User's Guide

NMServer/400

SNMP System Management Agent

for OS/400

COMTEK
NMServer SNMP Products

NMServer/400
for OS/400

Product Data

Date: **March 2006**

NMServer Version: **3.00**

OS/400 Version Required **V4R5 or higher**

NMServer
also available for
OpenVMS and Stratus VOS

COMTEK Contact Information

For more information or technical support on the NMServer product suite, contact:

COMTEK Services, Inc.
101 South Whiting St.
Suite 300
Alexandria, VA 22304

Sales Phone: (603) 881-9556
Sales Fax: (603) 881-5504
Support Phone: (703) 751-3997
sales@COMTEKServices.com
support@COMTEKServices.com
www.comtekservices.com

Table of Contents

1

Overview of NMServer for OS/400

NMServer Product Description

NMServer for OS/400 is an application which allows AS/400 systems to be monitored and managed by SNMP managers such as Bull ISM, HP OpenView, or IBM NetView/6000.

Simple Network Management Protocol (SNMP) is an Internet Standard Protocol designed to address the problem of managing complex networks. Since its adoption in 1988, SNMP has become the de facto standard with thousands of installations worldwide. Many organizations have SNMP managers in place managing network elements such as LANs, routers, and communications facilities. Only recently have products become available which allow computer systems to be included as managed elements.

NMServer for OS/400 provides SNMP instrumentation which allows AS/400 systems to be managed by the same SNMP managers presently installed, or by a new manager installed specifically for this purpose.

The NMServer for OS/400 product includes the NMServer application, MIB, documentation, sample configuration files, and product support.

NMServer Architecture

SNMP network management is composed of two major components - a manager which runs on a management station and an agent which runs in each of the devices to be managed. A variety of terms are commonly used for such SNMP managers as OpenView and NetView. This manual refers to these packages simply as SNMP managers.

In order to facilitate running multiple agents in a single managed device, the concept of extensible agents was developed. An extensible agent is composed of a single master agent which handles SNMP operations with the SNMP manager and one or more subagents which instrument the managed information for the device. IBM supplies a Distributed Protocol Interface

(DPI) master agent with the OS/400 operating system. NMServer for OS/400 is implemented as DPI subagent. In an extensible agent environment, all subagents operate independently of one another and may be stopped and started without restarting the master agent.

NMServer Subagent Operation

NMServer gathers and maintains critical data about the operation of the AS/400 system on which it is running. This information is represented in a MIB (Management Information Base), and is available to the SNMP manager whenever polled. The information available includes the following:

- CPU utilization percentage
- Number of jobs in the system
- Auxiliary storage pool (ASP) and percentage used
- Processes running and statistics regarding each process
- Jobs in the queue
- Status of the jobs
- User statistics
- Batch job statistics
- Line, controller, and device configuration and status information

NMServer Traps

NMServer uses trap messages to instantly alert the SNMP manager of critical situations on the managed system. The following are some of the traps generated:

- A user specified critical process is not running
- Auxiliary storage pool (ASP) usage exceeds the user specified threshold or has returned to an acceptable level
- CPU usage exceeds the user defined threshold
- An important QSYSOPR message was received
- An important message was received on a user specified message queue
- A user defined message was received

The user defined thresholds for ASP full, CPU overload, and many other parameters which govern the operation of NMServer can be established at

NMServer startup or modified while NMServer is running by using an SNMP set-request from the SNMP manager.

The confirmed trap reception feature permits retransmission of all traps generated by the subagent. Each subagent trap is assigned a sequence number which may be monitored by the SNMP manager. If the SNMP manager detects that a trap was not received, it may use an SNMP set-request to initiate retransmission of the missing trap.

NMServer Remote Console

The NMServer remote console feature allows OS/400 console commands to be entered from the SNMP manager. The subagent validates these commands, executes them, and optionally returns the output to the SNMP manager as a series of traps.

2

Product Installation

The following steps are required to install and configure NMServer for an AS/400 system.

1. Check that the prerequisite software is installed and that the authorization requirements are met.
2. Restore the COMTEK1 library files onto the AS/400 system.
3. Load the nmserveras400.mib MIB file onto the SNMP manager system.
4. Customize the default configuration files:

   COMTEK1/COMTEKCFG(INIT_CFG)
   COMTEK1/COMTEKCFG(CRIT_PROC1)
   COMTEK1/COMTEKCFG(SKIP_MSGID)
   COMTEK1/COMTEKCFG(Q_MON_CFG)

Before Installation

Before running NMServer for OS/400, the system must have the following prerequisite software installed and the following authorization requirements set.

Prerequisite Software

The list of all IBM AS/400 PTFs and their descriptions is available at:

http://as400service.ibm.com/as4sde/as4ptf.nsf

Check this source for the latest PTF information. To find the appropriate PTFs, search for "SNMP".

The WRKPTF OS/400 command may be used to determine if the required PTFs have been installed on an AS/400 system. COMTEK strongly recommends that all applicable PTFs be installed on each target OS/400

system to ensure proper operation of the AS/400 system and the NMServer product.

Authorization Requirements

The user profile under which the COMTEK subagent runs requires read, write, and create access to the COMTEK1 library. In addition, *JOB_CTL authority is required in order to check on active processes and job queues.

Installation

This section describes how to install the COMTEK NMServer for the OS/400 product. This product may either be provided on tape, CD, or may be retrieved from the COMTEK FTP site. Depending on how the product is provided, follow the appropriate instructions for installing the COMTEK1 library.

Restoring the NMServer Library Files from Tape

All of the COMTEK OS/400 subagent objects are located in the COMTEK1 library. The product is shipped as a saved library and is installed with the RSTLIB command. The following steps restore the product to the AS/400 system.

1. Insert the COMTEK NMServer product tape into the AS/400 tape drive.
2. Enter the following command:

   RSTLIB COMTEK1 DEV(TAPxx)

   In this command, TAPxx is the name of the tape drive.

Restoring the NMServer Library Files from CD

All of the COMTEK OS/400 subagent objects are located in the COMTEK1 library. The product is shipped as a saved library on a CD that can be read by a standard PC. The product is installed by transferring the save file to the AS/400 and using the RSTLIB command. The following steps restore the product to the AS/400 system.

1. Create save file on your AS/400 by issuing the following commands:

CRTSAVF FILE(yourlib/NM400SAV2)
TEXT('NMServer/400 Version 2 distribution save file')

Note: yourlib can be any library of your choice.

2. Mount the distribution CDROM in your PC.
3. Ftp the save files from your PC to the AS/400. Make sure you use a binary transfer.
4. Restore the objects to the COMTEK1 library by issuing the following command:

RSTLIB LIB(COMTEK1) SAVLIB(COMTEK1) DEV(*SAVF) SAVF(yourlib/NM400SAV2)

Acquiring the NMServer Library Files from the COMTEK FTP Site

To retrieve instructions for downloading NMServer from the COMTEK FTP site, complete the NMServer for AS/400 product download form located at http://www.comtekservices.com/as400_downloadform.htm. Follow the directions contained in rdme400.htm for installing the downloaded product.

Loading the COMTEK MIB

Transfer the MIB file nmserveras400.mib to the network management station and load it into the SNMP manager according to the instructions supplied with the manager. This file can be found in the mib directory on the product CD or in the mib directory in the NMSERVER400.ZIP installation kit.

Customizing the Configuration Files

The configuration files are customized by editing the initial configuration file, critical process file and the QSYSOPR Message ID filter file.

Refer to Chapter 3 of this manual for more information on the format and content of each of the configuration files.

Editing the Initial Configuration File

Edit the file:

COMTEK1/COMTEKCFG(INIT_CFG)

If the license key has not been entered by COMTEK, it must be entered.

When the COMTEK1/COMTEKCFG(INIT_CFG) file is displayed or edited, there is a line that looks like:

LICENSE_KEY=##################

If this line does not contain anything after the equal sign (=), the key must be entered. If no key or an invalid key is in the file, the NMServer subagent will not start. It will send a message to the QSYSYOPR message queue stating that a valid license key is required to use this product.

Modify any other values to replace the default values.

Editing the Critical Process File

As shipped, the INIT_CFG file is configured to use COMTEK1/COMTEKCFG(CRIT_PROC1) as the list of critical processes. The file may be edited or the user can create a new file and point the INIT_CFG file at the new critical process file. To modify the default critical process file, edit the file:

COMTEK1/COMTEKCFG(CRIT_PROC1)

Edit this file only if there are any other values to replace the default values. If the NMServer subagent is started by a user other than QSECOFR, the sample critical process file should be modified to prevent critProcMissing traps for the subagent process. Otherwise, no edits are required in this file.

Editing the QSYSOPR Message ID Filter File

Edit the file:

COMTEK1/COMTEKCFG(SKIP_MSGID)

Edit this file only if there are any other values to replace the defaults. No edits are required in this file. Enter any other values to replace the default values. Note that this is the default name for the file. The user can configure NMServer/400 to use any valid file member as the configuration file for filtering QSYSOPR messages based on message ID.

Editing the Message Queue Monitor Configuration File

Edit the file:

COMTEK1/COMTEKCFG(Q_MON_CFG)

Edit this file to designate which user specified message queues should be monitored by NMServer/400. By default, the queue monitor does not monitor any message queues. Also, this file contains message filtering

configuration parameters. See chapter 3, Configuration Files, for a full description of this file.

Editing the Message Queue Monitor Message ID Filter File(s)

NMServer/400 does not use a specific file member name for configuring message ID filtering for user specified message queues. Instead, the user should create a new configuration file for each message queue to be monitored. The format of the configuration file is identical to that of the QSYSOPR Message ID filter file, which can be used as a template. See chapter 3, Configuration Files, for a full description of this file.

Changing the Configuration While the Subagent is Running

All of the configuration tokens are available as MIB variables and can be changed with an SNMP set-request except:

- MAX_RETAINED_TRAPS
- GET_TABLE_BY_ROW
- QSYSOPR_ENFORCE_RETRY_LIMIT
- QSYSOPR_RETRY_LIMIT
- DELETE_ANSWERED_MESSAGE
- CONVERT_TO_ASCII

When using the SNMP set-request to change the configuration, select the appropriate MIB variable and set the new value. Do not enter the configuration token or the equal sign (=) when using an SNMP set-request, as these are only used in the initial configuration file.

All changes to the configuration that are made with SNMP set-requests take effect as soon as the set-request processing is complete. Changes made while the subagent is running remain in effect even if the subagent is stopped and restarted.

Starting the IBM Master Agent

To start the master agent, enter the following command:

```
STRTCPSVR SERVER(*SNMP)
```

Note that the authorization information and trap destinations are defined in configuration files for the IBM master agent. Consult the IBM master agent documentation for information on how to modify this data.

Starting the NMServer Subagent

To start the NMServer subagent, enter the following command:

COMTEK1/STRCOMTEK

By pressing enter, the command executes with the default options. By pressing PF4 instead, it brings up the following command prompt which allows the selection of which NMServer subagent modules are started by this command.

```
                Start COMTEK SNMP Sub-Agent (STRCOMTEK)

Type choices, press Enter.

Start kernel (COMTEKSNMP)?  *YES    Character value, *YES, *NO
Start COMTEKCPU? . . . . .  *YES    Character value, *YES, *NO
Start COMTEKDSK? . . . . .  *YES    Character value, *YES, *NO
Start COMTEKPSA? . . . . .  *YES    Character value, *YES, *NO
Start COMTEKPSB? . . . . .  *YES    Character value, *YES, *NO
Start COMTEKQSYS?  . . . .  *YES    Character value, *YES, *NO
Start COMTEKQHST?  . . . .  *NO     Character value, *YES, *NO
Start COMTEKUTRP?  . . . .  *YES    Character value, *YES, *NO
Start COMTEKRCONS? . . . .  *YES    Character value, *YES, *NO
Start COMTEKCMN  . . . . .  *YES    Character value, *YES, *NO
Start COMTEKGENQ?  . . . .  *YES    Character value, *YES, *NO
Start SNMP DPI debugging?   *NO     Character value, *YES, *NO
```

The kernel must be running for all of the remaining modules except COMTEKCMN to be doing meaningful work.

The last parameter allows DPI debugging to be enabled. DPI debugging produces a print file that contains all of the transactions between the master agent and the subagent as well as setting the OS/400 logging level to the maximum level for all of the NMServer processes.

Note: The subagent cannot be started unless the IBM supplied OS/400 master agent is running.

The following table describes the remaining modules.

Module	Description
COMTEKCPU	Gathers CPU information.
COMTEKDSK	Gathers disk information.
COMTEKPSA	Reports on active processes.
COMTEKPSB	Monitors jobs in the job queues waiting to execute, executing, and jobs on the output queue.
COMTEKQSYS	Monitors the QSYSOPR message queue.
COMTEKHST	Monitors the QHST log. This module is generally not used.
COMTEKUTRP	Handles user generated traps.
COMTEKRCONS	Performs remote console functions.
COMTEKCMN	Reports on lines, controllers, and devices.
COMTEKGENQ	Monitors user specified message queues

Once the kernel has been started, the other modules may be started and stopped without stopping the kernel. To start one of the modules that is down, type COMTEK1/STRCOMTEK and enter PF4 to bring up the menu. Select *YES for the module that is to be started and *NO for all other modules (answering *YES to a module that is already running will cause a second instance of the module to be started). The OS/400 command WRKACTJOB may be used to determine which modules are running.

Troubleshooting Guide

Failure to respond to SNMP commands. Verify the IBM master agent is running. Check this by looking at the Contact information variables from the SNMP manager. These variables are supported by the IBM master agent. Otherwise, use the WRKACTJOB SBS(QSYSWRK) command and look for the QSNMPSA, QTMSNMP, and QTMSNMPRCV jobs. If these jobs are not present, restart the IBM agent with the command STRTCPSVR SERVER(*SNMP).

Verify that the NMServer subagent is running. Issue the command WRKACTJOB SBS(QSYSWRK) JOB(COM*) and look for the COMTEKSNMP job. If it is not started, it can be started with the STRCOMTEK command.

Verify that the community name is correct. Use the IBM supplied command CHGSNMPA to check the community name. Make sure you prompt (press F4) on the command instead of pressing enter. Check the case of the

community name. If the community name was not in single quotes(') when it was entered, the community name was probably capitalized by the operating system.

Verify that the license key is valid and has not expired. If you start the subagent, then do a WRKACTJOB and cannot find the COMTEKSNMP module, check the QSYSOPR message queue for an error message. DSPMSG(QSYSOPR). If you see UCS0070 "The license key you have entered is invalid…" or UCS0071 "The trial period for COMTEK NMServer/400 has expired…", ensure that the key was correctly entered. If the key has expired, contact COMTEK to either arrange for a new demo key or purchase the product to obtain a permanent key.

Wrong IP address appears in traps. Verify that the local domain and host names are properly set and that the host table contains the correct IP address and hostname for the AS/400. If there is no entry in the hosts table that matches the local host name, the loopback address will be used. To check these configuration points, use the CFGTCP command and choose option 10 "Work with TCP/IP host table entries" and option 12 "Change local domain and host names". The name in the host table entry should be of the form:

<host name>.<domain name>.
For example, comtek400.comtekservices.com where comtek400 is the hostname and comtekservices.com is the domain name. Note: Some versions of the IBM master agent expect to see a domain name. If your company does not use domain names, you will need to make one up in order to get the correct address in traps.

Long delays in getting traps. The system is missing one or more PTFs for the IBM SNMP (master) agent. Install the appropriate PTFs. Get the PTF information from COMTEK1/COMTEKCFG(PTF_LIST) or from IBM support.

Cannot retrieve QHST table. By default, this data collection module is not started when the subagent is started. In order to start this module, press PF4 on the COMTEK1/STRCOMTEK command and specify *YES for starting the COMTEKQHST data collector.

SNMP Sets do not work. If the IBM master agent and COMTEK subagent are both up, the problem is most likely the community name. The community name is case sensitive on the AS/400. Note: OS/400 likes to capitalize text, where on most UNIX platforms, the text is left as lowercase.

