically linked to the **/price_files/normal** directory. When you distribute the new files, distribute them to a new directory, such as **/price_files/sale**. Because the files are not activated *when* the subscriber receives them, you are not concerned about when each subscriber receives them. Use the commit program to remove the symbolic link from **/product/pricing** to **/price_files/normal** and create a new symbolic link from **/product/pricing** to **/price_files/sale**.

The commit program is a basic feature that is available through the platform-specific options dialogs on the **File Package Properties** window, as well as through the export and import capabilities and from the command line.

On Error Program

An on error program runs on a subscriber if a fatal error occurs during a distribution or removal operation. This program also runs on the subscriber if a non-fatal error occurs and you set the **stop_on_error** keyword to **y**. Use this program to perform error reporting or recovery actions.

For example, you might want to use an on error program to send an event to the Tivoli Enterprise Console server, gather diagnostic information, or write to a log file. You can also use an on error program to restore any files created by before programs or by the distribution or removal operation.

The on error program is an advanced feature that is only available through the export and import capabilities and from the command line. Refer to the *Tivoli Software Distribution Reference Manual* for information on the *yyy*_**on_error_***xxx* keywords in the file package definition, where *yyy* is the platform-specific prefix.

Input Files

You can pass an input file to a configuration program, which enables you to supply data to the program as it runs. Input files also enable you to use a single program for multiple configuration programs (or for a program for multiple file packages), yet have that program perform different actions for each configuration program.

Multiple Configuration Programs and Input Files

You can specify multiple configuration programs of the same type to run on the subscriber. Additionally, you can designate input files for each configuration file. If you specify more than one configuration program of the same type, you must separate them with commas. Likewise, to specify input files for multiple programs, you must also separate them with commas. The programs must reside on the same subscriber or source and are processed in the order in which you specify them.

If you specify multiple before programs and a program fails when you distribute the file package, subsequent programs will run if you have not enabled the **Stop distribution on error** option on the **File Package Properties** window (*yyy***_before_skip_non_zero=y**).

Enabling this option stops the distribution if a program fails and does not allow subsequent programs to run.

If you specify multiple after programs and a program fails, Software Distribution does not run subsequent after programs or commit operations.

Finally, if you specify multiple removal or commit programs and a program fails, Software Distribution does not run subsequent programs of that type and does not perform the corresponding file package operation.

Return Status of Configuration Programs

The return status of a configuration program is written to the log file on a managed node specified by the **Log Information Options** section of the **File Package Properties** window if a configuration program fails.

Configuration programs for AS/400 can be run as follows:

- Before or after distributing software
- During a commit operation

- Before and after removing file packages
- If an error occurs on a target that stops distribution or removal

If a program is to be run on the target resides only on the source host, you must distribute it to the AS/400 system in a save file (*SAVF). The file containing the configuration program is automatically restored on the AS/400. If the file is already on the AS/400, ensure that it is restored or installed as a program in an AS/400 library.

OS/400 Keyword Options

Tivoli provides OS/400 keywords for configuration programs. These options are only available from the command line or by using the export and import capability to set the appropriate keywords. Similarly, you cannot specify after removal or on error programs from the desktop. You must set these configuration programs by setting keywords for the **wsetfpprgs** command. A complete list of the AS/400 keywords is listed in the following table:

OS/400 Configuration Program Options	Keyword Options
Before program	**os400_before_prog_path**
	os400_before_prog_from_src
	os400_before_input_path
	os400_before_input_from_src
	os400_before_skip_nonzero

OS/400 Configuration Program Options	Keyword Options
After program	**os400_after_prog_path**
	os400_after_prog_from_src
	os400_after_input_path
	os400_after_input_from_src
	os400_after_option
Removal programs	**os400_removal_prog_path**
	os400_removal_prog_from_src
	os400_removal_input_path
	os400_removal_input_from_src
	os400_removal_option
Commit programs	**os400_commit_prog_path**
	os400_commit_prog_from_src
	os400_commit_input_path
	os400_commit_input_from_src
	os400_commit_option

OS/400 Configuration Program Options	Keyword Options
After removal program	**os400_after_removal_prog_path**
	os400_after_removal_prog_from_src
	os400_after_removal_input_path
	os400_after_removal_input_from_src
	os400_after_removal_option
On error program	**os400_on_error_prog_path**
	os400_on_error_prog_from_src
	os400_on_error_input_path
	os400_on_error_input_from_src
	on_error_option
Platform prefix	**os400_platform_prefix**
Translate	**os400_translate_to_ebcdic**

The **os400_***xxx***_path** keywords specify the path to the configuration program on the source host or on the target AS/400 system. The configuration program can be either an AS/400 *SAVF file or a flat file that contains the source for a REXX program or an AS/400 batch script. If the configuration program is on the AS/400 target, the keyword value must specify the path in AS/400 IFS notation.

The **os400_translate_to_ebcdic** keyword on **wsetfpopts** indicates whether the file package is to be translated to EBCDIC. *SAVF files do not need to be translated to EBCDIC on distribution. Text files must be translated. Examples are shown in the next section.

OS/400 File Packages

To distribute and install software on your AS/400 systems, you must create an AS/400-specific file package. A ***file package*** is a Tivoli Software Distribution profile that describes which files and directories to distribute and how to distribute them. Configuring a file package generally involves the following steps:

1. Setting file package properties to define the source host from which the source files and directories are distributed
2. Setting platform-specific options to define destination path on the target system, to set platform-specific keywords, or to run configuration programs at selected points in the distribution

Tivoli provides the following methods to create and distribute file packages to an AS/400 endpoint:

- Using the Tivoli desktop to create an OS/400 file package if the source host is a network system that is *not* an AS/400 system. To create the file package from the desktop, complete the following steps:
  - Create a file package from the **File Package Properties** dialog
  - Set OS/400-specific distribution options from the **File Package OS/400 Options** dialog
  - Distribute the file package from the desktop or with the **wdistfp** command
- Using the Tivoli Object Packaging Utility for AS/400 to create an OS/400 file package if the source host is an AS/400 system. To use the Object Packaging Utility for AS/400, complete the following steps:
  1. Connect to the AS/400 source host using Client Access/400 on Windows 95 or Windows NT.
  2. Create and configure an OS/400 file package block (fpblock) on an AS/400 endpoint. An ***fpblock*** is a static file that contains the file package definition, file package attributes, other required file package data, and the software application or data to be distributed.

3. Distribute the file package with the **wdistfpblock** command.

The following sections provide procedures for creating and distributing OS/400 file packages. For general information about creating and defining file packages or fpblocks, refer to the *Tivoli Software Distribution User's Guide*.

Creating File Packages with the Tivoli Desktop

You can create an AS/400 file package in much the same way as you create file packages for other operating systems, such as UNIX or Windows NT. Software Distribution enables you to set properties and options from the Tivoli desktop, including the following:

- The source host from which the source files and directories are distributed
- Destination directory to which all files are distributed
- Files, directories, and other file packages to be distributed

When creating an OS/400 file package from the Tivoli desktop, you can only package directories and files that reside on the TMR server or on managed nodes in the Tivoli environment. If the source host is an AS/400 system, you must use the Tivoli Object Packaging Utility to create file packages. Refer to "Creating File Packages with the Object Packaging Utility" on page 7-18 for instructions.

Defining File Package Properties and OS/400 Options

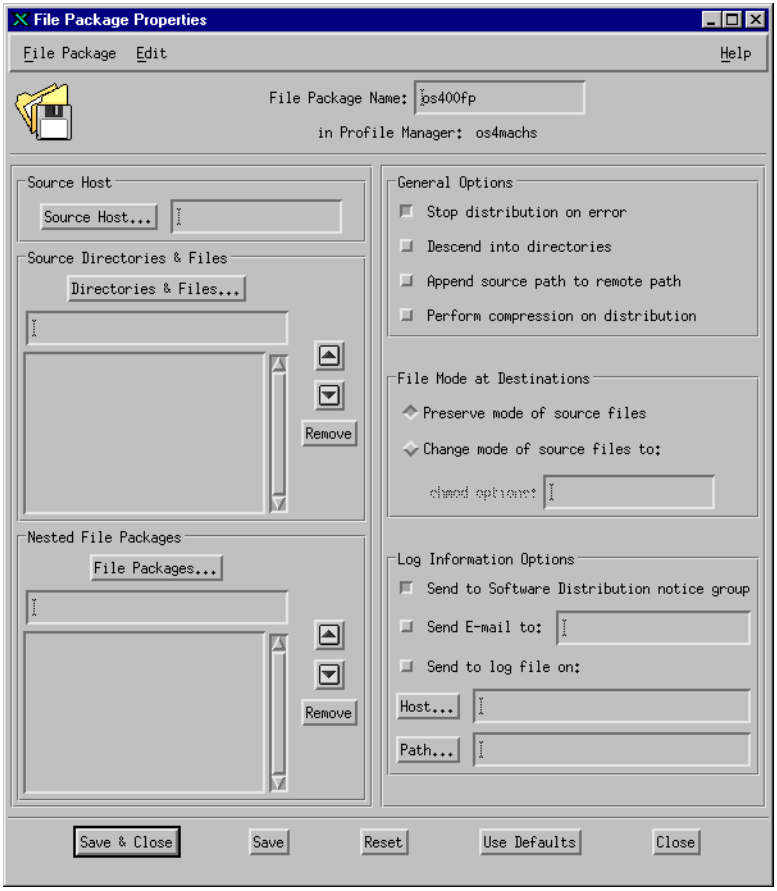
After creating a file package profile, you must set file package properties and options to specify information about the software or data that will be distributed and about the target systems. When setting file package options, you define platform-specific options, including the following:

- Destination directory path, where the distributed files will reside on platform-specific subscribers
- Configuration programs to run before or after distributing a file package, before or after removing a file package, during a commit operation, or if an error stops a distribution or a removal operation

The following procedures describe how to set file package properties and options. These steps assume that the file package profile, **os400fp**, exists in the profile manager, **os4machs**. Refer to the *Tivoli Software Distribution User's Guide* for procedures to create profile managers and file packages.

Desktop

1. From the **Profile Manager** window, double-click on a file package icon to display the **File Package Properties** window.