Traps are not sent. Verify that the IBM master agent is running. Check this by looking at the Contact information variables from the manager. These variables are supported by the IBM master agent. Otherwise, use the WRKACTJOB SBS(QSYSWRK) command and look for the QSNMPSA, QTMSNMP, and QTMSNMPRCV jobs. If these jobs are not present, restart the IBM master agent with the command STRTCPSVR SERVER(*SNMP).

Verify that the NMServer subagent is running. Issue the command WRKACTJOB SBS(QSYSWRK) JOB(COM*) and look for the COMTEKSNMP job. If it is not started, it can be started with the STRCOMTEK command.

Verify that the license key is valid and has not expired. If you start the subagent, then do a WRKACTJOB and cannot find the COMTEKSNMP module, check the QSYSOPR message queue for an error message. DSPMSG(QSYSOPR). If you see UCS0070 "The license key you have entered is invalid…" or UCS0071 "The trial period for COMTEK NMServer/400 has expired…", ensure that the key was correctly entered. If the key has expired, contact COMTEK to either arrange for a new demo key or purchase the product to obtain a permanent key.

Verify that the trap community contains your SNMP manager's IP address. Use the CHGSNMPA command to modify the trap community. Remember to prompt (press F4) on the CHGSNMPA command instead of pressing enter.

Verify that the trap throttle is running. Use WRKACTJOB SBS(QSYSWRK) JOB(COM*) and look for COMTEKTHRT. If the COMTEKTHRT process is missing, stop the subagent with the ENDCOMTEK command and restart it with the STRCOMTEK command.

Stopping the NMServer Subagent

The subagent is stopped by entering the command:

COMTEK1/ENDCOMTEK

To stop just one of the NMServer modules, use the F4 key to prompt on the ENDCOMTEK command to get the following menu:

```
          End COMTEK SNMP Sub-Agent (STRCOMTEK)

Type choices, press Enter.
```

```
End kernel (COMTEKSNMP)?  *YES    Character value, *YES, *NO
End COMTEKCPU? . . . . .  *YES    Character value, *YES, *NO
End COMTEKDSK? . . . . .  *YES    Character value, *YES, *NO
End COMTEKPSA? . . . . .  *YES    Character value, *YES, *NO
End COMTEKPSB? . . . . .  *YES    Character value, *YES, *NO
End COMTEKQSYS?  . . . .  *YES    Character value, *YES, *NO
End COMTEKQHST?  . . . .  *YES    Character value, *YES, *NO
End COMTEKUTRP?  . . . .  *YES    Character value, *YES, *NO
End COMTEKRCONS? . . . .  *YES    Character value, *YES, *NO
End COMTEKCMN  . . . . .  *YES    Character value, *YES, *NO
End COMTEKGENQ?  . . . .  *YES    Character value, *YES, *NO
```

Type *NO for the modules which should not be stopped and press enter.

3

Configuration Files

Distribution Media

Product Components for the SNMP Manager

NMServer for OS/400 provides the nmserveras400.mib MIB file for use on the SNMP manager. This file is in the mib directory on the product CD or in the NMSERVER400.ZIP installation kit on the COMTEK FTP site. This file is an ASCII text file.

Product Components for the NMServer for OS/400

The COMTEK1 library contains the following types of objects:

- Program (*PGM)
- Modules (*MODULE)
- Service program (*PGM)
- Data queue (*DTAQ)
- User space (*USRSPC)
- Source physical file (*SRCPF)
- Database files (*PF)
- Commands (*CMD)
- Message file (*MSGF)

Program Components

Component	Description
COMTEKSNMP	Main subagent kernel
COMTEKCMN	Subagent kernel for comm devices
COMTEKGENQ	Subagent kernel for queue monitor
CMTKBLDQ	Rebuilds the internal queues
CMTKCLRQ	Clears the internal queues

CMTKCMNC	Communications device data collection
CMTKCMNQ	Creates internal data queues
CMTKCPU	CPU data collection
CMTKDSK	Disk data collection
CMTKDSKCL	Disk data collection support program
CMTKPSA	Active process data collection
CMTKPSB	Job queue data collection
CMTKQHST	QHST data collection
CMTKQHSTCL	QHST data collection support program
CMTKQMON	Monitors user specified message queues
CMTKQSYS	QSYSOPR message monitor
CMTKRCON	Remote console program
CMTKTHRT	Trap throttle program
CMTKUTRP	User trap support
ENDCMTK2	Stops the subagent
STRCMTK	Starts the subagent

Module Components

Component	Description
CMTKGET	Get processing
CMTKGTNX	Get next processing
CMTKSNMP	Subagent kernel functions
CMTKGENQ	Subagent kernel functions
CMTKCMN	Subagent kernel functions

Service Program Components

Component	Description
CMTKGTNXSP	Inter-module communication
CMTKGETSP	between these modules and
CMTKSNMPSP	the subagent calling program

Data Queue Components

Component	Description
CFGCMDDTAQ	Subagent kernel command queue
CMNCMDDTAQ	Subagent kernel command queue
CMNDTAQ	Data queue for configuration data
CKTSATRPQ	Internal data queue for trap processing
CMTKRCONQ	Data queue for returning remote console data

CPUDATQ	Data queue for CPU data
CPUCMDDTAQ	CPU command queue
CTKCMDDTAQ	Subagent kernel command queue
CTKTRPDTAQ	Internal trap queue
CTSAQ	Data queue between the subagent and the IBM master agent
CTCFGSTSAQ	Data queue between the subagent and the IBM master agent
CTGENQSAQ	Data queue between the subagent and the IBM master agent
DSKCMDDTAQ	Disk command queue
DSKDTAQ	Disk data queue
HSTCMDDTAQ	QHST command queue
HSTDTAQ	QHST data queue
PSACMDDTAQ	Active process command queue
PSADTAQ	Active process data queue
PSBDTAQ	Job queue data queue
PSBCMDDTAQ	Job queue command queue
QMONCMD00	Internal data queue
QMONCMD01	Internal data queue
QMONCMD02	Internal data queue
QMONCMD03	Internal data queue
QMONCMD04	Internal data queue
QSYCMDDTAQ	QSYSOPR monitor command queue
QSYDTAQ	QSYSOPR monitor data queue
SNMP_USR_Q	Queue to accept user trap information
THRCMDDTAQ	Data queue to send commands to the trap throttle
USRCMDDTAQ	User trap command queue

User Space Components

Component	Description
CMTKSPCA1	Repository for active process data
CMTKSPCA2	Repository for active process data
CMTKSPCB1	Repository for job queue data
CMTKSPCB2	Repository for job queue data
CMTKSPCC1	Repository for line data
CMTKSPCC2	Repository for line data
CMTKSPCC3	Repository for controller data

CMTKSPCC4	Repository for controller data
CMTKSPCC5	Repository for device data
CMTKSPCC6	Repository for device data

Source File and Member Components

Component	Description
COMTEKCFG(CRIT_PROC1)	Sample critical process file
COMTEKCFG(INIT_CFG)	Configuration parameters at subagent startup
COMTEKCFG(PTF_LIST)	List of AS/400 PTFs required to be installed
COMTEKCFG(Q_MON_CFG)	Configuration file for user specified queues
COMTEKCFG(SKIP_MSGID)	Sample message filter file

Database File Components

Component	Description
DISKINFO	Temporary repository for disk data
QHSTINFO	Temporary repository for QHST data

Command Components

Component	Description
COMTEKBLDQ	Command to rebuild all of the subagent data queues. Only use on instruction from COMTEK
ENDCOMTEK	Ends the COMTEK SNMP subagent
STRCOMTEK	Starts the COMTEK SNMP subagent
DUMPTRAPS	Dumps the trap table to a text file for problem determination

Message File Components

Component	Description
COMTEKMSG	Subagent message file

Sample Configuration Files

Initial Configuration File

The initial execution parameters are read in at subagent startup from the initial configuration file. Many of the execution parameters may be dynamically changed without stopping the subagent by using an SNMP set-request. When parameters are changed via SNMP set-requests, the new value remains in force, even if the subagent is stopped. The initial execution parameters are contained in the file:

COMTEK1/COMTEKCFG(INIT_CFG).

The COMTEK1/COMTEKCFG(INIT_CFG) file is contained in the COMTEK1 saved library. An example file is shown below:

```
#
# THIS IS AN EXAMPLE INITIAL CONFIGURATION FILE FOR THE
COMTEK
# SNMP SUBAGENT
# COPYRIGHT COMTEK SERVICES, INC. 1999
#
# THE FOLLOWING LINE CONTAINS THE LICENSE KEY SUPPLIED BY
# COMTEK
LICENSE_KEY=##########
#
# SETTING CONVERT_TO_ASCII TO 1 CAUSES TRANSLATION, 0 LEAVES
# THE DATA AS EBCDIC
CONVERT_TO_ASCII=1

CMN_WAIT_TIME=600
CPU_RESEND_TRAP_COUNT=5
# THE CPU TRAP THRESHOLD IS SPECIFIED IN TENTHS OF A PERCENT
# THUS 95% WOULD BE SPECIFIED AS 950
CPU_TRAP_THRESHOLD=950
CPU_WAIT_TIME=300
#DELETE_ANSWERED_MSG=0
# DISK FULL REFERS TO SYSTEM ASP (AUX STORAGE POOL) AND IS
# SPECIFIED IN TEN-THOUSANDTHS OF A PERCENT, THUS 95% WOULD
BE
# SPECIFIED AS 950000
DISK_FULL_CLR_THRESHOLD=850000
DISK_FULL_THRESHOLD=900000
DISK_RESEND_TRAP_COUNT=5
DISK_WAIT_TIME=300
GET_TABLE_BY_ROW=0
# TRAP THROTTLING
HUNDREDTHS_SEC_BETWEEN_TRAPS=25
INITIAL_REQ_PS_FILENAME=COMTEK1/COMTEKCFG(CRIT_PROC1)
JQ_WAIT_TIME=300
MAX_RETAINED_TRAPS=200
PS_WAIT_TIME=300
QHST_WAIT_TIME=1000
```

```
# QSYSOPR MESSAGE FILTERING PARAMETERS
QSYSOPR_MSG_MAX_AGE_MINUTES=2
MIN_QSYSOPR_SEV_TO_SEND=0
QSYSOPR_MSG_FILTER_FILE=COMTEK1/COMTEKCFG(SKIP_MSGID)
QSYSOPR_RETRY_LIMIT=375
QSYSOPR_ENFORCE_RETRY_LIMIT=1
# REMOTE CONSOLE THROTTLE
RCONS_TRAPS_PER_SECOND=4
#
# INTERNAL TIMING TUNING KNOBS - DO NOT CHANGE WITHOUT
# EXPLICIT INSTRUCTIONS FROM COMTEK SERVICES, INC.
#
SET_COMMIT_QUENCH_TIMEOUT=5
```

The format of this file is:

CONFIGURATION_TOKEN = VALUE

The entries in the configuration file must obey the following rules:

- Any line which begins with the pound character (#) is considered a comment.
- The configuration tokens must be capitalized.
- There should be no space between the token name and the equal sign (=) or between the equal sign and the token value.

Note: The presence of the configuration parameters in the INIT_CFG file is considered mandatory for the proper operation of the subagent. Therefore, the user should only modify the settings and not comment out or delete any of the keyword parameters in the file. If particular functionality is not needed, disable the associated module when starting the subagent as described in Chapter 2 of this manual.

CMN_WAIT_TIME. Specifies the number of seconds that the subagent is to wait between polls of the AS/400 for communication line, controller, and device -related information.

Characteristics	**Allowed Values**
MIB Variable	cmnWaitTime
Valid Values	Integer greater than zero
Default Value	600
Set-requests	Yes

CONVERT_TO_ASCII. Determines if strings are converted between ASCII and EBCDIC when communicating with an SNMP manager. A value of 1 turns on the conversion process. A value of 0 indicates that the subagent should send strings in EBCDIC. This token can only be changed in the initial configuration file.

Characteristics	Allowed Values
MIB Variable	NONE
Valid Values	0,1
Default Value	1
Set-requests	No

CPU_RESEND_TRAP_COUNT. Determines how many times CPU utilization statistics are checked before repeating a cpuExcessive trap. The CPU_WAIT_TIME parameter specifies how often CPU utilization statistics are gathered.

Characteristics	Allowed Values
MIB Variable	cpuResendPeriod
Valid Values	Integer greater than zero
Default Value	5
Set-requests	Yes

CPU_TRAP_THRESHOLD. Determines the minimum percentage of CPU utilization at which cpuExcessive traps are generated. This value is specified in tenths of a percent therefore 95% would be specified as 950. To disable cpuExcessive traps, set this variable to a value greater than 1000.

Characteristics	Allowed Values
MIB Variable	cpuThreshold
Valid Values	1-1000
Default Value	950
Set-requests	Yes

CPU_WAIT_TIME. Specifies the number of seconds that the subagent is to wait between polls of the AS/400 for CPU related information. The total disk usage (or percent system Auxiliary Storage Pool or ASP) polling is also controlled by this parameter since the information is provided by the same OS/400 API as the CPU information.

Characteristics	Allowed Values
MIB Variable	cpuWaitTime
Valid Values	Integer greater than zero
Default Value	300
Set-requests	Yes

DELETE_ANSWERED_MSG. Determines the disposition of QSYSOPR messages that are answered by the subagent. Setting this token to 1 deletes QSYSOPR messages when they are answered. Setting this token to 0 causes the answered messages to remain in the message queue. This token can only be changed in the initial configuration file.

Characteristics	Allowed Values
MIB Variable	NONE
Valid Values	0,1
Default Value	0
Set-requests	No

DISK_FULL_CLR_THRESHOLD. Determines the amount of ASP utilization at which diskClear traps are generated**.** This value is specified in ten-thousandths of a percent, thus 95% would be specified as 950000.

Characteristics	Allowed Values
MIB Variable	diskFullClearThreshold
Valid Values	0-1000000
Default Value	850000
Set-requests	Yes

DISK_FULL_THRESHOLD. Determines the amount of ASP utilization at which diskFull traps are generated. This value is specified in ten-thousandths of a percent, thus 95% would be specified as 950000. To disable diskFull traps, set this variable to a value greater than 1000000.

Characteristics	Allowed Values
MIB Variable	diskFullThreshold
Valid Values	0-1000000
Default Value	900000
Set-requests	Yes

DISK_RESEND_TRAP_COUNT. Determines how many times disk utilization statistics are gathered before repeating a diskFull trap. CPU_WAIT_TIME specifies how frequently disk utilization statistics are gathered.

Characteristics	Allowed Values
MIB Variable	diskResendPeriod
Valid Values	Integer greater than zero
Default Value	5
Set-requests	Yes

DISK_WAIT_TIME. Specifies the number of seconds that the subagent is to wait between polls of the AS/400 for disk related information. This does not include total disk space utilization (system Auxiliary Storage Pool or ASP) as the AS/400 reports this statistic with the CPU information.

Characteristics	Allowed Values
MIB Variable	diskWaitTime
Valid Values	Integer greater than zero
Default Value	300
Set-requests	Yes

GET_TABLE_BY_ROW. Specifies the order by which tables are retrieved when walked by a series of SNMP getnext-requests. This parameter does not pertain to the COMTEKCMN and COMTEKGENQ SNMP subagents. A value of zero indicates that tables should be retrieved by lexicographical order (i.e., by column), a value of one indicates that tables should be retrieved by row. For example, if this value is set to zero, the following order results when walking a diskSystemTable with three instances:

Value Set to 0	Value Set to 1
diskSystemIndex.1	diskSystemIndex.1
diskSystemIndex.2	diskNumber.1
diskSystemIndex.3	diskSpace.1
diskNumber.1	diskSpaceUsed.1
diskNumber.2	diskPercentBusy.1
diskNumber.3	diskSystemIndex.2
diskSpace.1	diskNumber.2
diskSpace.2	diskSpace.2
diskSpace.3	diskSpaceUsed.2
diskSpaceUsed.1	diskPercentBusy.2
diskSpaceUsed.2	diskSystemIndex.3
diskSpaceUsed.3	diskNumber.3
diskPercentBusy.1	diskSpace.3
diskPercentBusy.2	diskSpaceUsed.3
diskPercentBusy.3	diskPercentyBusy.3

Characteristics	Allowed Values
MIB Variable	NONE
Valid Values	0, 1
Default Value	0
Set-requests	No

HUNDREDTHS_SEC_BETWEEN_TRAPS. Determines how long to wait between transmission of trap messages. This parameter allows the regulation of the amount of SNMP traffic that the subagent generates. See Chapter 4 of this manual for more information on trap throttling. This value is specified in hundredths of a second, so setting this token to 25 implies that a maximum of 4 traps per second are sent.

Characteristics	Allowed Values
MIB Variable	trapThrottle
Valid Values	Integer greater than zero
Default Value	50
Set-requests	Yes

INITIAL_REQ_PS_FILENAME. This is the fully qualified name of the file which contains the list of critical processes to verify on startup. Do not enclose the name in single quotes (‘) or double quotes (“). Do not leave spaces in the name. For example:

INITIAL_REQ_PS_FILENAME=COMTEK1/COMTEKCFG(CRIT_PROC1)

If this file name is invalid, critical process checking will be disabled until a valid file is provided.

Characteristics	Allowed Values
MIB Variable	psReqProcFileName
Valid Values	Any valid fully qualified member
Default Value	COMTEK1/COMTEKCFG(CRIT_PROC1)
Set-requests	Yes

JQ_WAIT_TIME. Specifies the number of seconds the subagent is to wait between polls of the AS/400 for job queue related information.

Characteristics	Allowed Values
MIB Variable	jobqWaitTime
Valid Values	Integer greater than zero
Default Value	300
Set-requests	Yes

MAX_RETAINED_TRAPS. Determines the total number of traps to be retained for later retransmission. This token can only be changed in the initial configuration file. See Chapter 4 of this manual for more information about retained traps and retransmission of traps.

Characteristics	Allowed Values
MIB Variable	NONE
Valid Values	1-200
Default Value	100
Set-requests	No

MIN_QSYSOPR_SEV_TO_SEND. Determines the minimum severity of QSYSOPR message that should be forwarded to the SNMP manager in the form of a qsysoprMsg trap.

Characteristics	Allowed Values
MIB Variable	qsysoprSeverity
Valid Values	0-99
Default Value	0
Set-requests	Yes

PS_RESEND_TRAP_COUNT. Determines how many times critical processes are checked before repeating a criticalProcessMissing trap. PS_WAIT_TIME specifies how frequently active job statistics are gathered.

Characteristics	Allowed Values
MIB Variable	psResendPeriod
Valid Values	Integer greater than zero
Default Value	0
Set-requests	Yes

PS_WAIT_TIME. Specifies the number of seconds the subagent is to wait between polls of the AS/400 for active process related information.

Characteristics	Allowed Values
MIB Variable	psWaitTime
Valid Values	Integer greater than zero
Default Value	300
Set-requests	Yes

QHST_WAIT_TIME. Specifies the number of seconds the subagent is to wait before rereading the QHST file.

Characteristics	Allowed Values
MIB Variable	qhistWaitTime
Valid Values	Integer greater than zero
Default Value	1000
Set-requests	Yes

QSYSOPR_ENFORCE_RETRY_LIMIT. Enables or disables limiting the number of times the COMTEKQSYS process should attempt to monitor the QSYSOPR message queue when another process has allocated the QSYSOPR message queue. If this is enabled (set to 1), the

COMTEKQSYS process will shut itself down upon reaching the maximum number of retries as specified by the QSYSOPR_RETRY_LIMIT configuration parameter, described below.

Characteristics	Allowed Values
MIB Variable	None
Valid Values	0, 1
Default Value	1
Set-requests	No

QSYSOPR_MSG_FILTER_FILE. Specifies the filename of the list that contains all of the QSYSOPR message IDs that should NOT be sent to the SNMP manager. Setting the token to the same value causes the current file to be read again.

Characteristics	Allowed Values
MIB Variable	qsysoprMsgFilterFile
Valid Values	Any fully qualified member name
Default Value	NONE
Set-requests	Yes

QSYSOPR_MSG_MAX_AGE_MINUTES. Configures a filter based on the age of QSYSOPR messages which prevents old QSYSOPR messages from being sent to the SNMP manager as qsysoprMsg traps. For example, a value of three indicates that messages more than three minutes old should not be sent to the SNMP manager. Caution should be used when setting this value to a large number since all QSYSOPR messages generated since that time will be transmitted as traps when the subagent starts up. When setting this value to a large value, HUNDRETHS_OF_SEC_BETWEEN_TRAPS should be tuned accordingly to prevent flooding the network with these traps. This value is specified in minutes.

Characteristics	Allowed Values
MIB Variable	QsysoprMaxMsgAge
Valid Values	Integer greater than zero
Default Value	3
Set-requests	Yes

QSYSOPR_RETRY_LIMIT. Specifies the number of times the COMTEKQSYS process should try to monitor the QSYSOPR message queue when another process has allocated the QSYSOPR message queue. If the maximum number of retries is reached and QSYSOPR_ENFORCE_RETRY_LIMIT is set to 1 (enabled), the COMTEKQSYS process will shut itself down.

Characteristics	Allowed Values
MIB Variable	None
Valid Values	Any positive integer
Default Value	375
Set-requests	No

RCONS_TRAPS_PER_SECOND. Determines how many traps per second can be generated while capturing the output of a remote console command.

Characteristics	Allowed Values
MIB Variable	NONE
Valid Values	Integer greater than zero
Default Value	2
Set-requests	No

Critical Process File

The critical process configuration file contains a list of the user names, job names, and number of instances of each that should be running in the system, and optionally the subsystem and a regular expression for expected active job status. The name of this file is specified either in the INIT_CFG file or can be set via the SNMP manager. By default, the initial critical process file is COMTEK1/COMTEKCFG(CRIT_PROC1). If an invalid file is specified, the critical process checking is disabled. An example entry is:

```
USER_1   JOB_NAME    3 SUBSYS  MSGW|DEQA
```

This example specifies that 3 instances of the job called JOB_NAME should be running under the user profile USER_1 in the subsystem called SUBSYS. Furthermore, the active job status should match the regular expression MSGW|DEQA. Separate the fields with one or more blanks. The subsystem and active job status fields are optional. Any entry that begins with # is considered a comment.

The following is an example of a critical process file:

```
# This is a comment at the beginning of the
# Critical Process Configuration file
USER_1   JOB_NAME      3
USER_2   COMTEKSNMP 1
QPGMR   PROG_JOB      1
# This is a comment at the end of the file
```

To add, delete or make changes to the list of critical processes, create another critical process configuration file and perform an SNMP set-request on the MIB variable psReqProcFileName to point to the new critical process configuration file. The change of the critical process list takes effect

immediately after the set-request command is processed. It is possible to keep multiple critical process configuration files on the AS/400 and select the appropriate configuration(s) as needed. Updates may also be made to the active critical process configuration file while the subagent is running. A set-request may then be performed without modifying the current value of the psReqProcFileName variable to cause the subagent to reread the active list of critical processes.

QSYSOPR Message ID Filter File

The QSYSOPR message ID filter configuration file contains a list of the message IDs that are either filtered out (not forwarded) or are forwarded to the SNMP manager as qsysoprMsg traps. The intention of the filter is determined by the inclusion of either the *FORWARD_SPEC_MSGS key word or the *FILTER_SPEC_MSGS key word. Only the first instance of either *FORWARD_SPEC_MSGS or *FILTER_SPEC_MSGS is accepted; all other instances are ignored. If the filter file does not contain either of these key words, then *FILTER_SPEC_MSGS is assumed. The *FORWARD_SPEC_MSGS indicates that only messages matching one of the entries in the filter file should be forwarded to the SNMP manager and all other messages should be discarded. The *FILTER_SPEC_MSGS indicates that messages matching one of the entries in the filter file should be filtered out and all other messages should be forwarded to the SNMP manager. The name of the file that contains QSYSOPR message IDs that are to be filtered is specified by the INIT_CFG file entry QSYSOPR_MSG_FILTER_FILE. The default setting in the INIT_CFG file for the QSYSOPR message ID filter file is COMTEK1/COMTEKCFG(SKIP_MSGID). If an invalid file is specified, QSYSOPR message ID filtering is disabled. The MIB variable qsysoprMsgFilterFile shows the current value of the filter filename and permits SNMP set-requests to modify the filter file which is being used while the subagent is running. To cause the subagent to reread the current filter file to pick up any modifications made to that file since the subagent was started, perform a set-request to the qsysoprMsgFilterFile without modifying the filename.

The following is an example of the QSYSOPR message ID filter configuration file that filters out three message Ids and forwards the rest:

```
#This is a comment in the QSYSOPR message ID
#configuration file
*FILTER_SPEC_MSGS
UCS1234
```

```
CPF5678
CPF1010
#This is a comment at the end of the file
```

Any entry that begins with a pound character (#) is considered a comment. In the above example, the all QSYSOPR message except those with the message ID UCS1234, CPF5678, or CPF1010 will be sent as qsysoprMsg traps to the SNMP manager.

The asterisk (*) can be used as a wildcard when specifying message IDS. When the asterisk is used, it cannot be the first character in the message ID.

```
#This is a comment in the QSYSOPR message ID
#configuration file
UCS12*
#This is a comment at the end of the file
```

In the above example filter configuration file, all QSYSOPR messages except those with the message ID UCS1200 through UCS12FF are forwarded to the SNMP manager as traps.

The following is an example of the QSYSOPR message ID filter configuration file that only forwards the messages whose Ids are specified in the filter file:

```
#This is a comment in the QSYSOPR message ID
#configuration file
*FORWARD_SPEC_MSGS
UCS*
CPF1*
#This is a comment at the end of the file
```

The effect of this filter is to send all QSYSOPR messages that have message Ids CPF1000 through CPF1FFF and UCS0000 through UCSFFFF to the SNMP manager.

Not all QSYSOPR messages contain a message ID. To filter out QSYSOPR messages which do not contain a message ID, use the special token *BLANK_ID.

```
#This is a comment in the QSYSOPR message ID
#configuration file
UCS1234
*BLANK_ID
#This is a comment at the end of the file
```

In the above example filter configuration file, all QSYSOPR messages except those with the message ID UCS1234 and those which have no message ID are forwarded to the SNMP manager as traps.

It is also possible to selectively forward or filter messages that do not have message IDs by using the special token *BLANK_REGEX, which specifies a regular expression to match. There can be up to 200 instances

of the special token *BLANK_REGEX in the file. The *FORWARD_REGEX_MESSAGES and *FILTER_REGEX_MESSAGES tokens control whether the messages which match the regular expression should be forwarded or filtered. If you specify the token *BLANK, then regular expression checking is disabled since *BLANK is the equivalent of specifying the regular expression ".*".

User Specified Message Queue Monitor Configuration

The user specified message queue monitor configuration is used to define which message queues are monitored and the initial settings for the message filtering facility. The file is

COMTEK1/COMTEKCFG(Q_MON_CFG)

Each line in the configuration file specifies a message queue, the library that the queue is contained in, the minimum severity message to forward, the maximum age of a message to be forwarded, and the message ID filter file to use. An example configuration file that monitors the QPGMR message queue might look like:

```
#DO NOT MAKE CHANGES TO THIS FILE WHILE COMTEKGENQ IS
#RUNNING
#The format of this file is:
#
#<MSGQ NAME>  <MSGQ LIB> <MIN SEV> <MAX AGE> <FILTER FILE>
#
QPGMR     QUSRSYS 0 15 COMTEK1/COMTEKCFG(SKIP_PGM)
```

The configuration file should not be updated while the COMTEKGENQ process is running since the queue monitor updates the configuration file every time an appropriate variable is set via SNMP. An changes that are made manually will be overwritten when an SNMP SET is processed.

Message Queue Name. Specifies the name of the message queue to be monitored. This parameter must be changed on the AS/400.

Characteristics	Allowed Values
MIB Variable	None
Valid Values	Any valid message queue name
Default Value	None
Set-requests	No

Message Queue Library. Specifies the name of the library which contains the message queue to be monitored. This parameter must be changed on the AS/400.

Characteristics	Allowed Values
MIB Variable	None
Valid Values	Any valid library name
Default Value	None
Set-requests	No

Minimum Severity. Determines the minimum severity of messages that should be forwarded to the SNMP manager in the form of a trap. There is an instance of this configuration parameter for each message queue that is monitored. This parameter is part of the os400MonQCfgTable table.

Characteristics	Allowed Values
MIB Variable	os400MsgQueueMinSeverity
Valid Values	0-99
Default Value	0
Set-requests	Yes

Maximum Age. Configures a filter based on the age of messages which prevents old messages from being sent to the SNMP manager as generic queue monitor traps. For example, a value of three indicates that messages more than three minutes old should not be sent to the SNMP manager.

Caution should be used when setting this value to a large number since all messages generated since that time will be transmitted as traps when the subagent starts up. When setting this value to a large value, HUNDRETHS_OF_SEC_BETWEEN_TRAPS should be tuned accordingly to prevent flooding the network with these traps. This value is specified in minutes.

Characteristics	Allowed Values
MIB Variable	os400MsgQueueMaxAge
Valid Values	Any positive integer
Default Value	None
Set-requests	Yes

Message ID Filter File. This configures a filter which uses message IDs to determine which messages should be forwarded to the SNMP manager in the form of a trap. There is an instance of this configuration parameter for each message queue that is monitored. This parameter is part of the os400MonQCfgTable table.

Characteristics	Allowed Values
MIB Variable	os400MsgQFilterFile
Valid Values	Any valid file member name
Default Value	None
Set-requests	Yes

User Specified Message Queue Message ID Filter Files

The user specified message queue message ID filter configuration file contains a list of the message IDs that are either filtered out (not forwarded) or are forwarded to the SNMP manager as qMonitor traps. The intention of the filter is determined by the inclusion of either the *FORWARD_SPEC_MSGS keyword or the *FILTER_SPEC_MSGS keyword. Only the first instance of either *FORWARD_SPEC_MSGS or *FILTER_SPEC_MSGS is accepted; all other instances are ignored. If the filter file does not contain either of these keywords, then *FILTER_SPEC_MSGS is assumed. The

*FORWARD_SPEC_MSGS indicates that only messages matching one of the entries in the filter file should be forwarded to the SNMP manager and all other messages should be discarded. The *FILTER_SPEC_MSGS indicates that messages matching one of the entries in the filter file should be filtered out and all other messages should be forwarded to the SNMP manager. Each message queue that is being monitored can use its own filter file or share a filter file. There is no default file name for this feature. The user should create as many files as required for their specific implementation. If an invalid file is specified, message ID filtering is disabled for the affected message queue. The column in the os400MonQCfgTable specified by the MIB variable os400MsgQFilterFile, shows the current value of the filter filename for each monitored message queue. The instances of the os400MsgQFilterFile variable permits SNMP set-requests to modify the filter file(s) which is being used while the subagent is running. To cause the subagent to reread the current filter file to pick up any modifications made to that file since the subagent was started, perform a set-request to the appropriate instance of os400MsgQFilterFile without modifying the filename.

The following is an example of the message ID filter configuration file:

```
#This is a comment in the a message ID
#configuration file
*FILTER_SPEC_MSGS
UCS1234
CPF5678
CPF1010
#This is a comment at the end of the file
```

Any entry that begins with a pound character (#) is considered a comment. In the above example, all messages except those with the message ID UCS1234, CPF5678, or CPF1010 will be sent as qMonitor traps to the SNMP manager.

The asterisk (*) can be used as a wildcard when specifying message IDS. When the asterisk is used, it cannot be the first character in the message ID.

```
#This is a comment in the message ID
#configuration file
UCS12*
#This is a comment at the end of the file
```

In the above example filter configuration file, all messages except those with the message ID UCS1200 through UCS12FF are forwarded to the SNMP manager as traps.

Not all messages contain a message ID. To filter out messages which do not contain a message ID, use the special token *BLANK_ID.

```
#This is a comment in the message ID
#configuration file
UCS1234
*BLANK_ID
#This is a comment at the end of the file
```

In the above example filter configuration file, all messages except those with the message ID UCS1234 and those which have no message ID are forwarded to the SNMP manager as traps.