File Package Name: os400fp

in Profile Manager: os4machs

Source Host

Source Directories & Files

Nested File Packages

General Options

☑ Stop distribution on error

☐ Descend into directories

☐ Append source path to remote path

☐ Perform compression on distribution

File Mode at Destinations

● Preserve mode of source files

○ Change mode of source files to:

chmod options:

Log Information Options

☑ Send to Software Distribution notice group

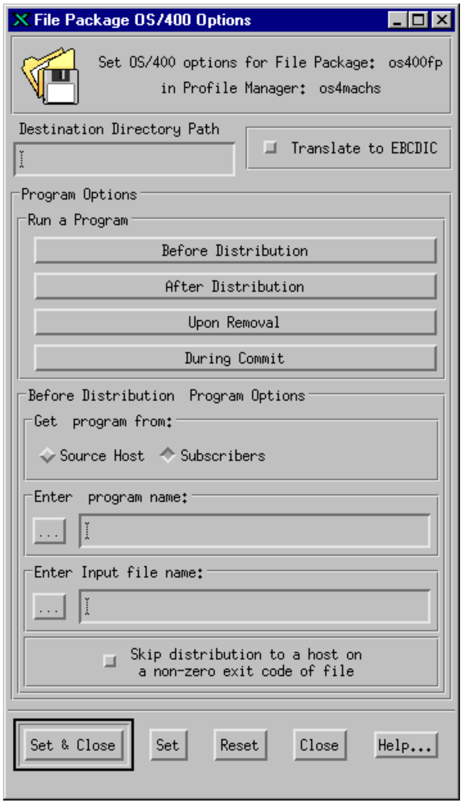
☐ Send E-mail to:

☐ Send to log file on:

The **File Package Properties** window shows the file package name, the files, directories, and nested file packages to be included, as well as the current file package option settings. The option settings displayed in this window are the default file package properties.

After defining file package properties as described in the *Tivoli Software Distribution User’s Guide*, you must define the AS/400 platform specific options.

2. Select **Platform-Specific Options** → **OS/400 Options** from the **Edit** menu to display the **File Package OS/400 Options** dialog.



File Package OS/400 Options

Set OS/400 options for File Package: os400fp
in Profile Manager: os4machs

3. In the **Destination Directory Path** field, specify the destination path for AS/400 subscribers, where the *SAVF will reside on AS/400 subscribers. Type the full path to which to distribute the file package on OS/400 subscribers. AS/400 objects can be stored in libraries that are accessed through a library called **QSYS**, or

they can be stored in the system's integrated file system. The destination path can be an integrated file system (IFS) or QSYS path name on the AS/400. However, the path name must always be specified in IFS format for example, **/qsys.lib/qgpl.lib/dist.file/**.

Note: OS/400 *SAVF can only be distributed to a QSYS destination. All types of objects can be contained in the *SAVF object. Stream files can also be sent to a physical file member in the library QSYS.

4. Select the **Translate to EBCDIC** check box to set EBCDIC translation requirements. Setting this option converts contents of the distributed files to EBCDIC on the OS/400 subscriber. You should specify this option only if you are distributing text files. Save files distributed to QSYS should not be translated.

5. Set configuration program options by using one of the buttons in the **Run a Program** group box. These buttons display options that enable you to run configuration programs as described in "Configuration Programs" on page 7-1. To run programs on a OS/400 subscriber before or after a file package operation, complete the following steps:

   a. Click the program button that corresponds to the configuration program you want to run.

   b. Specify the location of the configuration program and input files using the **Get program from** radio buttons.

      If you set the **Source host** radio button one of two things take place. Either Software Distribution copies each program from the source host to a temporary file on each subscriber or if the **install_progs** keyword is set to **y** then the programs will be restored into the destination library and left there. It then runs the programs on each subscriber before or after the file package operation is performed and removes each temporary file from each subscriber on completion. If you set the **Subscribers** radio button, Software Distribution runs each program directly from the subscriber.

   c. In the **Enter Program Name** field, enter the full path of a program to run. If you specify that the program is located on the AS/400 subscriber, the path must be a QSYS path specified in IFS format.

If you do not know the full path to the program and you have specified that the program is located on the source host, click the ellipsis (…) button to display a browser dialog that enables you to navigate through the directory structure of the machine on which it resides. You must enter the name of a program that exists at the time of the operation. The program must be located on the system (source or subscriber) specified by the **Get program from** radio buttons.

If you specify multiple programs, you must separate them with commas.

Note: Software Distribution does not check whether the programs exist until the file package operation. If a program is not found, Software Distribution logs an error.

d. If you want to specify an input file with the configuration program, enter the full path in the **Enter Input file name** field. If you specify that the configuration program is located on the subscriber, an AS/400 endpoint, the path must be a QSYS path specified in IFS format.

If you do not know the full path to the program and you specified that the program is located on the source host, click the ellipsis (…) button to display a browser dialog that enables you to traverse the directory structure of the machine where it resides. This input file must be located on the same system (source or subscriber) as the program indicated in the **Enter Program Name** field. The input file path is passed as the second argument to the program.

If you have specified multiple programs of the same type, you can also specify multiple input files separated by commas. If a program does not require an input file but subsequent programs do, you must specify the input files in the correct order and preceded by the same number of commas as the programs. Do not follow commas with spaces.

For example, if you have specified the following path in the **Enter Program Name** field:

```
/qsys.lib/progr1.pgm,
/qsys.lib/prog2.pgm,
/qsys.lib/usr.lib/prog3.pgm
```

and **prog2** does not require an input file, you must specify the following in the **Enter Input file name** field:

```
/qsys.lib/usr.lib/input1.file/input1.mbr,
/qsys.lib/usr.lib/input3.file/input3.mbr
```

Note: If you want to specify an input file that resides in a different location than the specified configuration program, use the **os400_*xxx*_input_from_src** keyword as shown in the OS/400 keywords tables where *xxx* is the program type.

e. If you are running a before program, select the **Skip distribution to a host on a non-zero exit code of program** check box to skip distributing the file package to a subscriber if the program fails and returns a non-zero exit code.

   This option allows you to implement dependency checking so that file packages are distributed to subscribers only if dependencies are met.

6. Click the **Set & Close** button to apply the changes and dismiss the **File Package Properties** window.

   —OR—

   Click the **Reset** button to set all OS/400 options of the file package to the values last saved to the file package.

Command Line

You can set or modify file package options, such as a timeout value for configuration programs specified in the file package, with the **wsetfpopts** command. To use the **wsetfpopts** command to enable EBCDIC translation in the os400fp file package distributed to an AS/400 subscriber, enter:

```
wsetfpopts -T os400 -K y @FilePackage:os400fp
```

where:

–T os400 Enables you to set os400 keywords using this command.

–K y Sets the **os400_translate_to_ebcdic** keyword, which specifies if translation is needed. A value of **y** indicates the file is to be translated to EBCDIC on the AS/400 subscriber. By default, this value is set to **n**. Text files should be translated.

@FilePackage:os400fp
Specifies to modify the **os400fp** file package.

For **wsetfpopts** command line syntax and usage as it applies for OS/400 systems, refer to the Appendix A, “Commands” on page A-1

Distributing an OS/400 File Package

After setting OS/400 file package options, use the Tivoli desktop or the **wdistfp** command to distribute the file package to AS/400 endpoints. Refer to the *Tivoli Software Distribution User’s Guide* for procedures to distribute file packages. Command line syntax and usage for the **wdistfp** command is available in the *Tivoli Software Distribution Reference Manual*.

Creating File Packages with the Object Packaging Utility

The Object Packaging Utility for AS/400 provides an easy-to-use Windows interface to AS/400 systems in the network. Because an endpoint cannot currently serve as the source host of a distribution, the Object Packaging Utility enables you to create a file package block of files and directories that are located on the AS/400 system.

With the Object Packaging Utility, you can select AS/400-specific files and directories to be packaged, set distribution options, and specify any configuration program that will run on the target during a distribution. The Object Packaging Utility creates an fpblock of the selected data, which you can distribute using Tivoli Software Distribution commands.

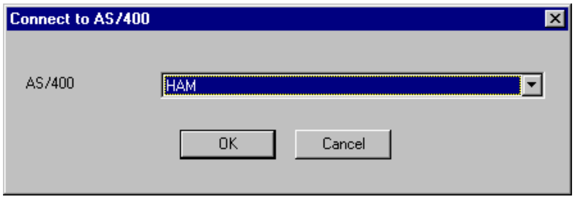
Note: To use the Tivoli Object Packaging Utility for AS/400, you must connect the Windows PC to the AS/400 source host using Client Access/400. Ensure that Client Access/400 is installed and working on the PC. For more information about installing or using Client Access/400, refer to the product manual that accompanies the Client Access/400 product.

The following procedure describes how to use the Object Packaging Utility to create an fpblock of data that resides on the AS/400 system named **ham**.

Selecting OS/400 Files and Directories

The Object Packaging Utility provides access to all files and directories that are available on the source AS/400 system. It also enables you to set distribution options to install a software application, including any necessary system-configuration changes. Use the following steps to create an fpblock with this tool:

1. Run the Object Packaging Utility by double-clicking on the **as4pack.exe** icon from the **ObjPackAS400** window or by selecting the **TME 10 Object Packaging Utility for AS/400** option from the **Start** → **Programs** → **Tivoli** menu. The **Connect to AS/400** dialog is displayed.



Connect to AS/400

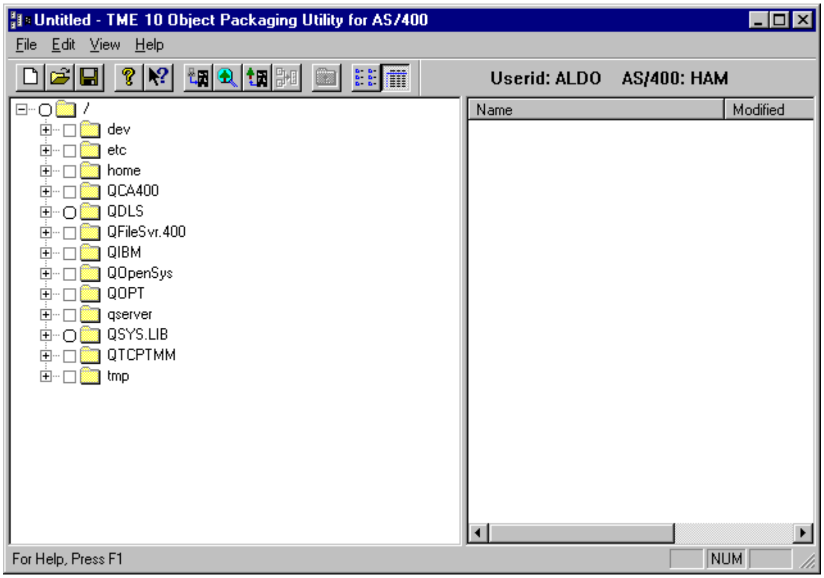
AS/400: HAM

2. In the text field, specify the name of the source AS/400 system to which the Windows PC will connect.

   —OR—

   Click the down arrow button to display a drop-down menu that lists all AS/400 system that are available. From this list, select the system to which the Windows system will connect.