The following is an example of a message ID filter configuration file that only forwards the messages whose IDs are specified in the filter file:

```
#This is a comment in the QSYSOPR message ID
#configuration file
*FORWARD_SPEC_MSGS
CPF1*
#This is a comment at the end of the file
```

The effect of this filter is to send all messages that have message IDs CPF1000 through CPF1FFF to the SNMP manager.

It is also possible to selectively forward or filter messages that do not have message IDs by using the special token *BLANK_REGEX, which specifies a regular expression to match. There can be up to 200 instances

of the special token *BLANK_REGEX in the file. The *FORWARD_REGEX_MESSAGES and *FILTER_REGEX_MESSAGES tokens control whether the messages which match the regular expression should be forwarded or filtered. If you specify the token *BLANK, then regular expression checking is disabled since *BLANK is the equivalent of specifying the regular expression ".*".

4

NMServer Traps

NMServer for OS/400 provides functionality that enables traps to be reliably received by the SNMP manager. Through the use of trap sequence numbering, the ability to resend traps, and a throttle for trap transmission, the SNMP manager can detect traps that are not received, resend recent traps, and prevent burst of traps from overloading the network.

Resending Traps

The NMServer subagent maintains an internal table of the last <n> traps that have been sent (where <n> is determined by the configuration file parameter MAX_RETAINED_TRAPS). Each trap that is generated by the subagent contains the trapNum trap sequence number MIB variable. The MIB variable firstTrapNum identifies the sequence number of the oldest trap available to be resent and the MIB variable lastTrapNum identifies the most recent sequence number assigned by the subagent.

To prevent traps from being lost, the SNMP manager should monitor the trapNum sequence number of traps that are received. If the trapNum is not one greater than the trapNum received with the previous trap, then at least one trap is missing. To resend a trap that was not received by the SNMP manager, the trapNum MIB variable should be set to the sequence number of the missing trap with an SNMP set-request. The trapNum value of the resent trap will contain its original sequence number.

In addition to monitoring the trapNum on received traps, the SNMP manager may choose to poll the subagent for the lastTrapNum as a heartbeat and additional lost trap detection mechanism.

Note that the size of the internal trap table (as governed by the MAX_RETAINED_TRAP configuration file parameter) may prevent the subagent from resending a lost trap. For example, if the MAX_RETAINED_TRAP parameter is set to 10, and the lastTrapNum variable indicates that the sequence number of the most recently sent trap is 11, only

traps 2-11 may be resent (and the firstTrapNum MIB variable will have a value of 2).

Trap Throttling

The NMServer for OS/400 subagent provides the capability to throttle traps to prevent them from flooding the network. This throttle takes the form of the HUNDREDTHS_SEC_BETWEEN_TRAPS configuration file parameter which is used by the subagent to determine the number of hundredths of a second to wait before sending the next trap.

Note: If the critical process monitor is being used, the missing process trap bypasses the throttle, and is sent immediately

Enterprise Specific Traps

The following table shows the traps that may be generated by the NMServer for OS/400 subagent. All traps generated by the subagent are enterprise specific and are sent with an enterprise of comtekos400Mib.

Trap Type	Trap Number	Cause
coldStart	0	The subagent was started.
diskFull	1	The system ASP exceeds the user specified threshold.
diskClear	2	The system ASP has returned to below the user specified threshold.
cpuExcessive	3	The CPU utilization has exceeded the user specified threshold.
critProcMissing	4	A process listed in the active critical process configuration file is not running.
qsysoprMsg	5	A QSYSOPR message was received.
userMsg	6	A user trap was received.
qsysoprHelpText	7	The QSYSOPR message help text was requested for the specified message key.
cpuClear	10	The CPU utilization has dropped below the excessive use threshold.
qMonitor	11	A message was received on a monitored message queue.

activeJobStatus	14	A critical process has an unexpected active job status

User Traps

User trap messages are designed to permit user defined printable data to be sent as a trap to the SNMP manager. User data up to 8000 bytes long can be formatted into userMsg trap messages. For user data longer than 255 bytes, the data is converted into a series of userMsg trap messages, each containing up to 255 characters. A continuation flag (userTrapContFlag) is used to indicate if the user data continues into the next userMsg trap message. See Chapter 5 of this manual for more information on generating user trap messages.

QSYSOPR Message Traps

Traps are generated for any QSYSOPR message that is at or above the minimum severity specified by the MIN_QSYSOPR_SEV_TO_SEND configuration file parameter. See Chapter 5 of this manual for more information on responding to QSYSOPR inquiry messages.

QSYSOPR Help Text Traps

If it is necessary to retrieve the help text for a QSYSOPR message, use and SNMP set-request to set the qsysoprMsgKey MIB variable to the qsysoprMsgKey message key value contained in the qsysoprMsg trap and the appropriate help text will be sent as a qsysoprHelpText trap.

User Specified Message Queue Traps

Traps are generated for any message that is received on any of the monitored message queues and meets the filtering parameters. See Chapter 5 of this manual for more information on responding to user specified message queue inquiry messages.

Dumping the Trap Table

It is possible to dump the trap table to a flat text file for the purpose of problem resolution. This is accomplished by issuing the DUMPTRAPS command. This will create (or replace) the file called COMTEK1/COMTEKCFG(DUMPTRAP). This file will contain a dump of the previously processed traps.

Sample Trap File

```
=====================================================
TRAP VERSION: 2
SENDING MODULE: COMTEK_SNMP
GENERIC TYPE :6
SPECIFIC TYPE: 0
SAVE IN TABLE: 1
ENTERPRISE: 1.3.6.1.4.1.597.3
ENTERPRISE OFFSET: 72
ENTERPRISE LENGTH: 18
NUMBER OF VARBINDS: 2
TOTAL LENGTH: 214
ALLOCATED SIZE: 3092
FIRST VARBIND OFFSET: 90
  VARBIND #0
   VARBIND LENGTH: 61
     DATA TYPE: 129
     SUBTREE LENGTH: 19
     SUBTREE OFFSET: 32
     OID LENGTH: 6
     OID OFFSET: 51
     DATA LENGTH: 4
     DATA OFFSET: 57
     SUBTREE: 1.3.6.1.4.1.597.3.
     OID: 4.3.0
     DATA: 0
   VARBIND #1
      VARBIND LENGTH: 63
      DATA TYPE: 2
      SUBTREE LENGTH: 19
      SUBTREE OFFSET: 32
      OID LENGTH: 6
      OID OFFSET: 51
      DATA LENGTH: 6
      DATA OFFSET: 57
      SUBTREE: 1.3.6.1.4.1.597.3.
      OID: 4.3.0
      DATA:
0000   C3 D6 D4 E3 C5 D2                        COMTEK

=====================================================
```

5

NMServer Component Operation

Remote Console

The remote console function allows the SNMP manager to send commands to a managed AS/400 system and optionally receive the output of those commands. This function is restricted to commands that allow their output to be sent to the print spooler and do not require an AS/400 terminal to display the output. If the user elects to receive the output of the command, the output is sent as a series of userMsg trap messages, with one trap message sent for each line in the resulting output file.

The remote console has a syntax that is similar to the standard OS/400 syntax. This syntax is:

CMD<user specified OS/400 command> OUT<specified output file> TAG<output id>

The only required field is CMD<user specified OS/400 command>. The other two fields are only meaningful if the output of the command is to be returned to the SNMP manager.

To receive the output from the command, include the statement OUTPUT(*PRINT) as part of the CMD field and specify the OUT field**.** The filename that should be in the OUT field is the file that the specified OS/400 command generates for print output. Since the file that is generated for output varies depending on the command entered, consult the OS/400 documentation for the name assigned to the output file for each command.

The optional TAG field adds a user-specified tag to the beginning of each line of output sent to the SNMP manager as user traps. This tag field is limited to 12 alphanumeric characters. The TAG can be useful in distinguishing the resulting traps from concurrent remote console commands.

The following is an example of a remote console command:

CMD<WRKSYSSTS OUTPUT (*PRINT)> OUT**<**QPDSPSTS**>** TAG<MY_TAG>

If the Remote Console command encounters an error, a UCS0050 error message is sent to the QSYSOPR message queue. The following table describes the cause of each error code.

Error Code	Cause
81	Nothing between <> after the CMD tag.
82	The specified command did not execute successfully. See the preceding UCS0051, UCS0052, or UCS0053 messages in the QSYSOPR message queue which include more information on the error.
83	Could not open the specified output file.
84	Could not copy the specified output (spool) file.
85	Could not delete the specified output (spool) file.
91	Could not find the tag : CMD.
92	Could not find < after CMD tag.
93	Could not find > after CMD tag.
94	Could not find < after OUT tag.
95	Could not find > after the OUT tag.
96	Could not find < after the TAG tag.
97	Could not find > after TAG tag.

Generating User Trap Messages

The subagent provides a data queue to send text as userMsg traps to the SNMP manager. The queue can accommodate messages up to 8000 bytes long. The queue is called SNMP_USR_Q and is located in the COMTEK1 library. The subagent disassembles the user's message into 255 byte null terminated character strings and sends each string along with a continuation flag (userTrapContFlag) to the SNMP manager. A continuation flag value of 1 indicates that the current user message is continued in the next userMsg trap. A continuation flag value of 0 indicates that this is the last string in the message. See Appendix A for an example program to place data into the user trap queue.

Responding to QSYSOPR Inquiry Messages

To respond to a QSYSOPR inquiry message, use an SNMP set-request to set the qsysoprResp MIB variable to the appropriate response text using the value contained in the qsysoprMsgKey variable of the trap as the instance of qsysoprResp in the set-request. For example, to respond to a qsysoprMsg

trap which has the qsysoprMsgKey value 1234, use an SNMP set-request for instance 1234 of qsysoprResp, i.e., set iso(1). org(3).dod(6).internet(1).private(4).enterprises(1).comtek(597).comtekos400Mib(3).qsysoprMsg(5).qsysoprResp(2).1234 to the appropriate response value.

If the SNMP set-request is successful, the QSYSOPR message response was correctly sent. If the configuration file parameter DELETE_ANSWERED_MSG is set to 0 (do not delete), there will be another QSYSOPR message trap with the same message key and a message type that indicates that the message was a reply message.

If the SNMP set-request is unsuccessful, the QSYSOPR message response failed and a QSYSOPR message trap with message ID UCS0016 is sent with the operating system exception code for the failure. For example, if the message key is not found in the QSYSOPR message queue, the set-request fails and a UCS0016 message is sent to the QSYSOPR message queue indicating a CPF2410 exception (key not found in queue).

Responding to User Specified Message Queue Inquiry Messages

To respond to a user specified message queue inquiry message, use an SNMP set-request to set the os400RspMsgResponse MIB variable to the appropriate response text using the name of the queue and value contained in the os400GenQTrapMsgKey variable of the trap as the instance of os400RspMsgResponse in the set-request. Keep in mind that when using a DisplayString as an index, the string must be turned into a series of integers which represent the length of the string followed by the ASCII codes for the characters in the string. For example, to respond to a qmonitor trap which is for the QPGMR (length is 5, QPGMR is 51 50 47 4D 52 in ASCII) message queue and has the os400GenQTrapMsgKey value 1234, use an SNMP set-request for instance 1234 of os400RspMsgResponse , i.e., set iso(1).org(3).dod(6).internet(1).private(4). enterprises(1).comtek(597).comtekos400(5). os400genericQmonitor(2).os400MsgRspTable(2). os400MsgRspTableEntry(1). os400RspMsgResponse(3).5.51.50.47.4D.52.1234 to the appropriate response value. The translation of ASCII to a series of integers, if done manually, can be error-prone; such translation should be done by software running on the Manager Station which can emit the proper Set Command to the Agent.

If the SNMP set-request is successful, the message response was correctly sent. There will be another user specified message queue message trap with the same message key and a message type that indicates that the message was a reply message.

If the SNMP set-request is unsuccessful, the message response failed and a QSYSOPR message trap with message ID UCS0016 is sent with the operating system exception code for the failure. For example, if the message key is not found in the specified message queue, the set-request fails and a UCS0016 message is sent to the QSYSOPR message queue indicating a CPF2410 exception (key not found in queue).

Issuing an SNMP Set-request

The following are general instructions on how to issue a SNMP set-request from an SNMP manager. Each manager has its own variations, but the general principles are the same. The SNMP set-request can be used either from the MIB browser or from the command line on the SNMP manager. The following sections describe how to use the SNMP set-request.

Setting a MIB Variable from the MIB Browser

This example shows how to set variables in the configuration group of the COMTEK OS/400 MIB. This method works for any MIB variable that is writeable, i.e., has an ACCESS clause of read-write. The qsysoprMsg, trapinfo, and remoteConsole groups of the COMTEK OS/400 MIB also contain writeable MIB variables.

In this case, open the MIB browser for the particular AS/400 that will issue the set-request. View the comtekos400Mib subtree. It should display the following groups:

- resources
- processes
- jobq
- trapinfo
- qsysoprMsg
- configuration
- remoteConsole

Select the configuration group and press the <<Down Tree>> button on the browser. This displays the configuration parameters that can be changed without stopping the subagent. Select the parameter to be changed and then press the button on the MIB browser to retrieve the value of the variable. On the browser, this button may be labeled <<Start Query>>. This retrieves the current value of the variable. Enter the new value over the displayed value and press the <<set>> button.

Setting a MIB Variable from the Command Line

This is used most often when a program is written to automatically respond to conditions on a managed system. The best way to learn how to use this command is to refer to the manual for the SNMP manager. Most UNIX based systems have online documentation available through the man utility. To use the man utility, enter the command "man snmpset".

The syntax of the snmpset command is:

snmpset <ip address> <object identifier> <data type> <new value>

The above command has the following values:

- ip-address is the TCP/IP address of the managed system
- object identifier refers to the ASN.1 encoded identifier for the SNMP variable. The best way to determine this is to look at the dotted notation for the variable that is displayed in the MIB browser when pressing the <<describe>> button.
- data-type refers to whether the variable is integer, displayString, octetString, or any other data type
- new-value refers to the new value for the variable

6

NMServer MIB Subtrees

comtekos400Mib Subtree

OS400 Group

System Name
System Model
Number of Entries in the History Table
History Table
    History Table Row Number
    History Message
UTC/GMT Time Offset

CPU Group

Percent of Process Unit Used
Number of Jobs in the System
Percent of Permanent Addresses Used
Percent of Temporary Addresses Used
System Auxiliary Storage Capacity
Percent of System Auxiliary Storage in Use
Total Auxiliary Storage
Storage for Temporary Objects
Maximum Storage for Temporary Objects
Number of Pools
Pool Table
    Pool Index
    Maximum Number of Active Jobs
    Active to Wait
    Wait to Ineligible
    Active to Ineligible
    Pool Name
    Subsystem Name

Subsystem Library Name

Disk Group

Number of Entries in the Disk System Table
Disk System Table
- Disk System Index
- Disk Number
- Disk Space
- Disk Space Used
- Disk Percent Busy
- Disk Percent Used
- Disk I/O Requests
- Disk Request Size
- Disk Read Requests
- Disk Write Requests
- Disk Read Kbytes
- Disk Write Kbytes

User Statistics Group

Number of Users Signed On
Number of Users Disconnected
Number of Users Suspended by System Request
Number of Users Suspended by Group Jobs
Number of Users Signed Off with Print Jobs Waiting

Batch Job Statistics Group

Number of Batch Jobs Waiting for Messages
Number of Batch Jobs Running
Number of Batch Jobs Held Running
Number of Batch Jobs Ending
Number of Batch Jobs Scheduled
Number of Batch Jobs Held on Job Queue
Number of Batch Jobs on Held Job Queue
Number of Batch Jobs on an Unassigned Job Queue
Number of Batch Jobs Ending with Printer Output Waiting

Process Statistics Group

Number of Processes
Process Statistics Table
- Process Index
- Subsystem Name

- Job Name
- User Name
- CPU Utilization
- Function Initiated by the Job
- Status of the Job
- Active Job Status

Critical Process Configuration Table Size
Critical Process Configuration Table
- Critical Process Configuration Table Row Number
- Name of Critical Process
- User Profile Associated with Critical Process
- Subsystem where Critical Process should Run
- Number of Instances of the Critical Process Expected
- Number of Instances of the Critical Process Running