3. Tivoli Object Packaging Utility for AS/400 displays the directory tree for the source AS/400 system, as follows:

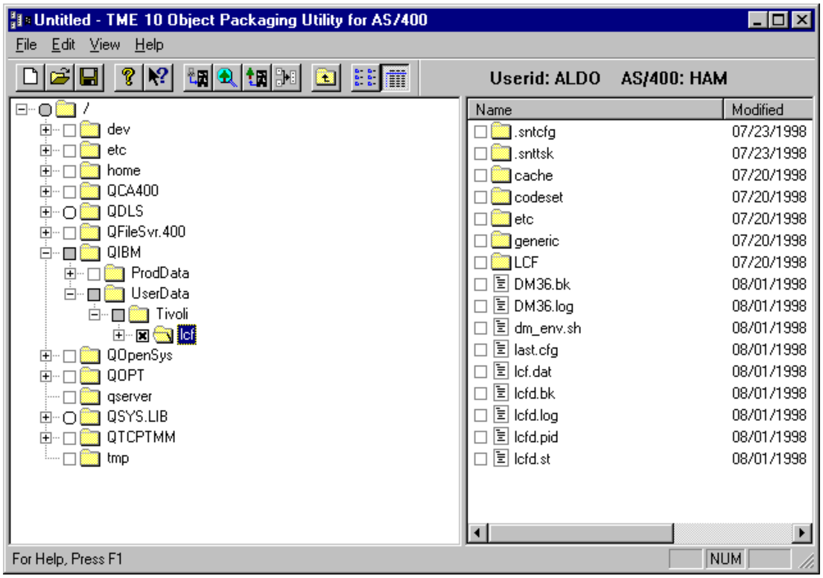


Userid: ALDO AS/400: HAM

The directory tree displays both available and unavailable directories and files. By selecting the plus sign (+) beside a directory icon, you can expand the list to view directory contents. Although you can view the entire directory structure of the AS/400 system, you can only select directories and files that appear next to a square check box.

Note: To search an IFS directory, select the **Find Tree Item** from the **View** menu and type the IFS path name in the **Enter IFS path of tree item** text field.

4. Select all files and directories that you wish to include in the AS/400 fpblock, as displayed in the following window:



Untitled - TME 10 Object Packaging Utility for AS/400

Userid: ALDO   AS/400: HAM

```
⊟ ● /
  ⊞ □ dev
  ⊞ □ etc
  ⊞ □ home
  ⊞ □ QCA400
  ⊞ ○ QDLS
  ⊞ □ QFileSvr.400
  ⊟ ▨ QIBM
      ⊞ □ ProdData
      ⊟ ▨ UserData
          ⊟ ▨ Tivoli
              ⊞ ☒ lcf
  ⊞ □ QOpenSys
  ⊞ □ QOPT
    □ qserver
  ⊞ ○ QSYS.LIB
  ⊞ □ QTCPTMM
    □ tmp
```

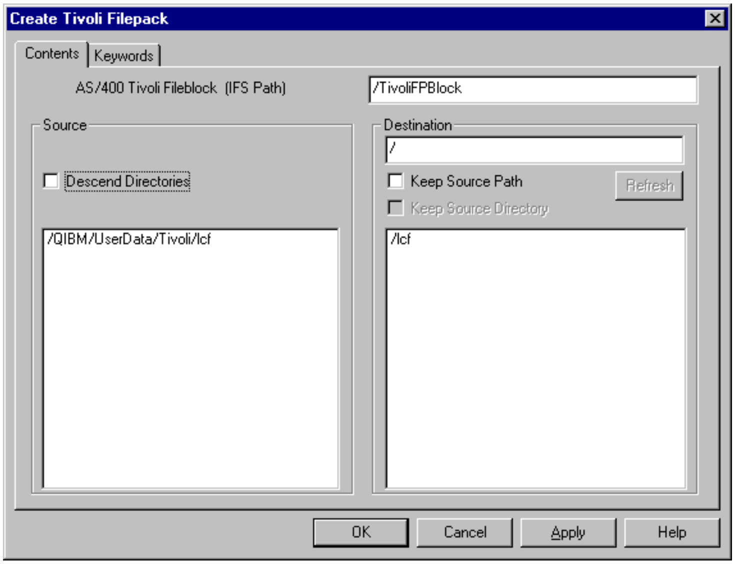
Name	Modified
□ .sntcfg	07/23/1998
□ .snttsk	07/23/1998
□ cache	07/20/1998
□ codeset	07/20/1998
□ etc	07/20/1998
□ generic	07/20/1998
□ LCF	07/20/1998
□ DM36.bk	08/01/1998
□ DM36.log	08/01/1998
□ dm_env.sh	08/01/1998
□ last.cfg	08/01/1998
□ lcf.dat	08/01/1998
□ lcfd.bk	08/01/1998
□ lcfd.log	08/01/1998
□ lcfd.pid	08/01/1998
□ lcfd.st	08/01/1998

For Help, Press F1

When you select a directory or file to package, the parent directory is grayed.

5. To create the fpblock, select the **Create Tivoli Filepack** option from the **File** menu. The Object Packaging Utility displays a confirmation screen before displaying the **Create Tivoli Filepack** window. Click the **Yes** button to continue.

6. The **Create Tivoli Filepack** window enables you to set distribution and configuration program options. This window contains the following property sheets: **Contents** and **Keywords**.



AS/400 Tivoli Fileblock (IFS Path): /TivoliFPBlock

Source
☐ Descend Directories
/QIBM/UserData/Tivoli/lcf

Destination
/
☐ Keep Source Path
☐ Keep Source Directory
/lcf

The **Contents** property sheet provides the following options to set distribution options:

a. The **AS/400 Tivoli Fileblock (IFS Path)** field specifies the complete IFS path in which the Tivoli fpblock is stored after it is created.

   Note: The fpblock cannot be stored in the QSYS file system.

b. Set the **Descend Directories** check box in the **Source** group box to enable the Object Packaging Utility to descend into source directories when creating the fpblock.

   All subdirectories and files are part of the fpblock. When this check box is active, an additional **/…** is added after the source and destination directories to indicate that the entire subtree from that point is part of the fpblock.

If any of the following conditions are true, the **Descend Directories** check box will not be active:

- The destination is a QSYS library
- All items being distributed are QSYS objects

c. Select the **Destination** text field to specify the destination directory to which the fpblock contents are distributed. Changing the contents of this field activates the **Refresh** button. Click the **Refresh** button to update the destination list to reflect the changes made in the **Destination** field.

d. Set the **Keep Source Path** check box to specify that the entire source path of the directory will be appended to the destination field contents. This check box can only be used in conjunction with the **Destination** field.

If any of the following conditions are true, the **Keep Source Path** check box will not be active:

- The destination is a QSYS library
- All items being distributed are QSYS objects

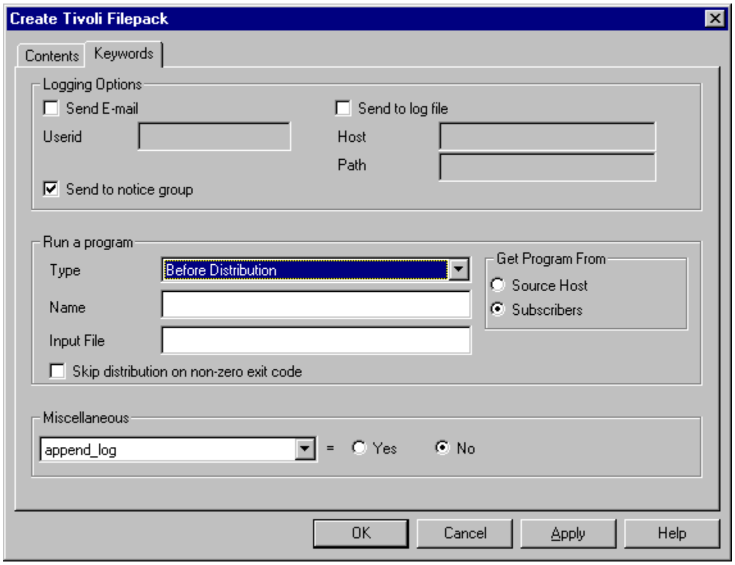
e. Set the **Keep Source Directory** check box to specify that the final directory designated in the source path will appear as part of the destination.

If any of the following conditions are true, the **Keep Source Directory** check box will not be active:

- The destination is a QSYS library
- All items being distributed are QSYS objects
- The **Descend Directories** check box is not active
- The **Keep Source Path** check box is active

f. Click the **Apply** button to save changes made to the **Contents** property sheet.

7. Set the following logging and configuration program options from the **Keywords** property sheet:



Create Tivoli Filepack

Logging Options

□ Send E-mail

Userid

□ Send to log file

Host

Path

☑ Send to notice group

Run a program

Type: Before Distribution

Name

Input File

□ Skip distribution on non-zero exit code

Get Program From

○ Source Host

● Subscribers

Miscellaneous

append_log = ○ Yes ● No

a. Set the **Send E-Mail to** check box for Software Distribution to send e-mail to the specified address when the fpblock is distributed. If you set **Send E-Mail**, provide a user ID for each user to receive status.

b. Set the **Send to log file** check box for Software Distribution to post a notice to the distribution log file. If you set **Send to log file**, you must specify the following:

   ■ In the **Host** field, type the name of the system where the log file resides.

   ■ In the **Path** field, type the complete path to the log file.

c. Set the **Send to Distribution notice group** check box for Software Distribution to send a notice to its notice group when the fpblock is distributed.

d. To run a configuration program on the target during a distribution, provide each type of configuration program in the **Type** field.

—OR—

Click the down arrow button to display a drop-down menu of exit programs that can be associated with the fpblock. You can specify separate programs for any combination of the following:

- After Distribution
- After Removal
- Before Distribution
- During Commit
- On Error
- Upon Removal

e. Type the full path and name of the configuration program in the **Name** field.

f. If you are specifying optional input files for a program, type the input file name to be given to the program in the **Input File** field.

g. In the **Get Program From** group box, set the **Source Host** or **Subscribers** radio button to specify the location of the program to be run.

If **Source Host** is active, Software Distribution expects the configuration program to be located on the Tivoli source host from where the fpblock is distributed. If **Subscribers** is active, Software Distribution expects the configuration program to be located on the AS/400 systems that will receive the fpblock.

Note: You can specify several configuration programs by repeating steps d through g for each program type.

h. The **Skip distribution on non-zero exit code** check box is only active for before distribution configuration programs. Set this check box to terminate the distribution if the before distribution program fails.

i. Select any Software Distribution keywords from the **Miscellaneous** drop-down menu. This menu contains several keywords that are supported by Software Distribution.

If the keyword has a boolean value for yes or no, a radio box appears when the keyword is active. If the keyword has a string value associated with it, a text field appears. For information about Software Distribution keyword use, refer to the *Tivoli Software Distribution Reference Manual*.

Note: You can specify multiple keywords by selecting and supplying the values for each keyword.

j. Click the **Apply** button to save changes made to the **Keywords** property sheet.

8. Click the **OK** button to accept all settings on the **Contents** and **Keywords** property sheets and begin the packaging operation. The Object Packaging Utility creates the fpblock and stores it in the IFS destination directory that you specified on the **Contents** property sheet.

Distributing an OS/400 File Package Block

After creating an OS/400 fpblock with the Tivoli Object Packaging Utility for AS/400, you must move this file to the Software Distribution source host to distribute the fpblock to AS/400 endpoints. A common method of transferring files from an AS/400 system to the source host is to use the File Transfer Protocol (FTP). Files created with the Object Packaging Utility are marked to be automatically restored on distribution to an AS/400.

To distribute the file package block to AS/400 endpoints, you must use the **wdistfpblock** command. Refer to the *Tivoli Software Distribution User’s Guide* for procedures and the *Tivoli Software Distribution Reference Manual* for reference information (including syntax) for the **wdistfpblock** command.

Using UserLink for Tivoli Software Distribution

UserLink for Tivoli Software Distribution enables you to retrieve Software Distribution profiles using the UserLink interface. This interface enables you to complete the following operations:

- Selectively download new software applications and data, or update existing software

- Download software packages that were not received at a prior distribution time

For information about UserLink for Tivoli Software Distribution, refer to the *Tivoli Software Distribution User’s Guide*.

The procedures to configure and use UserLink for Software Distribution are the same as with other operating systems. However, because AS/400 systems do not support web browsers, you must use a browser on a proxy system to connect to the Tivoli server to view file packages to which the endpoint is subscribed. Enter the following URL:

```
http://systemname:port/C/userlink.htm
```

where:

systemname	The name of the AS/400 endpoint.
port	The port number of the AS/400 endpoint.

A

Commands

Tivoli for AS/400 command line interface (CLI) commands enable you to perform system operations, such as installation or file package distribution from the command line instead of using the Tivoli desktop.

Using Tivoli Commands

It is often necessary or convenient to invoke a Tivoli management application operation from the command line rather than from the desktop. For example:

- You do not have access to a desktop, perhaps because you dialed up over a modem
- You want to group several operations in a shell script or batch file
- An operation is not available using the desktop
- You prefer to invoke a command from a shell

Command Line Syntax

The manual pages in this appendix use the following special characters to define the command syntax:

[ ] Identify optional arguments. Arguments not enclosed in brackets are required.

… Indicates that you can specify multiple values for the previous argument.

| Indicates mutually exclusive information. You can use the argument to the left of the separator or the argument to its right. You cannot use both arguments in a single use of the command.

{ } Delimits a set of mutually exclusive arguments when one of the arguments is required. If the arguments are optional, they are enclosed in [ ] (brackets).

For example:

wdistfp {**–a** | **–n** | **–s**} {**–b** | **–c** | **–d** | **–p**} *fp_name* [*subscriber…*]

The … (ellipses) following the *subscriber* argument indicate that you can specify multiple subscribers. Also, you must specify one argument in each set of arguments delimited by the | (logical or) and enclosed in { } (braces), and the *fp_name* argument.

Another example is the **wsetfpprgs** command.

```
wsetfpprgs {–t type | –T type} [keyword_option …] fp_name
```

In this example, the **–t** *type* and **–T** *type* arguments are mutually exclusive; you must use one or the other. The *keyword_option argument* is an optional argument and you can specify it more than once. The *fp_name* argument is required.

Object References

When you reference an object in a command, the reference is not an absolute object reference like those used in programming. Instead, the reference is the label you gave the object when it was created. For example, when you refer to the **Engineering** object in a command, this reference corresponds to the **Engineering** policy region.

There are two different forms of names that can be used with commands:

- Registered names
- Object paths

Tivoli commands support both naming schemes. Sometimes, you will find it more convenient to use one form over the other.

Registered Names

The key concept behind the name registry is a ***registered name***. A registered name is a resource instance that is registered with the name registry when it is created. Every resource has a name and is of some particular type. For example, an AS/400 machine called as4mach has a name **as4mach** and is of type **Endpoint**. An example of a registered name used as an argument for the **wdistfp** command is:

```
wdistfp {keyword_options} fp_name [subscriber]
```

```
w4inslcf.pl {keyword_options} os400fp @Endpoint:as4mach
```

The syntax for specifying a resource using a registered name is @*type*:*name*, where *type* is the resource type and *name* is the instance on which you wish to perform some operation. You must always specify the "@" before a registered name. In the above example, the file package **os400fp** is distributed to the endpoint **as4mach**.

Object Paths

Object paths provide another way for you to specify an object name. They are similar to paths in file systems and can be relative or absolute. An absolute path begins with a "/" character. A relative path begins with any character including the special path components "." (current directory) and ".." (parent directory). Some examples of object paths used as arguments for the **wcd** and **wdistfp** commands are:

```
wdistfp -a -d /Regions/Distr/Source/os400fp \
/Regions/as4mach-region/as4mach
```

```
wcd /Library/Endpoint
```

```
wdistfp -a -d /Library/FilePackage/os400fp./as4mach
```

The syntax for specifying a resource using an object path is **/Regions/***ObjectPath*/[*type*:]*name*, where *ObjectPath* is the path to the object, *type* is the resource type, and *name* is the particular instance on which you wish to perform some operation. Use the optional *type* specifier if the specified resource has the same name as another resource of a different type.

In the above examples, the **PS_docs** file package is distributed to the managed node **as4mach**. This file package resides in the **Distr** policy region and **Source** profile manager, as indicated by the object path after **/Regions**. You can also specify the file package and managed node in terms of their class library (**/Library**), as illustrated by the second and third examples.

Tivoli for AS/400 Commands

The following table lists the Tivoli Software Distribution commands that you can run from the command line of a managed node:

Command	Purpose
w4inslcf.pl	Installs an endpoint on an AS/400 system.
wcrtsec	Creates a new security record and adds it to a security profile.
wmodsec	Modifies the attributes and values of a security record.
wsetfpopts	Sets or modifies the options for a file package.
wsetfpprgs	Sets or modifies the before, after, remove, after removal, commit, or on error program information associated with a file package.

Command Syntax

This following section lists Tivoli commands that are specifically pertinent to managing operations on or from AS/400 endpoints. Syntax and descriptions of command functions are provided for each command.

w4inslcf.pl

Installs an endpoint on as AS/400 system.

Synopsis

w4inslcf.pl [-**v**} [**–a**] [**–F**] [**–g** *gwlabel*[+*port*]] [**–l** *ep_port*]
[**–L** *config_args*] [**–N** *code*] [**–P**] [**–s** *dir_name*] [**–Y**]
{*endpoint* | **–f** *filename*}

Description

The **w4inslcf** command installs and optionally starts an AS/400 endpoint daemon job on one or more AS/400 systems. You can install multiple endpoints by listing the system name on the command line or supplying a file containing the system names. The file must contain one system name per line.

The command checks for prerequisites, sends code to the endpoint to be installed, using FTP, and then restores the product, **1TMELCF**, to the AS/400, using **RSTLICPGM**. If requested, the endpoint is also started by issuing the AS/400 CL command **STRTMEEPT**.

Authorization

You must have root access to install endpoints but you do not need any Tivoli authorization roles. On the AS/400 you will need authority to use the **RSTOBJ** and **RSTLICPGM** commands and ***SAVSYS** special authority.

Arguments

–a Specifies asynchronous installation of endpoints. By default, the command waits for the endpoint to log in to its gateway before installing the next endpoint.

–F Forces an overwrite of an existing installation.

–f *filename* Specifies the file containing a list of systems on which to install endpoints. This file contains one system name per line, specifying the user ID and password to be used. Each line in this file must be in the following format:

```
host [userID [password]]
```

–g *gwlabel*[+*port*]
Specifies the IP address or host name and optionally the port number of the gateway to which the endpoint will log in.

–l *ep_port* Specifies the port number for the endpoint. The default port number is 9495.

–L *config_args* Passes configuration arguments to the daemon for starting the endpoint. To pass multiple arguments, enclose them in quotes. Refer to the **lcfd** command in the *Tivoli Management Framework Reference Manual* for a list of valid arguments.

–N *code* Specifies additional languages to support by AS/400 code. The following are the codes: 2980 (Brazilian, Portuguese), 2989 (Chinese, simplified), 2924 (English uppercase and lowercase), 2938 (English uppercase), 2928 (French), 2929 (German), 2932 (Italian), 2962 (Japanese), 2986 (Korean), and 2931 (Spanish). To specify multiple languages, enclose them in quotes. The defaults are 2924 and 2938.

–P Specifies to prompt the user for a password. This argument overrides existing entries in a **$HOME/.netrc** file used for automatic logins.

–s *dir_name* Specifies the directory that contains the endpoint installation image. This directory can be on a compact disc, the TMR server, a gateway, or a network drive.

–Y Specifies that the installation should proceed without confirmation. The default is to request confirmation.

–v Writes verbose messages to standard output. Error messages are still written to standard error.

endpoint … Specifies the names or IP addresses of AS/400 systems on which the endpoints will be installed.

Return Codes

This command returns the following codes to standard output:

0 Indicates successful completion.

-1 Indicates failure due to an error.

Note: Host names for failed installations are written to **os4LcfH.err** and can be retried with the command using the **–f os4LcfH.err** keyword specified on the command.

Examples

1. The following example installs the AS/400 endpoint on the AS/400 systems **as41.tivoli.com** and **as42.tivoli.com**, connects to gateway **smithers.tivoli.com** using port 9494. The installation image is located in the **/cdrom/1tmelcf** directory.

```
w4inslcf.pl -v -g smithers.tivoli.com+9494 as41.tivoli.com
as42.tivoli.com -s /cdrom/1tmelcf
```

2. The following example installs the AS/400 endpoint on the AS/400 system as**42.tivoli.com**, connects to gateway **smithers.tivoli.com** using port 9494, and installing French and German language support.

```
w4inslcf.pl -v -g smithers.tivoli.com+9494 -N '2928 2929'
as42,tivoli.com
```

wcrtsec

Creates new records and adds them to a security profile.

Synopsis

wcrtsec *record_type* [**–l**] [**–s** *attribute*=*value*]… *profile*
{*record_name* | *endpoint_type*:*resource_type*:*resource_name*}

Description

The **wcrtsec** command creates a new security profile record. You cannot create multiple records from a single iteration of the **wcrtsec** command. Information will not be updated on system files and databases until the security profile is distributed.

Notes

A group can be any set of users within an organization. Example groups include divisions, departments, project teams, or cross-functional groups, such as all tellers in a bank.