Actual Number of Critical Process Instances Running
Critical Process WRKACTJOB Statistics Table
- Critical Process Table Row Number
- Subsystem
- Job Name
- User Name
- Cpu Utilization
- Last High-Level Function
- Job Status
- Active Job Status

Job Queue Statistics Group

Number of Jobs
Job Queue Statistics Table
- Process Index
- User Name
- Job Name
- Job Type
- Job Status

Trap Message Group

First Trap Number
Last Trap Number
Resent Trap Number

Trap Continuation Flag
User Trap Message

QSYSOPR Message Group

Message Key
Response
Message Severity
Message ID
Message Type
Number of Help Message Lines
QSYSOPR Message Text Table
    Message Text
Short Message Text
Name of Job
Name of User
Number of Job
Name of Program

Configurable Parameters Group

CPU Wait Time
CPU Threshold
CPU Trap Resend Period
Disk Wait Time
Disk Full Threshold
Disk Full Clear threshold
Disk Trap Resend Period
Job Wait Time
Process Statistics Wait Time
Critical Process Configuration File Name
QSYSOPR Minimum Severity
Wait Time for QHST File
Trap Throttle
QSYSOPR Message Filter File Name
QSYSOPR Maximum Age of Message
Critical Process Trap Resend Period

Remote Console Group

Console Command

comtekos400 Subtree - os400cmn Group

Device Monitor Configuration Group

Software Version
Wait Time for checking

Communications Line Group

Line Count
Communications Line Table
- Communications Line Name
- Communications Line Status - Text
- Communications Line Status - Numeric
- Communications Line Category
- Communications Line Text Description
- Communications Line Index

Controller Group

Controller Count
Controller Table
- Controller Status - Text
- Controller Status - Numeric
- Controller Category
- Controller Text Description
- Controller Index
- Controller Name

Device Group

Device Count
Device Table
- Device Name
- Device Status - Text
- Device Status - Numeric
- Device Category
- Device Text Description
- Device Index

comtekos400 Subtree – os400genericQmonitor Group

Queue Monitor Configuration Group

Software Version

Number of actively monitored message queues
Configuration Table
- Monitored Message Queue Index
- Monitored Message Queue Name
- Monitored Message Queue Library
- Minimum Message Severity
- Maximum Message Age
- Message Queue Filter File Name

Message Response Group

Message Response Table
- Message Queue to Send Response to
- Message Key to Respond to
- Message Response

Trap Variable Group

Message Queue Name
Message Key
Message Severity
Message ID
Message Type
Message Text
Sending Job
User Name
Job Number
Program Name

Appendix A

User Trap Queue Example

The following source code sends user traps to the user trap queue.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <recio.h>
#include <signal.h>

/*****************************************/
/**                main                 **/
/**                                     **/
/*****************************************/

int main(void)
{
char   userInqName[11] =  “SNMP_USR_Q”;
char   userInqLib[11]  =  “COMTEK1   “;
char   msgText[8000];

strcpy(msgText,"This is a test message for the user trap manager.");

QSNDDTAQ(userInqName,userInqLib,
        (decimal(5,0)) strlen(msgText),
         msgText);

/*  if (errorOccurred)                                          */
/*     {                                                        */
/*PLACE APPROPRIATE ERROR HANDLING CODE  */
/*HERE                                                        */
/*     }      /* endif errorOccurred                          */

exit (0);
}
```

Appendix B

COMTEK Message File

The COMTEK NMServer subagent message file is called COMTEKMSG. This file is shipped in the COMTEK1 library. The COMTEK1 library must be in the library list or the message file must be copied to another library that is in the library list in order for messages to be correctly displayed.

The following messages have been defined:

Message	Message text	Description
UCS0003	&1: DPI unregister received from agent, reason = &2, subtree = &3	An SNMP DPI unregister command was received.
UCS0004	&1: Exiting due to error. See previous messages for explanation.	An unrecoverable error has occurred.
UCS0005	&1: Shutdown received. Exiting immediately.	A shutdown command was received.
UCS0006	&1: Recovering master agent connection. In procedure &2 at line &3.	The subagent received a shutdown command (most probably because of a time-out) and is attempting to reestablish the connection to the IBM master agent.
UCS0007	&1: Subagent registration failed - exiting, Error in procedure &2 at line &3.	The subagent could not register with the IBM master agent.
UCS0008	&1: Recovery of agent connection failed in procedure &2 at line &3.	The recovery of the subagent to master agent connection failed. The subagent will shutdown.
UCS0009	&1: Received close from agent, reason = &2.	The subagent received an SNMP DPI close command from the IBM master agent.

UCS0010	&1: Attempt to call QMHRCVM failed with exception &2.	An error occurred while reading the QSYSOPR. message queue (QMHRCVM API).
UCS0011	&1: Attempt to call QMHRCVM with exception &2. Help message could not be retrieved for message key &3.	An error occurred while reading the QSYSOPR message queue (QMHRCVM API).
UCS0012	&1: Attempt to call QUSLJOB failed with exception &2.	An error occurred while listing active or submitted jobs (QUSLJOB API).
UCS0013	&1: Get pointer to user space for &2 with exception &3.	An error occurred while getting a pointer to the user space that contains the list of active or submitted jobs (QUSPTRUS API).
UCS0014	&1: Attempt to create user space &2 failed with exception &3.	An error occurred while creating a user space to contain the list of active or submitted jobs (QUSCRTUS API).
UCS0015	&1: QWCRSSTS failed with exception &2.	An error occurred while getting WRKSYSSTS information (QWCRSSTS API).
UCS0016	&1: QMHSNDRM received exception &2 while responding to message key &3.	An error occurred while responding to a QSYSOPR message (QMHSNDRM API).
UCS0017	&1: Attempt to call QCRSVAL failed with exception &2 in procedure &3 at line &4.	Could not retrieve a system value from the operating system. Check the AS400 Codes and Messages documentation to determine why this error occurred.

UCS0020	&1: QSNDDTAQ received exception &2 for &3 in &4.	An error occurred while sending data on a data queue (QSNDDTAQ API).
UCS0021	&1: QRCVDTAQ received exception &2 for &3 in &4.	An error occurred while receiving data on a data queue (QRCVDTAQ API).
UCS0030	&1: Request to set unknown parameter &2 received.	A data collection module was asked to set a parameter that it does not handle.
UCS0031	&1: Received unknown message id &2.	A module was sent a command that it does not handle.
UCS0032	&1: Error in allocating memory for variable &2.	An error occurred while allocating memory for a variable.
UCS0033	&1: Received unknown trap type &2.	An invalid trap type was requested.
UCS0034	&1: Error reading file &2.	An error occurred while attempting to read the specified file.
UCS0035	&1: Bad entry in message ID filter list: &2.	An invalid message ID was found in the QSYSOPR message ID filter file. Correct the entry and try again.
UCS0036	&1: A message from an older NMServer/400 component was received. Please make sure all components are updated.	The NMServer/400 installation has incompatible versions of its components. Save your configuration files (COMTEK1/COMTEKCFG), clear the COMTEK1 library, and reinstall from the latest distribution media.

UCS0037	&1: Received a trap request that does not match the length specified in the header.	A malformed trap request was received.
UCS0040	&1: mkDPIset returned NULL for object identifier &2.	The mkDPIset API returned NULL.
UCS0041	&1: The DPI packet was not a response. Received DPI packet type: &2.	The subagent expected an SNMP DPI response packet but received something else.
UCS0042	&1: Error return code &2 on sendDPI packet.	An error occurred while calling the SNMP DPI sendDPIpacket API.
UCS0043	&1: Error &2 on waitDPIpacket.	An error occurred while calling the SNMP DPI waitDPIpacket API.
UCS0044	&1: Error return code &2 on disconnectSNMP.	An error occurred while calling the SNMP DPI disconnectSNMP API
UCS0045	&1: pDPIpacket returned NULL in procedure &2 at line &3.	An error occurred while calling the SNMP DPI pDPIpacket API.
UCS0046	&1: Error return code &2 in response packet in procedure &3 at line &4.	An internal SNMP DPI error occurred.
UCS0047	&1: connectSNMP returned rc = &2 in procedure &3 at line &4.	An error occurred while calling the SNMP DPI connectDPI API.
UCS0048	&1: Received unknown DPI packet type &2 in procedure &3 at line &4.	A packet with an unknown DPI packet type identifier was received.
UCS0049	&1: Exiting because received rc = &2 on packet type &3 in procedure &4 at line &5.	The specified return code was received for the specified DPI packet type.
UCS0050	&1: Error issuing remote console command. RC = &2.	A remote console error occurred.

UCS0051	&1: Error executing command &2. RC = &3.	An error occurred while attempting to parse or execute a remote console command.
UCS0052	&1: Received error &2 while copying &3 to &4.	An error occurred while attempting to copy the captured output from a remote console command.
UCS0053	&1: Received error &2 while deleting file &3.	An error occurred while attempting to delete the file that contains the captured output from a remote console command.
UCS0060	&1: mkDPIopen returned NULL in procedure &2 at line &3.	The SNMP DPI API mkDPIopen returned a NULL pointer.
UCS0061	&1: mkDPIregister returned NULL in procedure &2 at line &3.	The SNMP DPI API mkDPIregister returned a NULL pointer.
UCS0062	&1: mkDPIunregister returned NULL in procedure &2 at line &3.	The SNMP DPI API mkDPIunregister returned a NULL pointer.
UCS0063	&1: mkDPIclose returned NULL in procedure &2 at line &3.	The SNMP DPI API mkDPIclose returned a NULL pointer.
UCS0064	&1: mkDPItrap returned NULL in procedure &2 at line &3.	The SNMP DPI API mkDPItrap returned a NULL pointer.
UCS0065	&1: mkDPIresponse returned NULL in procedure &2 at line &3.	The SNMP DPI API mkDPIresponse returned a NULL pointer.
UCS0070	&1: The license key you have entered is invalid. Please try again. If you still have difficulty, please contact COMTEK	The INIT_CFG file contains an invalid license key.

UCS0071	&1: The trial period for the COMTEK NMServer product will expire &3/&4/&2. After that date, the product will not run.	NMServer/400 is running with a test license key that will expire on the given date.
UCS0072	&1: The trial period for COMTEK NMServer has expired. If you wish to purchase the product or extend the trial, contact COMTEK	The test license key for NMServer/400 has expired.
UCS0080	&1: The hostname and domain have not been configured for this system. These are required for the proper operation of SNMP.	The OS/400 TCP/IP configuration is missing either the host name or domain name.
UCS0081	&1: The hostname of this system was not found in the hosts table. This is necessary for the proper operation of SNMP.	The OS/400 TCP/IP configuration does not contain the expected host name for this AS/400 in its hosts file.

Appendix C

Using SNMP DPI Debugging

The COMTEK NMServer subagents provide DPI level 2 tracing when the question (Start SNMP DPI debugging?) on the COMTEK1/STRCOMTEK form is set to YES. The trace is taken at the IBM master agent level . It contains a record of all transactions that occur between the master agent and the subagents. The subagents that interact with the master agent are: COMTEKSNMP, COMTEKCMN, and COMTEKGENQ. The other NMServer/400 programs on the start menu do not directly interact with the IBM master SNMP agent.

To start debugging, type COMTEK1/STRCOMTEK on the command line, press PF4, change the default option for DPI debugging, and press enter. If you want to only run an SNMP DPI trace for one of the 3 subagents, only start the subagent you want to debug when you specify SNMP DPI Debugging. For example, the prompt screen may look like:

```
                Start Comtek SNMP Sub-Agent (STRCOMTEK)

Type choices, press Enter.

Start kernel (COMTEKSNMP)?  *YES    Character value, *YES, *NO
Start COMTEKCPU? . . . . .  *YES    Character value, *YES, *NO
Start COMTEKDSK? . . . . .  *YES    Character value, *YES, *NO
Start COMTEKPSA? . . . . .  *YES    Character value, *YES, *NO
Start COMTEKPSB? . . . . .  *YES    Character value, *YES, *NO
Start COMTEKQSYS?  . . . .  *YES    Character value, *YES, *NO
Start COMTEKQHST?  . . . .  *NO     Character value, *YES, *NO
Start COMTEKUTRP?  . . . .  *YES    Character value, *YES, *NO
Start COMTEKRCONS? . . . .  *YES    Character value, *YES, *NO
End COMTEKCMN  . . . . . .  *YES    Character value, *YES, *NO
End COMTEKGENQ?  . . . . .  *YES    Character value, *YES, *NO
Start SNMP DPI debugging?   *YES    Character value, *YES, *NO
```

Each NMServer sends its DPI trace to a print file in the print queue. This file will not be closed until entering the COMTEK1/ENDCOMTEK command. At this point, display or print the trace. Use option 5 from the WRKSPLF command to display the trace on the screen, otherwise, allow the trace to print.

Appendix D

MIB File

NMSERVERAS400.MIB

```
-- file: nmserveras400.mib
--
-- This file contains all of the NMServer for AS/400 MIBs.  These MIBs
-- were previously contained in the following files: comtek.mib,
cmtka4v2.mib,
-- comtektp.mib, cmtkcmn.mib and cmtkgenq.mib.  These MIBs were collapsed
into
-- this single file in order to simplify loading the MIBs, no functional
changes
-- have been made to the MIBs.
--
-- formerly file: comtek.mib
--
-- COMTEK Services, Inc.
-- Date         March 2006
-- Author       JS
--
-- Copyright 1994-2006 COMTEK Services, Inc.  All Rights Reserved.
--
-- This COMTEK Services SNMP Management Information Base Specification
-- (Specification) embodies COMTEK Services' confidential and
-- proprietary intellectual property.  COMTEK Services retains all
-- title and ownership in the Specification, including any
-- revisions.
--
-- This Specification is supplied "AS IS," and COMTEK Services makes
-- no warranty, either express or implied, as to the use,
-- operation, condition, or performance of the Specification.
--

COMTEKA4-MIB DEFINITIONS ::= BEGIN

IMPORTS
     enterprises    FROM RFC1155-SMI
     DisplayString  FROM RFC1213-MIB
     OBJECT-TYPE    FROM RFC-1212
     TRAP-TYPE      FROM RFC-1215;

comtek OBJECT IDENTIFIER ::= { enterprises 597 }

comtekvosMib OBJECT IDENTIFIER                  ::= { comtek 1 }
comtekvosAgent OBJECT IDENTIFIER                ::= { comtek 2 }
comtekos400Mib OBJECT IDENTIFIER                ::= { comtek 3 }
comtekVms OBJECT IDENTIFIER                     ::= { comtek 4 }
comtekos400 OBJECT IDENTIFIER                   ::= { comtek 5 }
```

```
-- SUBAGENT DEFINITIONS
comtekSubagent OBJECT IDENTIFIER                ::= { comtek 100 }
sampleMib OBJECT IDENTIFIER                     ::= { comtek 101 }
sampleSubagent OBJECT IDENTIFIER                ::= { comtek 102 }

-- VOS Subagents:

-- OS/400 Subagents:
os400cmn OBJECT IDENTIFIER                      ::= { comtekos400 1 }
os400genericQmonitor OBJECT IDENTIFIER           ::= { comtekos400 2 }

-- OpenVMS Subagents:
-- Note:  The following pairs of object identifiers must match the values
-- in by the subagent code. These numbers uniquely identify the MIB and
-- subagent.  Object identifiers for new subagents and their corresponding
-- MIBs should be added to the end of this list.
comtekVmsNMMasterMib OBJECT IDENTIFIER          ::= { comtekVms 1 }
comtekVmsNMMasterAgent OBJECT IDENTIFIER        ::= { comtekVms 2 }
comtekVmsNMSysMgrMib OBJECT IDENTIFIER          ::= { comtekVms 3 }
comtekVmsNMSysMgrSubagent OBJECT IDENTIFIER     ::= { comtekVms 4 }
comtekVmsNMTrpMgrMib OBJECT IDENTIFIER          ::= { comtekVms 5 }
comtekVmsNMTrpMgrSubagent OBJECT IDENTIFIER     ::= { comtekVms 6 }
comtekVmsNMConsoleMib OBJECT IDENTIFIER         ::= { comtekVms 7 }
comtekVmsNMConsoleSubagent OBJECT IDENTIFIER    ::= { comtekVms 8 }
comtekVmsNMOpcomMib OBJECT IDENTIFIER           ::= { comtekVms 13 }
comtekVmsNMOpcomSubagent OBJECT IDENTIFIER      ::= { comtekVms 14 }
comtekVmsNMVmsMonMib OBJECT IDENTIFIER          ::= { comtekVms 15 }
comtekVmsNMVmsMonSubagent OBJECT IDENTIFIER     ::= { comtekVms 16 }