A role defines the set of capabilities required to carry out a job function within an organization. In general, there should be one role defined per unique job function.

Resources are programs, files, systems, or similar objects in a distributed enterprise that need access protection.

System policy sets password, audit, login, and other global policies on an enterprise-wide level, instead of on a per-user basis.

Many of the attributes allow you to define access permissions for resources. The valid permissions differ for each resource type. The *Tivoli Security Management User’s Guide* lists and defines the permissions associated with each type of resource. However, the following list provides brief descriptions for access permissions that are used by more than one resource type:

Read (R) Allows users to read the resource.

Write (W) Allows users to write (or modify) the resource.

Execute (X) Allows users to start the resource as a process or change directories to the resource.

Delete (D) Allows users to delete the resource.

Update (U) In UNIX, allows users to read, write, and execute the resource. In NT, allows users to read, write, execute, and delete the resource.

Full Control (F)
Allows users to have full control of a resource and its attributes, including the ability to change permissions and ownership.

No Access (N) Indicates access to the resource is not permitted.

Access (A) Allows users to have access to the resource.

Ownership (O)
Allows users to change ownership of the resource.

Change Permissions (P)
Allows users to change permissions of the resource.

Authorization

security_admin and **admin**

Arguments

–l Locks the record. The default is **unlocked**. This argument does not have any effect on records that are part of a security profile that is in a dataless profile manager.

–s *attribute*=*value*
Assigns a value to the specified attribute. See the "Attributes" section for a list of possible attributes or use the **wlssec** command to display a list of record-type specific attributes.

Use commas to separate multiple values within a single **–s** argument. Use spaces to separate multiple **–s** arguments. Enclose the value in quotation marks if it contains spaces, commas, or special characters.

*endpoint_type***:***resource_type***:***resource_name*
For resource records, specifies the record's endpoint type, resource type, and resource name. The following

list provides information about valid values for each item:

endpoint_type Examples of valid endpoint types are **NT** and **UX** and **OS4**.

resource_type Valid values differ by platform. For OS/400, valid values are FILE, PROGRAM, COMMAND, DIRECTORY, and STREAMFILE.

resource_name The resource name must be the name of a managed resource. Tivoli Security Management does not create operating system resources. For OS/400 names, IFS notation should be used.

For resources of type FILE and DIRECTORY, you can use standard wildcard characters in the resource name to protect a collection of resources. These resources must have a similar naming structure and security access needs. An asterisk (*) matches more than one character in the same position of the resource name. The asterisk must be at the end of the resource name for OS/400. If you use wildcard characters, enclose the resource name in quotation marks.

profile Specifies the name of the security profile in which to create the record.

record_name Specifies the name of the group, role, or system policy record. Group, role, and system policy names must be unique within a profile manager.

If the record is to be distributed to an OS/400 endpoint, then the specified record name cannot be the same as an existing user profile name, or the name of any other group or role record.

record_type Specifies the type of security record to create. Valid values are **Group**, **Role**, **Resource**, and **SystemPolicy**.

For resource records, the combination of *record_name* or *endpoint_type*:*resource_type*:*resource_name* and *record_type* must be unique within a profile manager.

Attributes

This section lists the attributes that are available for various record types.

General Attributes

The following attributes are available for all record types.

Description Specifies a textual description of the record. Enclose the description in quotation marks if the text includes spaces, commas, or special characters.

Name Specifies the name of the record. This attribute cannot be assigned from the command line. It is inherited from the *record_name* argument or from the *endpoint_type*:*resource_type*:*resource_name* argument.

RecordType Specifies whether the record is a **Group**, **Role**, **Resource**, or **SystemPolicy** record. This attribute cannot be changed after it has been set using the *record_type* argument.

Group Attributes

OS4LoginAudit

Specifies whether login attempts are audited. Valid values are **All**, **None**, **Success**, and **Failure**.

OS4LoginTimes

Specifies the days and times that members of the group can log in. The value of this attribute must take the format:

```
'days(Days) time(Time)'
```

The syntax of *Days* is:

```
{Anyday | Weekdays | [Mon], [Tue], [Wed],
[Thu], [Fri], [Sat], [Sun]}
```

The syntax of *Time* is:

```
{Anytime | start_time:end_time}
```

where *start_time* and *end_time* are expressed in the format *hhmm* (two hour digits and two minute digits). The default value of this attribute is **days(Anyday) time(Anytime)**. A time value of **0000:2359** is the equivalent to **Anytime**.

Enclose the specified values in quotation marks.

OS4Name Specifies a platform-specific name for the group. For OS/400 names, there is a limit of 10 characters.

OS4Audit Specifies whether resource access attempts by members of the group are audited. Valid values are **All**, **None**, **Success**, and **Failure**.

Roles Specifies the security roles assigned to the group. This attribute is for referential integrity purposes only. It cannot be set.

TMEUserMembers

Specifies a list of Tivoli User Administration user names that are members of the security group being defined. You must specify **TMEUserMembers** in the following format:

```
'UserProfileName:TMEUserName, …'
```

Use commas to separate multiple user members.

OS4UserMembers

Specifies a list of system-specific user names that are members of the group being defined. These names should be listed in the same manner as they are defined on the endpoint.

Use commas to separate multiple user members.

Resource Attributes

AccessTimes Specifies the days and times that users can access the resource. The value of this attribute must take the format:

```
'days(Days) time(Time)'
```

The syntax of *Days* is:

```
{Anyday | Weekdays | [Mon], [Tue], [Wed], [Thu],
[Fri], [Sat], [Sun]}
```

The syntax of *Time* is:

```
{Anytime | start_time:end_time}
```

where *start_time* and *end_time* are expressed in the format *hhmm* (two hour digits and two minute digits). The default value of this attribute is **days(Anyday) time(Anytime)**. A time value of **0000:2359** is the equivalent to **Anytime**.

Enclose the specified values in quotation marks.

DefAccess Specifies the default access permissions for the resource record. Values must be specified in the following format:

```
perms(permissions,…)
```

See *Tivoli Security Management User’s Guide* for a list of permissions for each resource type.

Use commas to separate multiple access permissions.

EpType Specifies the endpoint type of the resource.Valid endpoint type is **OS4**. An endpoint’s type cannot be changed after it has been set using the *endpoint_type* argument.

ResAudit Specifies whether resource access attempts are audited. Valid values are **All**, **None**, **Success**, and **Failure**.

ResType Specifies the resource type of the record. This attribute cannot be assigned from the command line. It is inherited from the *resource_type* argument.

TCPAccess This parameter is not supported for OS/400.

Role Attributes

Children Specifies a list of role records that inherit resource access information from a parent role. This attribute is for referential integrity purposes only. It cannot be set. A role’s **Children** attribute can only be changed by changing the **Parent** attribute in another role record.

OS4Groups Specifies a list of platform-specific groups directly assigned to the role.

OS4ResAccess Specifies a list of platform-specific resources and associated permissions that have not previously been defined as resource records.ou must specify values in the following format:

```
'resource_type:resource_name perms(permissions,…)
[via(via_type:via_name,…)], …'
```

where:

resource_type:resource_name
Specifies the type of resource and the name of the resource.

permissions Assigns access permissions.

OS4TMEResAccess Specifies a list of platform-specific resources and associated permissions that have previously been defined as Tivoli resource records. You must specify values in the following format:

```
'[profile:]resource_type:resource_name
perms(permissions,…)
[via(via_type:via_name,…)], …'
```

where:

profile Specifies the name of a security profile.

resource_type:resource_name
Specifies the type of resource and the name of the resource.

permissions Assigns access permissions.

Parent Specifies the parent record. The child role inherits resource access permissions from the parent role. Inherited resource access permissions can be overridden in the child record by setting a new value in the child record. You must specify a value in the format [*SecurityProfileName***:**]*ParentRoleName*.

TMEGroups Specifies a list of Tivoli Security Management groups directly assigned to the role.

For NT groups, you must specify a domain-qualified value, in the format *domain_name\group_name*, if the group name is not unique among the global groups of all trusted domains. (A group record corresponds to an NT global group, and a role record corresponds to an NT local group.)

System Policy Attributes

OS4Lockout Specifies whether user accounts are locked based on the number and frequency of failed login attempts, and for what duration the lockout should remain in effect. Valid values are **Never** or *number* where *number* specifies how many failed login attempts must occur before the account is locked.

OS4MaxConLogins
Specifies the maximum number of concurrent logins that a user can have open at the same time from each machine. Valid values are **None**, or **Unlimited.** If any number is set for **MaxConLogins,** it is mapped to **Unlimited**.

MaxGraceLogins
Specifies the maximum number of times that a user can log in after the user's password has expired. Valid values are **None**, **Unlimited**, or any integer greater than zero.

MaxInactiveDays, OS4MaxInactiveDays

Locks the user’s account if the user does not log in within the specified number of days. Valid values are **Never** or any integer greater than zero.

PwChk, OS4PwChk

Specifies whether password checking is enabled on subscribing endpoints. Valid values are **True** and **False**.

PwExpireWarning

Specifies whether a warning should be issued before a password expires. Valid values are **True** and **False**.

PwHistory, OS4PwHistory

Specifies the number of previous passwords to store to prevent the reuse of old passwords. Valid values for OS/400 are 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, and 32.

PwMaxAge, OS4 Max Age

Specifies the maximum number of days a user can go without changing the password. Valid values are Unlimited or any integer greater than zero.

PwMaxReps, OS4PwMaxReps

Specifies the maximum number of repetitive characters the password can contain. The value of this attribute can be:

0 for no repetitive characters

1 for no adjacent repeated characters

2 for no restrictions on repeated characters

MinPwAge Specifies the minimum number of days that must elapse before the user can change the password. Valid values are **Now**, indicating the user can change the password at any time, or any integer greater than zero.

PwMinAlphanums

Specifies the minimum number of alphanumeric characters (a-z, A-Z, 0-9) that must be present in the password. The value of this attribute must be an integer equal to or greater than zero.

PwMinAlphas Specifies the minimum number of alphabetical characters (a-z, A-Z) that must be present in the password. The value of this attribute must be an integer equal to or greater than zero.

PwMinLen, OS4PwMinLen
Specifies the minimum number of characters that must be present in the password. The value of this attribute must be an integer equal to or greater than zero. A value of zero specifies that passwords may be of any length and that password length will not be checked.

PwMinLowers Specifies the minimum number of lowercase alphabetical characters (a-z) that must be present in the password. The value of this attribute must be an integer equal to or greater than zero and less than 10.

PwMinNums Specifies the minimum number of numbers (0-9) that must be present in the password. The value of this attribute must be an integer equal to or greater than zero.

PwMinSpecials, OS4MinSpecials
Specifies the minimum number of special characters that must be present in the password. A special character is any non-alphanumeric character. The value of this attribute can be:

0 for no special characters

1 for one or more

PwMinUppers Specifies the minimum number of uppercase alphabetical characters (A-Z) that must be present in the password. The value of this attribute must be an integer equal to or greater than zero.

OS4ResTypeAudit
Indicates whether attempts to access the resources of the specified type (or all resource types) should be

logged. You must specify values in the following format:

```
'[resource_type | all]
([None] | [All] | [Success] | [Failure]), …'
```

OS4ResTypeAccess

Specifies—globally—the access rights to all UNIX resources of a specific resource type that have not previously been defined as resource records.

You must specify values in the following format:

```
'[resource_type {[perms(permissions,…)]
[state(Enabled | Disabled)]}, …'
```

where:

resource_type Specifies the type of resource to protect.

permissions Assigns access permissions. See the *Tivoli Security Management User’s Guide* for a list of permissions for each resource type.

state Indicates whether the **ResTypeAccess** feature is enabled or disabled.

wmodsec

Modifies the attributes and values of a security record.

Synopsis

wmodsec *record_type* [**–l** | **–u**] [**–a** *attribute=value…*]
[**–r** *attribute=value…*] [**–s** *attribute=value…*] [**–y**] *profile*
[*record_name* | *endpoint_type:resource_type:resource_name*]

Description

The **wmodsec** command modifies the values assigned to attributes within a security profile record. When you modify a security record, the **wmodsec** command stores the changed information in a security profile database and posts a notice to the **Security** notification group. Information will not be updated on system files and databases until the security profile is distributed.

Use the **wlssec** command to display previously defined attributes and values for security profile records.

Notes

Two administrators cannot simultaneously modify a security record. The security profile locks until the modifications to the record are complete. When you modify a record from the command line, the lock is held only while the command executes. However, if the record is modified from the Tivoli desktop, the profile will be locked until the appropriate dialog is closed.

If you do not specify which records to modify, all records of the specified *record_type* will be modified as indicated by the **–a**, **–r**, and **–s** arguments.

Authorization

security_admin and **admin**

Arguments

–a *attribute*=*value*

Adds or replaces a value for the specified attribute. If the attribute accepts only one value, the specified value replaces the previous value. If the attribute accepts multiple values, the specified value is added to the list of current values.

Use commas to separate multiple values within a single **–a** argument. Use spaces to separate multiple **–a** arguments. Enclose the value in quotation marks if it contains spaces, commas, or special characters.

–l Locks the record.

–r *attribute*=*value*

Removes the specified value previously assigned to the attribute. The **–r** argument is only valid for attributes that accept multiple values, such as those that define group and role membership or assign permissions.

Use commas to separate multiple values within a single **–r** argument. Use spaces to separate multiple **–r** arguments. Enclose the value in quotation marks if it contains spaces, commas, or special characters.

–s *attribute*=*value*

Replaces the previously-assigned value with the specified value. This argument does not add or delete single values from a list of values.

Use commas to separate multiple values within a single **–s** argument. Use spaces to separate multiple **–s** arguments. Enclose the value in quotation marks if it contains spaces, commas, or special characters.

–u Unlocks the record.

*endpoint_type***:***resource_type***:***resource_name*

For resource records, specifies the record’s endpoint type, resource type, and resource name. The following list provides information about valid values for each item:

endpoint_type Valid endpoint type is **OS4**.

resource_type Valid values differ by platform. For OS/400, valid values are FILE, PROGRAM, COMMAND, DIRECTORY, and STREAMFILE.

resource_name The resource name must be the name of a managed resource. Tivoli Security Management does not create operating system resources. the resource name in quotation marks. For OS/400 names, IFS notation should be used.

For resources of type FILE and DIRECTORY, you can use standard wildcard characters in the resource name to protect a collection of resources. These resources must have a similar naming structure and security access needs. An asterisk (*) matches more than one character in the same position of the resource name. The asterisk must be at the end of the resource name for OS/400. If you use wildcard characters, enclose the resource name in quotation marks.

profile Specifies the name of the security profile in which to modify the record.

record_name Specifies the name of the group, role, or system policy record. Group, role, and system policy names must be unique within a profile manager.If the record is to be distributed to an OS/400 endpoint, then the specified record name cannot be the same as an existing user profile name, or the name of any other group or role record.

record_type Specifies the type of security record to modify. Valid values are **Group**, **Role**, **Resource**, and **SystemPolicy**.

For resource records, the combination of *record_name* or *endpoint_type***:***resource_type***:***resource_name* and *record_type* must be unique within a profile manager.

Attributes

This section lists the attributes that are available for various record types.

General Attributes

The following attributes are available for all record types.

Description Specifies a textual description of the record. Enclose the description in quotation marks if the text includes spaces, commas, or special characters.

Name Specifies the name of the record.

This attribute cannot be assigned from the command line. It is inherited from the *record_name* or *endpoint_type:resource_type:resource_name* argument.

RecordType Specifies whether the record is a **Group**, **Role**, **Resource**, or **SystemPolicy** record. This attribute cannot be changed after it has been set using the *record_type* argument of the **wcrtsec** command.

Group Attributes

LoginAudit, OS4LoginAudit

Specifies whether login attempts are audited. Valid values are **All**, **None**, **Success**, and **Failure**.

LoginTimes, OS4LoginTimes

Specifies the days and times that members of the group can log in. The value of this attribute must take the format:

```
'days(Days) time(Time)'
```

The syntax of *Days* is:

```
{Anyday | Weekdays | [Mon], [Tue], [Wed],
[Thu], [Fri], [Sat], [Sun]}
```

The syntax of *Time* is:

```
{Anytime | start_time:end_time}
```

where *start_time* and *end_time* are expressed in the format *hhmm* (two hour digits and two minute digits). The default value of this attribute is **days(Anyday) time(Anytime)**. A time value of **0000:2359** is the equivalent to **Anytime**.

Enclose the specified values in quotation marks.

OS4Name Specifies a platform-specific name for the group. By default, the value for this attribute is the same as the **Name** attribute.For OS/400, it can have a maximum length of 10 characters.

ResAudit, OS4ResAudit
Specifies whether resource access attempts by members of the group are audited. Valid values are **All**, **None**, **Success**, and **Failure**.

Roles Specifies the security roles assigned to the group. This attribute is for referential integrity purposes only. It cannot be set.

TMEUserMembers
Specifies a list of Tivoli User Administration user names that are members of the security group being modified. You must specify **TMEUserMembers** in the following format:

```
'UserProfileName:TMEUserName, …'
```

Use commas to separate multiple user members.

OS4UserMembers
Specifies a list of system-specific user names that are members of the security group being modified. These names should be listed in the same manner as they are defined on the endpoint.

Use commas to separate multiple user members.

Resource Attributes

AccessTimes Specifies the days and times that users can access the resource. The value of this attribute must take the format:

```
'days(Days) time(Time)'
```

The syntax of *Days* is:

```
{Anyday | Weekdays | [Mon], [Tue], [Wed], [Thu],
[Fri], [Sat], [Sun]}
```

The syntax of *Time* is:

```
{Anytime | start_time:end_time}
```

where *start_time* and *end_time* are expressed in the format *hhmm* (two hour digits and two minute digits). The default value of this attribute is **days(Anyday) time(Anytime)**. A time value of **0000:2359** is the equivalent to **Anytime**.

Enclose the specified values in quotation marks.

DefAccess Specifies the default access permissions for the resource record. Values must be specified in the following format:

```
perms(permissions,…)
```

Use commas to separate multiple access permissions.

EpType Specifies the endpoint type of the resource. Valid endpoint type value is **OS4**. An endpoint’s type cannot be changed after it has been set using the *endpoint_type* argument.

ResAudit Specifies whether resource access attempts are audited. Valid values are **All**, **None**, **Success**, and **Failure**.

ResType Specifies the resource type of the record. This attribute cannot be assigned from the command line. It is inherited from the *resource_type* argument.

TCPAccess This parameter is not supported for OS/400.

Role Attributes

Children Specifies a list of role records that inherit resource access information from a parent role. This attribute is for referential integrity purposes only. It cannot be set. A role's **Children** attribute can only be changed by changing the **Parent** attribute in another role record.

OS4Groups, Specifies a list of platform-specific groups directly assigned to the role.

OS4ResAccess Specifies a list of platform-specific resources and associated permissions that have not previously been defined as resource records.

OS4TMEResAccess
Specifies a list of platform-specific resources and associated permissions that have previously been defined as Tivoli resource records.

Parent Specifies the parent record. The child role inherits resource access permissions from the parent role. Inherited resource access permissions can be overridden in the child record by setting a new value in the child record. You must specify a value in the format [*SecurityProfileName***:**]*ParentRoleName*

TMEGroups Specifies a list of Tivoli Security Management groups directly assigned to the role.

For NT groups, you must specify a domain-qualified value, in the format *domain_name\group_name,* if the group name is not unique among the global groups of all trusted domains. (A group record corresponds to an NT global group, and a role record corresponds to an NT local group.)

OS4ResTypeAccess
Specifies the access rights to all resources of a specific resource type that have not previously been defined as resource records.

You must specify values in the following format:

```
'resource_type perms(permissions,…)
[via(via_type:via_name,…)] …'
```

where:

resource_type Specifies the type of resource to which access will be globally allowed or disallowed for this role.

permissions Assigns access permissions.

System Policy Attributes

Lockout, OS4Lockout

Specifies whether user accounts are locked based on the number and frequency of failed login attempts, and for what duration the lockout should remain in effect. Valid values are **Never** or *number* where *number* specifies how many failed login attempts must occur before the account is locked.

MaxConLogins, OS4MaxConLogins

Specifies the maximum number of concurrent logins that a user can have open at the same time from each machine. Valid values are **None**, **Unlimited**, or any integer greater than zero.

MaxGraceLogins, OS4MaxGraceLogins

Specifies the maximum number of times that a user can log in after the user's password has expired. Valid values are **None**, **Unlimited**, or any integer greater than zero. For OS/400, if a number is specified it is mapped to unlimited.

MaxInactiveDays, OS4MaxInactiveDays

Locks the user's account if the user does not log in within the specified number of days. Valid values are **Never** or any integer greater than zero.

PwChk, OS4PwChk

Specifies whether password checking is enabled on subscribing endpoints. Valid values are **True** and **False**.

PwExpireWarning

Specifies whether a warning should be issued before a password expires. Valid values are **True** and **False**.

PwHistory, OS4PwHistory

Specifies the number of previous passwords to store to prevent the reuse of old passwords. Valid values for OS/400 are 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, and 32.

PwMaxAge, OS4PwMaxAge

Specifies the maximum number of days a user can go without changing the password. Valid values are **Unlimited** or any integer equal to or greater than zero.