-- formerly file: cmtka4v2.mib

resources  OBJECT IDENTIFIER  ::= { comtekos400Mib 1 }

os400  OBJECT IDENTIFIER  ::= { resources 1 }

os400SysName OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "OS/400 system name.  "
     ::= { os400 1 }

os400SysModel OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "OS/400 system model.  "
     ::= { os400 2 }

os400NumQHistoryEntries OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of entries in the subagent's QHST table
          (os400QHistoryTable).  The QHST table has a maximum
```

```
          of 100 entries.  "
     ::= { os400 3 }

os400QHistoryTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF Os400QHistoryTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "A simplified version of what is read from the
          QHST file.  The MIB variable qhistWaitTime specifies
          how often the QHST file is reread for new entries.
          This table contains the most recent 100 entries read
          from the QHST file.  "
     ::= { os400 4 }

os400QHistoryTableEntry OBJECT-TYPE
     SYNTAX  Os400QHistoryTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "A row in the QHST table.  "
     INDEX {os400QHistoryRowNumber}
     ::= { os400QHistoryTable 1 }

Os400QHistoryTableEntry ::= SEQUENCE {
   os400QHistoryRowNumber       INTEGER,
   os400HstMessage      DisplayString
          }

os400QHistoryRowNumber OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The row number of the QHST table. This value is used as
          the index into the os400QHistoryTable.  "
     ::= { os400QHistoryTableEntry 1 }

os400HstMessage OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The text of the QHST entry.  "
     ::= { os400QHistoryTableEntry 2 }

os400UtcOffset OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The UTC (or GMT) offset for the location of this AS/400.
                  The format is + or - followed by HHMM where HH denotes
                  hours and MM denotes minutes."
     ::= { os400 5 }

cpu  OBJECT IDENTIFIER  ::= { resources 2 }

cpuPercentProcessUnitUsed OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
```

```
     DESCRIPTION "The average of the elapsed time during which the processing
          units were in use.  This value is expressed in tenths of a
          percent.  For example, a value of 411 would indicate 41.1%.

          When this value exceeds the threshold specified by the MIB
          variable cpuThreshold, a cpuExcessive trap is generated.
          While CPU utilization continues to exceed the cpuThreshold
          value, the cpuExcessive trap is repeated every <n> times the
          CPU utilization statistics are gathered.  <n> is specified
          by the cpuResendPeriod MIB variable.  "
     ::= { cpu 1 }

cpuJobsInSys OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The total number of user jobs and system jobs that are
          currently in the system.  The total includes: all jobs on
          queues waiting to be processed, all jobs currently active
          (being processed), and all jobs that have completed running
          but still have output on output queues to be produced.  "
     ::= { cpu 2 }

cpuPercentPermAddresses OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Percentage of the maximum possible addresses for permanent
          objects that have been used.  This value is expressed in
          thousandths of a percent.  For example, a value of 44123 would
          indicate 44.123%  "
     ::= { cpu 3 }

cpuPercentTempAddresses OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Percentage of the maximum possible addresses for temporary
          objects that have been used.  This value is expressed in
          thousandths of a percent.  For example, a value of 44123 would
          indicate 44.123%  "
     ::= { cpu 4 }

cpuSystemASP OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The storage capacity of the system auxiliary storage pool
          (ASP). When OS/400 checksum protection is in effect, this is
          the amount of space available for the storage of protected
          data only.  Otherwise this represents the amount of space
          available for storage of both protected and unprotected data.
          This value is in millions of bytes.  "
     ::= { cpu 5 }

cpuPercentSystemASP OBJECT-TYPE
     SYNTAX  INTEGER
```

```
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Percentage of system storage pool currently in use. When
          OS/400 checksum protection is in effect, this percentage
          refers only to protected storage currently in use.
          Otherwise, it is the percentage of the total system storage
          pool currently in use.  This value is expressed in ten-
          thousandths of a percent.  For example, 44123 would indicate
          4.4123%.

          When this value exceeds the threshold specified by the MIB
          variable diskFullThreshold, a diskFull trap is generated.
          While disk utilization continues to exceed the diskFullThreshold
          value, the diskFull trap is repeated every <n> times the
          disk utilization statistics are gathered.  <n> is specified by
          the diskResendPeriod MIB variable.  "
     ::= { cpu 6 }

cpuTotalAuxStorage OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
  DESCRIPTION "Total auxiliary storage (in millions of bytes) on the system.
"
     ::= { cpu 7 }

cpuTotalUnprotStorageUsed OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The current amount of storage in use for temporary objects
          and machine data that are stored in unprotected storage
          when OS/400 checksum protection is in effect.  This is used
          in conjunction with maximum unprotected storage to determine
          how much unprotected storage should be reserved when checksum
          protection is started.  This value is in millions of bytes.  "
     ::= { cpu 8 }

cpuMaximumUnprotStorageUsed OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The largest amount of storage (for temporary objects and
          machine data that are stored in unprotected storage when
          checksum protection is in effect) used at any one time since
          the last IPL.  This value is in millions of bytes.  "
     ::= { cpu 9 }

cpuNumberPoolTableEntries OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of system pools allocated when the information
          was gathered. There is one pool table entry for each pool.  "
     ::= { cpu 10 }

cpuPoolTable OBJECT-TYPE
```

```
     SYNTAX  SEQUENCE OF CpuPoolTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "There is one pool table for each system pool.  "
     ::= { cpu 11 }

cpuPoolTableEntry OBJECT-TYPE
     SYNTAX  CpuPoolTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "This is a table of CPU pool information.  "
     INDEX {cpuPoolindex}
     ::= { cpuPoolTable 1 }

CpuPoolTableEntry ::= SEQUENCE {
   cpuPoolindex         INTEGER,
   maxActiveJobs        INTEGER,
   activeToWait         INTEGER,
   waitToIneligible     INTEGER,
   activeToIneligible   INTEGER,
   poolName     DisplayString,
   subsystemName        DisplayString,
   subsystemLibraryName         DisplayString
         }

cpuPoolindex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Location in CPU pool table.  "
     ::= { cpuPoolTableEntry 1 }

maxActiveJobs OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Maximum number of jobs that can be active in the pool
          at any one time.  "
     ::= { cpuPoolTableEntry 2 }

activeToWait OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The rate (in tenths), in transactions per minute, of
          transitions of jobs from an active condition to a wait
          condition.  For example, a value of 123 would indicate
          a rate of 12.3 transactions per minute.  "
     ::= { cpuPoolTableEntry 3 }

waitToIneligible OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The rate (in tenths), in transactions per minute, of
          transitions of jobs from a waiting condition to an
          ineligible condition.  For example, a value of 123 would
```

```
          indicate a rate of 12.3 transactions per minute.  "
     ::= { cpuPoolTableEntry 4 }

activeToIneligible OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The rate (in tenths), in transactions per minute, of
          transitions of jobs from an active condition to an
          ineligible condition.  For example, a value of 123 would
          indicate a rate of 12.3 transactions per minute.  "
     ::= { cpuPoolTableEntry 5 }

poolName OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(10))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The name of this storage pool.  The name may be a number,
          in which case it is a private pool associated with a subsystem.
          The following special values may be returned: *MACHINE, *BASE,
          *INTERACT *SPOOL, or *SHRPOOL1-*SHRPOOL10.  "
     ::= { cpuPoolTableEntry 6 }

subsystemName OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(10))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The subsystem with which this storage pool is associated.
          This field will be blank for shared pools.  "
     ::= { cpuPoolTableEntry 7 }

subsystemLibraryName OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(10))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The library containing the subsystem description.  This
          field will be blank for shared pools.  "
     ::= { cpuPoolTableEntry 8 }

disk  OBJECT IDENTIFIER  ::= { resources 3 }

diskSystemNumberEntries OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Number of entries in disk system table (diskSystemTable).
"
     ::= { disk 1 }

diskSystemTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DiskSystemEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "Table of disk system statistics.  "
     ::= { disk 2 }

diskSystemEntry OBJECT-TYPE
```

```
     SYNTAX  DiskSystemEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "Disk system table structure.  "
     INDEX {diskSystemIndex}
     ::= { diskSystemTable 1 }

DiskSystemEntry ::= SEQUENCE {
   diskSystemIndex      INTEGER,
   diskNumber   INTEGER,
   diskSpace    INTEGER,
   diskSpaceUsed        INTEGER,
   diskPercentBusy      INTEGER,
   diskPercentUsed      INTEGER,
   diskIoRequests       INTEGER,
   diskRequestSize      INTEGER,
   diskReadRequests     INTEGER,
   diskWriteRequests    INTEGER,
   diskReadK    INTEGER,
   diskWriteK   INTEGER
          }

diskSystemIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Location in table.  "
     ::= { diskSystemEntry 1 }

diskNumber OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The disk unit identifier is the same number used by the
          display disk configuration function of the OS/400
          system service tools.  If the disk unit identifier is 0,
          the disk unit is not configured.  "
     ::= { diskSystemEntry 2 }

diskSpace OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The total amount of storage that the unit can contain in
          millions of bytes.  When the checksum protection provided
          by OS/400 system software is on, this is the protected
          storage.  "
     ::= { diskSystemEntry 3 }

diskSpaceUsed OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The percentage of the disk that is currently allocated.
          When the checksum protection provided by OS/400 system
          software is on, this is the percentage of protected storage
          that is currently allocated.  This value is expressed in
```

```
          tenths of a percent.  For example, the value 455 would
          indicate 45.5%.  "
     ::= { diskSystemEntry 4 }

diskPercentBusy OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The estimated percentage of time that the disk unit is
          being used during the elapsed time.  This estimate is based
          on the number of I/O requests, the amount of data transferred,
          and the performance characteristics of the type of disk unit.
          This value is expressed in tenths of a percent.  For example,
          the value 455 would indicate 45.5%.  "
     ::= { diskSystemEntry 5 }

diskPercentUsed OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The estimated percentage (in tenths) of disk space used.  "
     ::= { diskSystemEntry 6 }

diskIoRequests OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of I/O requests  on this disk.  "
     ::= { diskSystemEntry 7 }

diskRequestSize OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The disk request size.  "
     ::= { diskSystemEntry 8 }

diskReadRequests OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of disk read requests.  "
     ::= { diskSystemEntry 9 }

diskWriteRequests OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of disk write requests.  "
     ::= { diskSystemEntry 10 }

diskReadK OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The amount of data read in Kilobytes (KB).  "
     ::= { diskSystemEntry 11 }
```

```
diskWriteK OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The amount of data written in Kilobytes (KB).  "
     ::= { diskSystemEntry 12 }

userStatistics  OBJECT IDENTIFIER  ::= { resources 4 }

usersSignedOn OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of users currently signed on.  "
     ::= { userStatistics 1 }

usersDisconnected OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of users temporarily signed off (disconnected).
"
     ::= { userStatistics 2 }

usersSuspBySysReq OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of users suspended by system request.  "
     ::= { userStatistics 3 }

usersSuspByGrpRJobs OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of users suspended by group jobs.  "
     ::= { userStatistics 4 }

usersSignedOffPrintWait OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
    DESCRIPTION "The number of users signed off with printer output waiting.
"
     ::= { userStatistics 5 }

batchJobStatistics  OBJECT IDENTIFIER  ::= { resources 5 }

batchJobsWaitMsg OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of batch jobs waiting for messages.  "
     ::= { batchJobStatistics 1 }

batchJobsRunning OBJECT-TYPE
```

```
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of batch jobs running.  "
     ::= { batchJobStatistics 2 }

batchJobsHeldRun OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of batch jobs held running.  "
     ::= { batchJobStatistics 3 }

batchJobsEnding OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of batch jobs ending.  "
     ::= { batchJobStatistics 4 }

batchJobsScheduled OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of batch jobs waiting to run
                  or already scheduled.  "
     ::= { batchJobStatistics 5 }

batchJobsHeldOnJobQ OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of batch jobs held on a job queue.
             Note: This means the job itself is held.  "
     ::= { batchJobStatistics 6 }

batchJobsOnHeldJobQ OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of batch jobs on a held job queue.
              Note: This means the whole job queue is held.  "
     ::= { batchJobStatistics 7 }

batchJobsUnassigned OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of batch jobs on an unassigned job queue.  "
     ::= { batchJobStatistics 8 }

batchJobEndedPrtWait OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of batch jobs ended with printer output
              waiting to print.  "
```

```
     ::= { batchJobStatistics 9 }

processes  OBJECT IDENTIFIER  ::= { comtekos400Mib 2 }

psNumProcs OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Number of processes.  "
     ::= { processes 1 }

psTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF PsTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "A simplified version of what is read from WRKACTJOB.  "
     ::= { processes 2 }

psTableEntry OBJECT-TYPE
     SYNTAX  PsTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "A row in the process table.  "
     INDEX {psProcessIndex}
     ::= { psTable 1 }

PsTableEntry ::= SEQUENCE {
   psProcessIndex       INTEGER,
   psSubsystemName      DisplayString,
   psJobName    DisplayString,
   psUser       DisplayString,
   psCpuUtilization     INTEGER,
   psFunction   DisplayString,
   psStatus     DisplayString,
   psActiveJobStatus DisplayString
          }

psProcessIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "This is the index into the process table.  "
     ::= { psTableEntry 1 }

psSubsystemName OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(20))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The subsystem in which the job is running.  "
     ::= { psTableEntry 2 }

psJobName OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(10))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The name of the job as identified to the system.  "
     ::= { psTableEntry 3 }
```

```
psUser OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(10))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The user name identifies the user who submitted the job and
          the user profile under which the job is run.  "
     ::= { psTableEntry 4 }

psCpuUtilization OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Number of milliseconds of processor time used by the job.
"
     ::= { psTableEntry 5 }

psFunction OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(10))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The last high-level function initiated by the job.  This
          field is blank when the OS/400 logged function has not been
          performed.  "
     ::= { psTableEntry 6 }

psStatus OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The status of the job.  Only one status is displayed per
job.
          A blank status field represents a job that is in transition.  "
     ::= { psTableEntry 7 }

psActiveJobStatus OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The active job status.  Only one status is displayed per
job.
          A blank status field represents a job that is in transition.  "
     ::= { psTableEntry 8 }

pscritProcCfgTableSize OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
DESCRIPTION
"This indicates the number of processes listed in the critical process table
"
     ::= { processes 3 }

critProcCfgTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF critProcCfgTableEntry
     ACCESS not-accessible
     STATUS  mandatory
```

```
     DESCRIPTION "Table description "
     ::= { processes 4 }

critProcCfgTableEntry OBJECT-TYPE
     SYNTAX  critProcCfgTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "Row  in table description "
     INDEX {critProcCfgIndex}
     ::= { critProcCfgTable 1 }

critProcCfgTableEntry ::= SEQUENCE {
   critProcCfgIndex        INTEGER,
   critProcCfgName         DisplayString,
   critProcCfgUser         DisplayString,
   critProcCfgSubsystem    DisplayString,
   critProcCfgNumInstancesReq      INTEGER,
   critProcCfgNumActive    INTEGER
          }

critProcCfgIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "This is the index into the critical process table. "
     ::= { critProcCfgTableEntry 1 }

critProcCfgName OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "This is the name of the critical process "
     ::= { critProcCfgTableEntry 2 }

critProcCfgUser OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "This is the user profile that the critical process should
be

running under.  The * character means match anything to the end of the
string. "
     ::= { critProcCfgTableEntry 3 }

critProcCfgSubsystem OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
DESCRIPTION "This is the subsystem under which the critical process should
 run.
       The * character is a wildcard. "
     ::= { critProcCfgTableEntry 4 }

critProcCfgNumInstancesReq OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
```