PwMaxReps, OS4PwMaxReps

Specifies the maximum number of repetitive characters the password can contain. The value of this attribute can be:

0 for no repetitive characters

1 for no adjacent repeated characters

2 for no restrictions on repeated characters

PwMinAge, OS4PwMinAge

Specifies the minimum number of days that must elapse before the user can change the password. Valid values are **Now**, indicating the user can change the password at any time, or any integer greater than zero.

PwMinAlphanums, OS4PwMinAlphanums

Specifies the minimum number of alphanumeric characters (a-z, A-Z, 0-9) that must be present in the password. The value of this attribute must be an integer equal to or greater than zero.

PwMinAlphas, OS4PwMinAlphas

Specifies the minimum number of alphabetical characters (a-z, A-Z) that must be present in the password. The value of this attribute must be an integer equal to or greater than zero.

PwMinLen, OS4PwMinLen

Specifies the minimum number of characters that must be present in the password. The value of this attribute must be an integer equal to or greater than one and less than 10.

PwMinLowers, OS4PwMinLowers

Specifies the minimum number of lowercase alphabetical characters (a-z) that must be present in the password. The value of this attribute must be an integer equal to or greater than zero.

PwMinNums, OS4PwMinNums

Specifies the minimum number of numbers (0-9) that must be present in the password. The value of this attribute must be an integer equal to or greater than zero.

PwMinSpecials, OS4PwMinSpecials

Specifies the minimum number of special characters that must be present in the password. A special character is any non-alphanumeric character. The value of this attribute must be an integer equal to or greater than zero.

PwMinUppers, OS4PwMinUppers

Specifies the minimum number of uppercase alphabetical characters (A-Z) that must be present in the password. The value of this attribute must be an integer equal to or greater than zero.

OS4ResTypeAudit

Indicates whether attempts to access the resources of the specified type (or all resource types) should be logged. You must specify values in the following format:

```
'[resource_type | all] ([None] | [All] |
[Success] | [Failure]), …'
```

Examples

1. The following command sets values for the **OS4ResAccess** attribute for the **Development** role in the **SecProf** security profile.

   ```
   wmodsec Role –s \
   OS4ResAccess='FILE:/usr/local/bin/my_prog \
   perms(R,W,X)' SecProf Development
   ```

2. The following command adds members to the **Marketing** group.

   ```
   wmodsec Group –a OS4UserMembers='henry, marina' \
   SecProf Marketing
   ```

3. The following command replaces the user **psmith** with user **rdiaz** in all group records in which user **psmith** was defined as a OS4 member:

   ```
   wmodsec Group –r OS4UserMembers=psmith \
   –a OS4UserMembers=rdiaz SecProf
   ```

wsetfpopts

Sets or modifies the options for a file package.

Synopsis

wsetfpopts {**–T** *type* | **–t** *type*} {*keyword_options*} *fp_name*

Description

The **wsetfpopts** command manipulates file package options by setting keyword options, either to a specified value or to their default. It cannot manipulate the contents of a file package (such as **wsetfpcontents**), nor the configuration program information associated with a file package (like **wsetfpprgs**).

Authorization

admin, **senior**, or **super**

Arguments

keyword_options

Sets the keyword values for the file package. If used with the **–t** argument, the keyword is reset to its default value. If used with the **–T** argument, the keyword is set to a user-specified value as indicated below. Valid values are as follows:

path	A fully-specified path for a file.
y | **n**	**YES** or **NO**, respectively.
string	A command to be performed when appropriate for the specified keyword.
gid	The numeric group ID.
uid	The numeric user ID.
int	An integer.
path	A fully-specified path for a file or directory.
Pmode	A file permission setting (see the **chmod** command reference).

time — A **date** value for the date and time (conforming to **date** semantics).

mail_id — An E-mail address for a user; for example: **joe@work.com**.

log_host — Label of the managed node.

fp_name — Specifies an object path or registered name of the file package whose options are modified by this operation.

–t *type* — Sets the specified keyword option to its default value. The *type* argument specifies the type of platform on which the keyword is set. Software Distribution looks for default settings in the file package's policy region first. If the default is not set there, Software Distribution adopts the program defaults as described in the *Tivoli Software Distribution Reference Manual*.

Do *not* specify arguments with **–t** *type* keyword options—for example, **–t unix –v** does not require any arguments after the **–v** keyword option. Valid types are shown below, with their valid keyword options:

gen — All platforms; keyword options are not platform-specific (apply to all platforms). Valid keyword options are: **–a**, **–A**, **–b**, **–B**, **–c**, **–C**, **–d**, **–D**, **–e**, **–E**, **–f**, **–F**, **–h**, **–H**, **–i**, **–I**, **–j**, **–k**, **–l**, **–L**, **–m**, **–M**, **–n**, **–N**, **–o**, **–P**, **–q**, **–Q**, **–r**, **–R**, **–s**, **–S**, **–v**, and **–x**. See below for keyword option descriptions.

os400 — Keyword options that apply specifically to OS/400 target machines. The valid keyword options for **os400** are **–p** and **–K**. See below for keyword option descriptions.

–T *type* — Sets the specified keyword option to the specified argument. The *type* argument specifies the type of platform on which the keyword is set. The valid *type* for AS/400 is **os400** or **gen**.

General Keyword Options

Use the general following keyword options with the **–T gen** argument:

–a [**y** | **n**] Sets the **append_log** keyword, which toggles whether to append a notice to a log file when an operation is performed (**y**) or to replace the previous notice (**n**).

–A Sets the new keyword, **ams_inst_sw_comp_handle**, to denote the AMS Installed Software Component Handle, which identifies the installed software component handle created for the software component by the Tivoli Developer’s Toolkit. Software Distribution passes the value of this keyword as the third argument to before and after configuration programs.

–b *string* Sets the **backup_fmt** keyword, which specifies a backup path for files if they already exist at the target. For example, if you specify *string* to be **/backup/%p/%f%n**. **%p** provides the base path, **%f** provides the filename, and **%n** is an auto-incrementing numeric value. Thus, if you specify **/etc/motd** as the backup file, the resulting file’s path becomes **/backup/etc/motd1**. The next time the file is distributed, the resulting backup file is named **/backup/etc/motd2**, and so on.

–B *uid* Sets the **log_file_uid** keyword, which sets the UID of the log file specified by the **log_file** keyword.

–c [**y** | **n**] Sets the **do_compress** keyword, which toggles whether to compress the file package before distributing it (**y**) or not (**n**).

–C [**y** | **n**] Sets the **create_dirs** keyword. This keyword toggles whether to create intermediate directories at the destination for the files in a file package if those directories do not exist (**y**) or not (**n**).

–d [**y** | **n**] Sets the **descend_dirs** keyword, which toggles whether to traverse component directories in the file package listing (**y**) or not (**n**). If **y**, The directories and all files and directories therein are distributed.

–D *path* Sets the **default_dest** keyword, which specifies the default destination directory (entered as an absolute path) for all entries in the file package.

–e *mail_id* Sets the **mail_id** keyword, which is the complete E-mail address (such as joe@work.com) of person(s) to whom to send mail. For multiple mail recipients, separate addresses by a single space and enclose the collection of addresses in double quotes.

–E *string* Sets the **prog_env** keyword, which specifies a string as the **putenv** value before a configuration program runs on a UNIX machine. The *string* argument is a list of *name*=*value* pairs where *name* is an environment variable set by *value*. Refer to the *Tivoli Software Distribution Reference Manual* for more information on the **prog_env** keyword.

–f [**y** | **n**] Sets the **follow_links** keyword, which toggles whether to follow symbolic links to the original files (thereby copying the files to the target instead of the links) (**y**), or to create new links to the original files at the destination, and not distribute the files (**n**).

–F [**y** | **n**] Sets the **file_cksums** keyword, which toggles whether to compute the checksum on individual files to detect differences between the source and target (**y**).

–h *log_host* Sets the **log_host** keyword, which specifies the label of the managed node where the log file is generated.

–H *gid* Sets the **log_file_gid** keyword, which sets the GID of the log file specified by the **log_file** keyword.

–i *time* Sets the **default_mtime** keyword, which specifies the default modification time (conforming to **date** semantics) for all entries distributed in the file package.

–I [**y** | **n**] Sets the **install_progs** keyword, which toggles whether to remove the configuration programs on the target machine after they run. If you specify **y**, the programs are not removed.

–j *mode* Sets the **log_file_mode** keyword, which sets the file mode of the log file specified by the **log_file** keyword.

–k [**y** | **n**] Sets the **keep_paths** keyword, which toggles whether to concatenate the entire source file path into the destination file path (**y**). The default is **n**.

–l *path* Sets the **list_path** keyword, which specifies a directory on the target host where the list of distributed files to that target are stored. The file is named *fpname***.log**.

–L *path* Sets the **log_file** keyword, which appends notices to the specified complete path of a file if the **–a** argument (the **append_log** keyword) is set to **y**, or replaces notices if set to **n**. The managed node for this file is specified by the **–h** argument. If no host is specified, the TMR server is used. If the file does not exist, an attempt is made to create it.

–m *mode* Sets the **default_file_mode** keyword, which specifies the mode of files (according to **chmod** numeric specification) for files that are distributed.

–M *mode* Sets the **default_dir_mode** keyword, which specifies the mode of directories (according to **chmod (2)** numeric specification) for directories that are distributed.

–n [**y** | **n**] Sets the **no_overwrite** keyword, which toggles to not overwrite files existing on the target with those in the file package (**y**).

–N [**y** | **n**] Sets the **nested_first** keyword, which unpacks files within nested file packages first at the target (**y**). The default is to unpack them last (**n**).

–o [**y** | **n**] Sets the **skip_older_src** keyword, which does not distribute source files in the file package if they are older than those on the target (**y**).

–P [**y** | **n**] Sets the **post_notice** keyword, which toggles whether to post notices to the Software Distribution notice group when file package operations are performed (**y**).

–q *string* Sets the **postproc** keyword, which specifies a processing filter to be run before the file package is sent out from the host (but after it is packaged).

–Q *string* Sets the **preproc** keyword, which specifies a processing filter to be run before the distribution is unpacked at the target.

–r [**y** | **n**] Sets the **rm_empty_dirs** keyword, which toggles whether to remove all empty target directories, up to /*xxx***_platform_prefix/default_dest** at the target after the file package is removed (**y**) or not (**n**). The default is **n**.

–R [**y** | **n**] Sets the **rm_extraneous** keyword, which toggles whether to remove files in an existing destination directory that are extraneous to the file package (y) or not (n) during a distribution (of all file package entries). The default is **n**. If the **–b** argument (the **backup_fmt** keyword) is used, the removed files are first copied to a backup directory.