```
     STATUS  mandatory
     DESCRIPTION "This is the number of instances of the critical process
that
                 should be running on the system.  If there are less than
this
      number of instances, a critical process missing trap will be generated.
"
     ::= { critProcCfgTableEntry 5 }

critProcCfgNumActive OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "This is the actual number instances of critical processes
                  that are currently running on the system "
     ::= { critProcCfgTableEntry 6 }

psCritNumProcs OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Number of critical processes running.  "
     ::= { processes 5 }

psCritTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF psCritTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "A simplified version of what is read from WRKACTJOB
constrained to
                  the jobs specified as critical processes.  "
     ::= { processes 6 }

psCritTableEntry OBJECT-TYPE
     SYNTAX  psCritTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "A row in the critical process table.  "
     INDEX {psProcessIndex}
     ::= { psCritTable 1 }

psCritTableEntry ::= SEQUENCE {
   psCritProcessIndex       INTEGER,
   psCritSubsystemName      DisplayString,
   psCritJobName    DisplayString,
   psCritUser       DisplayString,
   psCritCpuUtilization     INTEGER,
   psCritFunction   DisplayString,
   psCritStatus     DisplayString,
   psCritActiveJobStatus DisplayString
          }

psCritProcessIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "This is the index into the process table.  "
```

```
     ::= { psCritTableEntry 1 }

psCritSubsystemName OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(20))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The subsystem in which the job is running.  "
     ::= { psCritTableEntry 2 }

psCritJobName OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(10))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The name of the job as identified to the system.  "
     ::= { psCritTableEntry 3 }

psCritUser OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(10))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The user name identifies the user who submitted the job and
          the user profile under which the job is run.  "
     ::= { psCritTableEntry 4 }

psCritCpuUtilization OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Number of milliseconds of processor time used by the job.
"
     ::= { psCritTableEntry 5 }

psCritFunction OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(10))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The last high-level function initiated by the job.  This
          field is blank when the OS/400 logged function has not been
          performed.  "
     ::= { psCritTableEntry 6 }

psCritStatus OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The status of the job.  Only one status is displayed per
job.
          A blank status field represents a job that is in transition.  "
     ::= { psCritTableEntry 7 }

psCritActiveJobStatus OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The active job status.  Only one status is displayed per
job.
          A blank status field represents a job that is in transition.  "
```

```
     ::= { psCritTableEntry 8 }




jobq  OBJECT IDENTIFIER  ::= { comtekos400Mib 3 }

jqNumProcs OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Number of jobs in the job queue.  "
     ::= { jobq 1 }

jqTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF JqTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "A simplified version of what is read from WRKSBMJOB.  "
     ::= { jobq 2 }

jqTableEntry OBJECT-TYPE
     SYNTAX  JqTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "A row in the job queue table.  "
     INDEX {jqProcessIndex}
     ::= { jqTable 1 }

JqTableEntry ::= SEQUENCE {
   jqProcessIndex       INTEGER,
   jqUser       DisplayString,
   jqJobName    DisplayString,
   jqType       DisplayString,
   jqStatus     DisplayString
          }

jqProcessIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Process index.  This field is the index into the job
          queue table.  "
     ::= { jqTableEntry 1 }

jqUser OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(10))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The user name identifies the user profile under which the
          job is run. The user name is specified either on the SBMJOB
          (submit job) command, in the job description referred to by
          the BCHJOB or SBMJOB commands, or in the user parameter of
          the job schedule entry.  "
```

```
     ::= { jqTableEntry 2 }

jqJobName OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(10))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The simple job name of the job.  "
     ::= { jqTableEntry 3 }

jqType OBJECT-TYPE
     SYNTAX  DisplayString(SIZE(10))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The type of job.  Possible types are:
               - BATCH:  batch
               - BATCHI: batch immediate
               - MRT:    multiple requestor terminal  "
     ::= { jqTableEntry 4 }

jqStatus OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The status of the job (in two fields).  Possible
          values for the first field are:
               - ACTIVE:  The job has been started.
               - OUTQ:    The job has completed running and has spooled
                          files on an output queue.
               - DSC:     The job is disconnected.
               - JOBQ:    The job is on a job queue, but not the result
                          of a transfer job (TFRJOB) or transfer batch
                          job (TFRBCHJOB) command.
               - TFRJOB:  The job is on a job queue as a result of a
                          transfer job (TFRJOB) command.
               - TFRBCH:  The job is on a job queue as a result of a
                          transfer batch job (TFRBCHJOB) command.
               - SYSREQ:  The job is suspended by a system request.
               - FIN:     The job has finished.
               - END:     The job is ending as the result of the end job
                          (ENDJOB) or the end subsystem (ENDSBS) command.
               - EOJ:     The job is ending for any reason other than end
                          job (ENDJOB) or end subsystem (ENDSBS).
               - MSGW:    The job has a message waiting.
               - SCD:     The job is scheduled for a particular time and
                          date.
          The second field indicates whether the job is being held (HELD)
          or not held (if the field is blank).  "
     ::= { jqTableEntry 5 }

trapinfo  OBJECT IDENTIFIER  ::= { comtekos400Mib 4 }

firstTrapNum OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The sequence number of the oldest trap available in the
          subagent's internal trap table.  This indicates the lowest
```

```
          sequence number that is available to be resent.  "
     ::= { trapinfo 1 }

lastTrapNum OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Last trap sequence number maintained by the subagent.  "
     ::= { trapinfo 2 }

trapNum OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "Trap sequence number. This variable accompanies every trap
          sent by the subagent and may also be used to resend traps to
          the SNMP manager.  To resend a trap message, set this variable
          to the sequence number of the trap that is to be resent.  "
     ::= { trapinfo 3 }

userTrapContFlag OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "This variable is used as a continuation indicator in
userMsg
          traps.  If the text of the userMsg trap is too long to be
          contained in a single userMsg trap, the text is split into
          multiple traps with the userTrapContFlag set to 1 to indicate
          that the data is continued in a subsequent trap and set to 0
          to indicate that this is the final trap for this user data.
          This variable only has meaning in the context of a particular
          userMsg trap and is therefore not accessible by SNMP get and
          set requests.  "
     ::= { trapinfo 5 }

userTrapMsgText OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "This is the text of a userMsg trap. The subagent
disassembles
          user data placed in the SNMP_USR_Q data queue into 255
          byte null terminated character strings.  Each userMsg trap
          also contains a continuation flag to indicate if the user
          message is continued in a subsequent trap. This variable
          only has meaning in the context of a particular userMsg trap
          and is therefore not accessible by SNMP get and set requests.  "
     ::= { trapinfo 6 }

qsysoprMsgGroup  OBJECT IDENTIFIER  ::= { comtekos400Mib 5 }

qsysoprMsgKey OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "This variable provides an identifying key for QSYSOPR
```

```
          messages. This key is part of every qsysoprMsg trap and can
          also be used to request the help text for a QSYSOPR message.
          To retrieve the help text for a QSYSOPR message, set this
          variable to the message key in the subject qsysoprMsg trap
          and the appropriate help text will be sent as a
          qsysoprHelpText trap.  "
     ::= { qsysoprMsgGroup 1 }

qsysoprResp OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "This variable provides the mechanism for responding to
          QSYSOPR inquiry messages.  This variable requires that the
          qsysoprMsgKey received in the qsysoprMsg trap be used as the
          instance for this variable in the set-request.  For example,
          if responding to a qsysoprMsg trap that had a qsysoprMsgKey
          of 1234, a set-request would be performed on instance 1234
          of qsysoprResp (i.e., qsysoprResp.1234).  "
     ::= { qsysoprMsgGroup 2 }

qsysoprMsgSeverity OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "This variable contains the severity of the QSYSOPR message.
          This variable is only available in qsysoprMsg traps.   "
     ::= { qsysoprMsgGroup 5 }

qsysoprMsgID OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "This variable contains the 7 character message ID for a
          QSYSOPR message. This variable is only available in qsysoprMsg
          traps. An example of a message ID is CPF9801.  "
     ::= { qsysoprMsgGroup 6 }

qsysoprMsgType OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "This variable identifies the type of the QSYSOPR message.
          This variable is only available in qsysoprMsg traps.  "
     ::= { qsysoprMsgGroup 7 }

qsysoprNumMsgHelpLines OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "This indicates the number of lines of help text returned
          in a qsysoprHelpText trap. This variable is only available
          in qsysoprMsg traps.  "
     ::= { qsysoprMsgGroup 8 }

qsysoprMsgTxtTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF QsysoprMsgTxtEntry
```

```
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "The QSYSOPR message specified by qsysoprMsgKey.  "
     ::= { qsysoprMsgGroup 9 }

qsysoprMsgTxtEntry OBJECT-TYPE
     SYNTAX  QsysoprMsgTxtEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "A line in the message text.  This variable is only
          available in qsysoprMsg traps.  "
     INDEX {msgRowNumber}
     ::= { qsysoprMsgTxtTable 1 }

QsysoprMsgTxtEntry ::= SEQUENCE {
   qsysoprMessage       DisplayString,
   msgRowNumber         INTEGER
          }

qsysoprMessage OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "The text of the QSYSOPR message.  This variable is only
          available in qsysoprMsg traps.  "
     ::= { qsysoprMsgTxtEntry 1 }

msgRowNumber OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "Index into the qsysoprMsgTxtTable.  This variable is not
          accessible.  "
     ::= { qsysoprMsgTxtEntry 2 }

qsysoprShortMsgText OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "This variable contains the first level message text for the
          message indicated by qsysoprMsgKey. This variable is only
          available in qsysoprMsg traps.  "
     ::= { qsysoprMsgGroup 10 }

qsysoprSendJob OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "This variable contains the name of the job that sent the
          message to the QSYSOPR message queue.  This variable is only
          available in qsysoprMsg traps.  "
     ::= { qsysoprMsgGroup 11 }

qsysoprUserName OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS not-accessible
     STATUS  mandatory
```

```
     DESCRIPTION "This variable contains the name of the user profile that
          sent the message to the QSYSOPR message queue.  This variable
          is only available in qsysoprMsg traps.  "
     ::= { qsysoprMsgGroup 12 }

qsysoprJobNumber OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "This variable contains the job number of the job that sent
          the message to the QSYSOPR message queue.  This variable is
          only available in qsysoprMsg traps.  "
     ::= { qsysoprMsgGroup 13 }

qsysoprProgramName OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "This variable contains the name of the program that sent
          the message to the QSYSOPR message queue.  This variable is
          only available in qsysoprMsg traps.  "
     ::= { qsysoprMsgGroup 14 }

configuration  OBJECT IDENTIFIER  ::= { comtekos400Mib 6 }

cpuWaitTime OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "Number of seconds to wait between updates of the CPU group
          statistics. This variable is equivalent to the CPU_WAIT_TIME
          configuration file parameter.  "
     ::= { configuration 1 }

cpuThreshold OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "Percentage of CPU utilization at which a cpuExcessive trap
          should be sent to the SNMP manager. This value is expressed
          in tenths of a percent.  Thus 95% would be denoted 950.  This
          variable is equivalent to the CPU_TRAP_THRESHOLD configuration
          file parameter.  "
     ::= { configuration 2 }

cpuResendPeriod OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "Number of times the CPU statistics should be updated before
          repeating a cpuExcessive trap. This variable is equivalent to
          the CPU_RESEND_TRAP_COUNT configuration file parameter.  "
     ::= { configuration 3 }

diskWaitTime OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
```

```
     STATUS  mandatory
     DESCRIPTION "Number of seconds to wait between updates of the disk group
          statistics. This variable is equivalent to the DISK_WAIT_TIME
          configuration file parameter.  "
     ::= { configuration 4 }

diskFullThreshold OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "Percentage of system storage pool (disk) utilization at
which
          a diskFull trap should be sent to the SNMP manager.  This value
          is expressed in ten thousandths of a percent.  For example,
          44123 would indicate 4.4123%.  This variable is equivalent to
          the DISK_FULL_THRESHOLD configuration file parameter.  "
     ::= { configuration 5 }

diskFullClearThreshold OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "Percentage of system storage pool (disk) utilization at
which
          a diskClear trap should be sent to the SNMP manager to indicate
          the disk full condition no longer exists.  This value is
          expressed n ten thousandths of a percent.  For example, 44123
          would indicate 4.4123%. This variable is equivalent to the
          DISK_FULL_CLR_THRESHOLD configuration file parameter.  "
     ::= { configuration 6 }

diskResendPeriod OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "Number of times to gather disk group statistics before
          repeating a diskFull trap. This variable is equivalent to the
          DISK_RESEND_TRAP_COUNT configuration file parameter.  "
     ::= { configuration 7 }

jobqWaitTime OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "Number of seconds to wait between updates of the job queue
          group statistics.  This variable is equivalent to the
          JQ_WAIT_TIME configuration file parameter.  "
     ::= { configuration 8 }

psWaitTime OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "Number of seconds to wait between updates of the active
          process group statistics.  This variable is equivalent to
          the PS_WAIT_TIME configuration file parameter.  "
     ::= { configuration 9 }
```

```
psReqProcFileName OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-write
     STATUS  mandatory
    DESCRIPTION "The fully qualified name of the critical process
configuration
          file.  The critical process configuration file specifies jobs
          which are required to be active on the system.  If a required
          job is not active on the system, a critProcCfgMissing trap is
          generated for the missing job.  This variable permits SNMP
          set-requests so that the critical process list may be modified
          without stopping and restarting the subagent. This variable is
          equivalent to the INITIAL_REQ_PS_FILENAME configuration file
          parameter.  "
     ::= { configuration 11 }

qsysoprSeverity OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "The minimum QSYSOPR message severity level which is to be
          forwarded to the SNMP manager. QSYSOPR messages with a severity
          level below this value are not sent as qsysoprMsg traps.
          This variable is equivalent to the MIN_QSYSOPR_SEV_TO_SEND
          configuration file parameter.  "
     ::= { configuration 12 }

qhstWaitTime OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "Number of seconds to wait before rereading the QHST file.
          This variable is equivalent to the QHST_WAIT_TIME configuration
          file parameter.  "
     ::= { configuration 13 }

trapThrottle OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
    DESCRIPTION "Number of hundredths of seconds to wait between sending
traps.
          This variable may be used as a throttle to prevent traps
          generated by the subagent from flooding the network. This
          variable is equivalent to the HUNDREDTHS_SEC_BETWEEN_TRAPS
          configuration file parameter.  "
     ::= { configuration 14 }

qsysoprMsgFilterFile OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-write
     STATUS  mandatory
    DESCRIPTION "The fully qualified filename for the QSYSOPR message ID
filter
          file. The filter can be configured to either forward or discard
those
```

```
          message IDs which are specified in the filter. This variable is
          equivalent to the QSYSOPR_MSG_FILTER_FILE configuration file
          parameter."
     ::= { configuration 15 }

qsysoprMaxMsgAge OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "Maximum age of QSYSOPR messages to forward to the SNMP
          manager as qsysoprMsg traps.  This value is specified in
          minutes.  This variable is equivalent to the QSYSOPR_MSG_MAX_
          AGE_MINUTES configuration file parameter.  "
     ::= { configuration 16 }

psResendPeriod OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "The number of times to check for a critical process before
repeating
                  a critical process missing trap. If this is set to a large
number, it
                  will make it seem like a critical process missing trap is
only sent once. "
     ::= { configuration 17 }

remoteConsole  OBJECT IDENTIFIER  ::= { comtekos400Mib 7 }

rConsCommand OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "The remote console command to be executed on the AS/400.
          This feature permits the user to select if the output of
          the command is to be captured and converted into a series
          of userMsg traps. A remote console command has the format:

               CMD<xxx> OUT<yyy> TAG<zzz>

          where xxx is the OS/400 command to be executed, yyy is the name
          of the file that the OS/400 command generates for print output,
          and zzz is a user selected alphanumeric string which is to
          appear in each userMsg trap that is generated.  For example,

               CMD<WRKACTJOB OUTPUT(*PRINT)> OUT<QPDSPAJB> TAG<actJobs>

          If the results of the command are not to be sent as userMsg
          traps, both the OUT and TAG fields should be omitted.