–s [**y** | **n**] Sets the **stop_on_error** keyword, which toggles whether to stop processing the distribution for a target if an error occurs during distribution to that target (**y**) or not (**n**). The default is **y**.

–S *path* Sets the **src_relpath** keyword, which specifies the absolute path for all relative paths used in the file package definition.

–v *int* Sets the **progs_timeout** keyword, which sets a timeout value for all configuration programs specified in the file package. Specify *int* in seconds; the default is -1, which enables the configuration program to run until completion (no timeout).

–x [**y** | **n**]
Sets the **do_checksum** keyword, which performs a checksum parity check on the file package when it is distributed.

OS/400 Keyword Options

–p *path*
Sets the **os400_platform_prefix** keyword, which prepends the specified path to all destination paths for distributions to OS/400 targets.

–K translate_to_EBCDIC
Sets the **os400_translate_to_ebcdic** keyword, which specifies if translation is needed. A value of y indicates that the file should be translated to EBCDIC on the AS/400 subscriber. A value of n, the default, indicates no translation. Text files should be translated.

Return Codes

The **wsetfpopts** command returns the following codes to standard output:

0 Successful completion.

1 The commands failed due to an error.

Examples

To use the **wsetfpopts** command to set the OS/400 translate to EBCDIC option in the **PS_docs** file package distributed to an OS/400 subscriber, enter:

```
wsetfpopts -T os400 -K y @FilePackage:PS_docs
```

where:

T os400
Enables you to set OS/400 keywords using this command.

K y
Specifies that the files should be translated to EBCDIC on the OS/400 subscriber.

wsetfpprgs

Sets or modifies the before, after, remove, after removal, commit, or on error program information associated with a file package.

Synopsis

wsetfpprgs {**–t** *type*| **–T** *type*} [*keyword_options*] *fp_name*

Description

This command enables you to specify the configuration program to be run on the source host or target. If you specify multiple configuration programs of the same type, you must separate them with commas. Likewise, to specify input files for multiple programs, you must also separate them with commas. The programs must reside on the same target or source and are processed in the order in which you specify them. If a program does not require an input file but subsequent programs do, you must specify the input files in the correct order and preceded by the same number of commas as the programs. Do not follow commas with spaces.

If you specified multiple before programs and one fails, subsequent before programs will run if you did not enable the **Stop distribution on error** option on the **File Package Properties** window (**os400_before_skip_non_zero=y**). Otherwise, the distribution stops and subsequent programs will not run. If you specified multiple programs of any other type and one fails, the distribution stops and Software Distribution will not run subsequent programs.

The **wsetfpprgs** command also enables you to specify whether programs and input files reside on the source host or a target. If the programs and input files reside on the source host (using the **–A**, **–B**, **–C**, **–D**, **–E**, **–F**, **–G**, **–I**, **–J**, **–K**, **–L**, **–N**, and **–R** arguments), you can specify a relative path to them. Software Distribution runs the programs and obtains the input files from the path specified by the **src_relpath** keyword. If the programs and input files reside on the target, you must specify a full path to each. By default, programs that run on a target reside on that target. Source before and after programs reside on the source.

The **wsetfpprgs** command cannot manipulate the files of a file package (**wsetfpcontents**), nor the file package options (**w4inslcf.pl**).

Authorization

admin, **senior**, or **super**

Arguments

–t *type*
: Sets the specified keyword option to its default value. The *type* argument specifies the type of platform on which the program is run. Software Distribution looks for default settings in the file package's policy region first. If the default is not set there, Software Distribution adopts the defaults as described in "File Package Definitions" Chapter of the Software Distribution Reference Manual. Do not specify arguments with the keyword option.

Valid types are shown below, with their valid keyword options:

src
: Keyword options that apply to running a program on the source host. Valid keyword options for **src** are: **–a**, **–b**, **–i**, **–s**, **–u**, and **–U**. See below for keyword option.

os400
: Keywords that apply to running a program on an OS/400 target machine. Valid keyword options for **os4** are: **–a**, **–A**, **–b**, **–B**, **–c**, **–C**, **–d**, **–D**, **–e**, **–E**, **–f**, **–F**, **–g**, **–G**, **–i**, **–I**, **–j**, **–J**, **–k**, **–K**, **–l**, **–L**, **–r**, **–R**, and **–s**. See below for keyword option descriptions.

–T *type*
: Sets the specified keyword option to the specified argument. The *type* argument specifies the type of platform on which the program is run. The valid value is os400.

keyword_options

Specifies the configuration program keyword values to be set for the file package. If used with the **–t** specifier, the keyword is reset to its default value. If used with the **–T** specifier, the keyword is set to a user-specified value as indicated below. Valid values are as follows:

path	A fully-specified path for a file.
y | **n**	**YES** or **NO**, respectively.
uid	A numeric user ID.

Each of the keyword options can specify different keywords, depending on the type specified with the **–T** argument. For this reason, *xxx* is used in the keyword description in place of the host or target type.

For example, for the keyword option **–a** (listed as *xxx***_after_prog_path**) the keyword is set to **src_after_prog_path** if you specify type as **src**.

Specific keywords are described in detail in the “File Package Definitions” chapter of the *Software Distribution Reference Manual*.

–a *path*
Sets the *xxx***_after_prog_path** keyword, which runs *path* on each target (or the source host) after a file package’s files are applied.

–A [**y** | **n**]
Sets the *xxx***_after_prog_from_src** keyword, where **y** indicates that *xxx***_after_prog_path** resides on the source host, and **n** that it reside on each target.

–b *path*
Sets the *xxx***_before_prog_path** keyword, which runs *path* on each target (or the source host) before a file package’s files are applied.

–B [**y** | **n**]
Sets the *xxx***_before_prog_from_src** keyword, where **y** indicates that

*xxx***_before_prog_path** resides on the source host, and **n** that it resides on each target.

–c *path*

Sets the *xxx***_commit_prog_path** keyword, which runs *path* on each target during a commit operation.

–C [**y** | **n**]

Sets the *xxx***_commit_prog_from_src** keyword, where **y** indicates that *xxx***_commit_prog_path** resides on the source host, and **n** that it resides on each target.

–d *path*

Sets the *xxx***_after_removal_prog_path** keyword, which runs *path* on the target after removing the file package.

–D [**y** | **n**]

Sets the *xxx***_after_removal_prog_from_src** toggle, where **y** indicates that *xxx***_after_removal_prog_path** resides on the source host, and **n** that it resides on each target.

–e *path*

Sets the *xxx***_after_removal_input_path** keyword, which passes *path* as standard input (or **arg[2]** on AS/400s) to *xxx***_after_removal_prog_path**. This keyword is only applicable if the **–d** argument is set.

–E [**y** | **n**]

Sets the *xxx***_after_removal_input_from_src** toggle, where **y** indicates that *xxx***_after_removal_input_path** resides on the source host, and **n** that it resides on each target.

–f *path*

Sets the *xxx***_on_error_prog_path** keyword, which runs *path* on the target if an error stops a file package distribution.

–F [**y** | **n**] Sets the *xxx***_on_error_prog_from_src** toggle, where **y** indicates that *xxx***_on_error_prog_path** resides on the source host, and **n** that it resides on each target.

–g *path* Sets the *xxx***_on_error_input_path** keyword, which passes *path* as standard input (or **arg[2]** on AS/400s) to *xxx***_on_error_prog_path**. This keyword is only applicable if the **–f** argument (*xxx***_on_error_prog_path**) is set.

–G [**y** | **n**] Sets the *xxx***_on_error_input_from_src** toggle, where **y** indicates that *xxx***_on_error_input_path** resides on the source host, and **n** that it resides on each target.

–i *path* Sets the *xxx***_after_input_path** keyword, which passes *path* as standard input (or **arg[2]** on AS/400s) to *xxx***_after_prog_path**. This keyword is only applicable if the **–a** argument (*xxx***_after_prog_path**) is set.

–I [**y** | **n**] Sets the *xxx***_after_input_from_src** keyword, where **y** indicates that *xxx***_after_input_path** resides on the source host, and **n** that it resides on each target.

–j *path* Sets the *xxx***_before_input_path** keyword, which passes *path* as standard input (or **arg[2]** on AS/400s) to *xxx*_**before_prog_path**. This keyword is only applicable if the **–b** argument (*xxx*_**before_prog_path**) is set.

–J [**y** | **n**] Sets the *xxx*_**before_input_from_src** keyword, where **y** indicates that *xxx*_**after_prog_path** resides on the source host, and **n** that it resides on each target.

–k *path* Sets the *xxx*_**commit_input_path** keyword, which passes *path* as standard input (or **arg[2]** on AS/400s) to *xxx*_**commit_prog_path**. This keyword is applicable only if the **–c** argument is set.

–K [**y** | **n**] Sets the *xxx*_**commit_input_from_src** keyword, where **y** indicates that *xxx*_**commit_input_path** resides on the source host, and **n** that it resides on each target.

–l *path* Sets the *xxx*_**removal_input_path** keyword, which passes *path* as standard input (or **arg[2]** on AS/400s) to *xxx*_**removal_prog_path**. This keyword is applicable only if the **–r** argument is set.

–L [**y** | **n**] Sets the *xxx*_**removal_input_from_src** toggle, where **y** indicates that *xxx*_**removal_input_path** resides on the source host, and **n** that it resides on each target.

—r *path* Sets the *xxx*_**removal_prog_path** keyword, which runs *path* on the target before removing the file package from the subscribers.

–R [**y** | **n**] Sets the *xxx***_removal_prog_from_src** keyword, where **y** indicates that *xxx***_removal_prog_path** resides on the source host, and **n** that it resides on each target.

–s [**y** | **n**] Sets the *xxx***_before_skip_non_zero** keyword, where **y** indicates that distribution is skipped if *xxx***_before_prog_path** exits with a non-zero exit code, and **n** indicates that it is not skipped.

fp_name Specifies an object path or registered name of the file package, as an absolute path or as a registered name.

Return Codes

The **wsetfpprgs** command returns the following codes to standard output:

0 Successful completion.

1 The commands failed due to an error.

Examples

1. To set the value of the **os400_after_prog_path** keyword, enter the following commands:

```
wsetfpprgs -T os400
-a /qsys.lib/qgpl.lib/myprog.pgm @example_fp
```

where:

–T os400 Enables you to set the OS/400 options on the command line.

–a /qsys.lib/qgpl.lib/myprog.pgm
Sets the after program for the OS/400 subscribers to the **myprog** program in the **qgpl** library.

@example_fp Specifies to modify the example_fp file package.

2. To reset the value of the **os400_after_prog_path** keyword back to its default, enter:

   ```
   wsetfpprgs -t os400 -a @example_fp
   ```

Index