          The OUT field is required if the print file produced by the
          command is to be sent as a series of userMsg traps to the
          SNMP manager. When using the OUT field, it is required that
          the CMD field include OUTPUT(*PRINT) to direct the output
          of the command to a file.  The name that should be specified
          for the OUT field is dependent on the output file that is
          generated by the specified OS/400 command.
```

```
          The TAG field is intended to assist the user in distinguishing
          userMsg traps that are generated by multiple remote console
          commands. The TAG field is optional.  "
     ::= { remoteConsole 1 }

-- formerly file: comtektp.mib

coldStart TRAP-TYPE
     ENTERPRISE  comtekos400Mib
     VARIABLES   { trapNum, os400SysName }
     DESCRIPTION
          "A coldStart trap signifies that the sending protocol entity
          is reinitializing itself such that the agent's configuration
          or the protocol entity implementation may be altered."
     ::= 0

diskFull TRAP-TYPE
     ENTERPRISE  comtekos400Mib
     VARIABLES   { trapNum, cpuPercentSystemASP }
     DESCRIPTION
          "Disk full.  The disk space in use has reached or exceeded the
          threshold specified by diskFullThreshold. This trap is repeated
          as specified by diskResendPeriod while the disk full condition
          persists.  When this situation is resolved, a diskClear trap is
          sent."
     ::= 1

diskClear TRAP-TYPE
     ENTERPRISE  comtekos400Mib
     VARIABLES   { trapNum, cpuPercentSystemASP  }
     DESCRIPTION
          "Disk full clear.  The disk space usage has gone below the
          diskFullClearThreshold threshold after having previously exceeded
          the diskFullThreshold threshold."
     ::= 2

cpuExcessive TRAP-TYPE
     ENTERPRISE  comtekos400Mib
     VARIABLES   { trapNum, cpuPercentProcessUnitUsed }
     DESCRIPTION
          "Excessive CPU utilization. The total CPU utilization has
          reached or exceeded the threshold specified by cpuThreshold.
          This trap is repeated as specified by cpuResendPeriod while
          the excessive utilization condition persists.  When this
          situation is resolved, a cpuClear trap is sent."
     ::= 3

critProcCfgMissing TRAP-TYPE
     ENTERPRISE  comtekos400Mib
     VARIABLES   { trapNum, psJobName, psUser }
     DESCRIPTION
          "A user specified critical process is not active.  A process
          that the user defined as critical in the critical process
          configuration file is not running. This trap is repeated
          every time the process statistics are updated (as specified
          by psWaitTime) and the named process is not located."
```

```
     ::= 4

qsysoprMsg TRAP-TYPE
     ENTERPRISE  comtekos400Mib
     VARIABLES   { trapNum, qsysoprMsgKey, qsysoprMsgID, qsysoprMsgType,
                   qsysoprMsgSeverity, qsysoprShortMsgText, qsysoprSendJob,
                   qsysoprUserName, qsysoprJobNumber, qsysoprProgramName }
     DESCRIPTION
          "A QSYSOPR message.  An event has occurred that posted a
          message in the QSYSOPR message queue."
     ::= 5

userMsg TRAP-TYPE
     ENTERPRISE  comtekos400Mib
     VARIABLES   { trapNum, userTrapContFlag, userTrapMsgText }
     DESCRIPTION
          "User queue message.  A message has been written by a user
          process to the SNMP_USR_Q data queue.  This message is called
          a user defined trap."
     ::= 6

qsysoprHelpText TRAP-TYPE
     ENTERPRISE  comtekos400Mib
     VARIABLES   { trapNum, qsysoprMsgKey, qsysoprMsgID,
                   qsysoprNumMsgHelpLines, qsysoprMsgTxtTable }
     DESCRIPTION
          "This trap is the result of requesting the help message text
          part of a QSYSOPR message.  The message key uniquely identifies
          the QSYSOPR message.  Each line of the help text occupies a
          separate instance of qsysoprMessage.  The qsysoprNumMsgHelpLines
          variable indicates the number of qsysoprMessage instances and
          therefore the number of help text lines that make up this trap."
     ::= 7

cpuClear TRAP-TYPE
     ENTERPRISE  comtekos400Mib
     VARIABLES   { trapNum, cpuPercentProcessUnitUsed }
     DESCRIPTION
         "This trap indicates that the excessive CPU usage that was
         reported by a cpuExcessive trap has been resolved, i.e.,
         the CPU utilization has dropped below the level specified
         by cpuThreshold."
     ::= 10

activeJobStatus TRAP-TYPE
     ENTERPRISE  comtekos400Mib
     VARIABLES   { trapNum, psJobName, psUser,psSubsystemName,
                   psActiveJobStatus }
     DESCRIPTION
          "A user specified critical process does not have the expected

          active job status. This trap is repeated
          every time the process statistics are updated (as specified
          by psWaitTime) and the named process does not have the
          expected active job status."
     ::= 14
```

```
-- formerly file: cmtkgenq.mib
--
-- This is the User Specified Message Queue Monitor MIB for
-- NM*Server for OS/400.  It is subject to revision
-- during product development/enhancement.

cmtkGenQCfg  OBJECT IDENTIFIER  ::= { os400genericQmonitor 1 }

cmtkGenQVersion OBJECT-TYPE
     SYNTAX DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "subagent version   "
     ::= { cmtkGenQCfg 1 }

os400monitoredQCount OBJECT-TYPE
     SYNTAX INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "number of message queues being monitored"
     ::= { cmtkGenQCfg 2 }

os400MonQCfgTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF Os400MonQCfgTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "Table description"
     ::= { cmtkGenQCfg 3 }

os400MonQCfgTableEntry OBJECT-TYPE
     SYNTAX Os400MonQCfgTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "Row  in table description"
     INDEX {os400MsgQueueIndex}
     ::= { os400MonQCfgTable 1 }

Os400MonQCfgTableEntry ::= SEQUENCE {
   os400MsgQueueIndex   INTEGER,
   os400MsgQueueName    DisplayString,
   os400MsgQueueLib     DisplayString,
   os400MsgQueueMinSeverity     INTEGER,
   os400MsgQueueMaxAge  INTEGER,
   os400MsgQueueFilterFile      DisplayString
          }


os400MsgQueueIndex OBJECT-TYPE
     SYNTAX INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "This is the index for the message
                  queue monitor configuration table. "
     ::= { os400MonQCfgTableEntry 1}

os400MsgQueueName OBJECT-TYPE
     SYNTAX DisplayString
```

```
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "This is the name of the queue that
                 is being monitored."
     ::= { os400MonQCfgTableEntry 2 }

os400MsgQueueLib OBJECT-TYPE
     SYNTAX DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "This is the library where the message queue is located"
     ::= { os400MonQCfgTableEntry 3 }

os400MsgQueueMinSeverity OBJECT-TYPE
     SYNTAX INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "This is the minimum severity message to
                  forward to the SNMP Manager for this message queue"
     ::= { os400MonQCfgTableEntry 4 }

os400MsgQueueMaxAge OBJECT-TYPE
     SYNTAX INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "This is the maximum message age in minutes for
                  filtering messages from this message queue. "
     ::= { os400MonQCfgTableEntry 5 }

os400MsgQueueFilterFile OBJECT-TYPE
     SYNTAX DisplayString
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "The name of the file that contains the specifications
            for filtering messages based on the message ID.  The filter
            can be configured to either forward or discard those message
            IDs which are specified in the filter."
     ::= { os400MonQCfgTableEntry 6 }

os400MsgRspTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF Os400MsgRspTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "This is a sparse table that is used for
              responding to inquiry messages from the monitored message
queues.
              The method for responding is to use the queue name and message
              number as indices to set the response."
     ::= { os400genericQmonitor 2 }

os400MsgRspTableEntry OBJECT-TYPE
     SYNTAX Os400MsgRspTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "Row  in table description"
     INDEX {os400RspMsgQName, os400RspMsgNumber}
     ::= { os400MsgRspTable 1 }
```

```
Os400MsgRspTableEntry ::= SEQUENCE {
   os400RspMsgQName     DisplayString,
   os400RspMsgNumber    INTEGER,
   os400RspMsgResponse  DisplayString
          }

os400RspMsgQName OBJECT-TYPE
     SYNTAX DisplayString (SIZE (1..114))
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "This is the name of the message queue
                  that is being responded to."
     ::= { os400MsgRspTableEntry 1 }

os400RspMsgNumber OBJECT-TYPE
     SYNTAX INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "This is the message number of the message
            that a response is being sent to"
     ::= { os400MsgRspTableEntry 2 }

os400RspMsgResponse OBJECT-TYPE
     SYNTAX DisplayString
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "This is the response that is being sent
                  to the monitored message
                  queue for the specified message number"
     ::= { os400MsgRspTableEntry 3 }



cmtkGenQTrapVars  OBJECT IDENTIFIER  ::= { os400genericQmonitor 3 }

os400GenQTrapMsgQName OBJECT-TYPE
     SYNTAX DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "Name of the message queue that received a message"
     ::= { cmtkGenQTrapVars 1 }

os400GenQTrapMsgKey OBJECT-TYPE
     SYNTAX INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The message key for the received message"
     ::= { cmtkGenQTrapVars 2 }

os400GenQTrapMsgSeverity OBJECT-TYPE
     SYNTAX INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The severity of the message"
     ::= { cmtkGenQTrapVars 3 }
```

```
os400GenQTrapMsgId OBJECT-TYPE
     SYNTAX DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The message ID for the message"
     ::= { cmtkGenQTrapVars 4 }

os400GenQTrapMsgType OBJECT-TYPE
     SYNTAX DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The type of message"
     ::= { cmtkGenQTrapVars 5 }

os400GenQTrapMsgText OBJECT-TYPE
     SYNTAX DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The message text"
     ::= { cmtkGenQTrapVars 6 }

os400GenQTrapMsgSendJob OBJECT-TYPE
     SYNTAX DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The job that sent the message"
     ::= { cmtkGenQTrapVars 7 }

os400GenQTrapMsgUserName OBJECT-TYPE
     SYNTAX DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The user profile that sent the message"
     ::= { cmtkGenQTrapVars 8 }

os400GenQTrapMsgJobNumber OBJECT-TYPE
     SYNTAX DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The job number of the job that sent the message"
     ::= { cmtkGenQTrapVars 9 }

os400GenQTrapMsgProgramName OBJECT-TYPE
     SYNTAX DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The name of the program that sent the message"
     ::= { cmtkGenQTrapVars 10 }


qMonitor TRAP-TYPE
     ENTERPRISE  comtekos400Mib
     VARIABLES   { trapNum, os400GenQTrapMsgQName, os400GenQTrapMsgKey,
                   os400GenQTrapMsgSeverity, os400GenQTrapMsgId,
                   os400GenQTrapMsgType, os400GenQTrapMsgText,
                   os400GenQTrapMsgSendJob, os400GenQTrapMsgUserName,
                   os400GenQTrapMsgJobNumber, os400GenQTrapMsgProgramName }
```

```
     DESCRIPTION
         "This trap indicates that a message was received on a user
         specified message queue."
     ::= 11

-- formerly file: comtekcmn.mib
--
-- This is the Communications Line, Controller, and Device MIB for
-- NM*Server for OS/400.  It is subject to revision
-- during product development/enhancement.

cmtkCmnCfg  OBJECT IDENTIFIER  ::= { os400cmn 1 }

cmtkCmnVersion OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "subagent version   "
     ::= { cmtkCmnCfg 1 }

cmtkCmnWaitTime OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-write
     STATUS  mandatory
     DESCRIPTION "The number of minutes to wait between checking all of the
                  communications lines, controllers, and devices.  "
     ::= { cmtkCmnCfg 2 }

os400CmnLineCount OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "num lines   "
     ::= { os400cmn 2 }

os400CmnLineTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF Os400CmnLineTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "Table description   "
     ::= { os400cmn 3 }

os400CmnLineTableEntry OBJECT-TYPE
     SYNTAX  Os400CmnLineTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "Row  in table description   "
     INDEX {os400CmnLineIndex}
     ::= { os400CmnLineTable 1 }

Os400CmnLineTableEntry ::= SEQUENCE {
   os400CmnLineName  DisplayString,
   os400CmnLineStatusText   DisplayString,
   os400CmnLineStatusNumeric      INTEGER,
   os400CmnLineCatagory    DisplayString,
   os400CmnLineTextDescription    DisplayString,
   os400CmnLineIndex       INTEGER
```

```
          }

os400CmnLineName OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The name of the configured communications line.  "
     ::= { os400CmnLineTableEntry 1 }

os400CmnLineStatusText OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "status as text  "
     ::= { os400CmnLineTableEntry 2 }

os400CmnLineStatusNumeric OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "line status numeric code  "
     ::= { os400CmnLineTableEntry 3 }

os400CmnLineCatagory OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "the type of line e.g. *ETH, *FR, etc  "
     ::= { os400CmnLineTableEntry 4 }

os400CmnLineTextDescription OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The long text description of the line  "
     ::= { os400CmnLineTableEntry 5 }

os400CmnLineIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "line number   "
     ::= { os400CmnLineTableEntry 10 }

os400CmnCtlrCount OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of OS/400 communications
                  controllers in the table  "
     ::= { os400cmn 4 }

os400CmnCtlrTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF Os400CmnCtlrTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "Table description  "
```

```
     ::= { os400cmn 5 }

os400CmnCtlrTableEntry OBJECT-TYPE
     SYNTAX  Os400CmnCtlrTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "Row  in table description  "
     INDEX {os400CmnCtlrIndex}
     ::= { os400CmnCtlrTable 1 }

Os400CmnCtlrTableEntry ::= SEQUENCE {
   os400CmnCtlrName  DisplayString,
   os400CmnCtlrStatusText   DisplayString,
   os400CmnCtlrStatusNumeric      INTEGER,
   os400CmnCtlrCatagory    DisplayString,
   os400CmnCtlrTextDescription    DisplayString,
   os400CmnCtlrIndex       INTEGER
         }

os400CmnCtlrName OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The name of the communications controller  "
     ::= { os400CmnCtlrTableEntry 1 }

os400CmnCtlrStatusText OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The text describing the status of the
                  communications controller  "
     ::= { os400CmnCtlrTableEntry 2 }

os400CmnCtlrStatusNumeric OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The status of the communications controller
                  represented numerically  "
     ::= { os400CmnCtlrTableEntry 3 }

os400CmnCtlrCatagory OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "the category of communications controller  "
     ::= { os400CmnCtlrTableEntry 4 }

os400CmnCtlrTextDescription OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The long text description
                  of the communications controller  "
     ::= { os400CmnCtlrTableEntry 5 }
```

```
os400CmnCtlrIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The index into the communications
                  controller table  "
     ::= { os400CmnCtlrTableEntry 10 }

os400CmnDevCount OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The number of entries in the communications
                  device table  "
     ::= { os400cmn 6 }

os400CmnDevTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF Os400CmnDevTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "Table description  "
     ::= { os400cmn 7 }

os400CmnDevTableEntry OBJECT-TYPE
     SYNTAX  Os400CmnDevTableEntry
     ACCESS not-accessible
     STATUS  mandatory
     DESCRIPTION "Row  in table description  "
     INDEX {os400CmnDevIndex}
     ::= { os400CmnDevTable 1 }

Os400CmnDevTableEntry ::= SEQUENCE {
   os400CmnDevName   DisplayString,
   os400CmnDevStatusText    DisplayString,
   os400CmnDevStatusNumeric       INTEGER,
   os400CmnDevCatagory     DisplayString,
   os400CmnDevTextDescription     DisplayString,
   os400CmnDevIndex  INTEGER
          }

os400CmnDevName OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The name of the communications device  "
     ::= { os400CmnDevTableEntry 1 }

os400CmnDevStatusText OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The status of the communications
      device displayed as text  "
     ::= { os400CmnDevTableEntry 2 }

os400CmnDevStatusNumeric OBJECT-TYPE
     SYNTAX  INTEGER
```

```
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The status of the communications
      device represented numerically  "
     ::= { os400CmnDevTableEntry 3 }

os400CmnDevCatagory OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The category of the communications device  "
     ::= { os400CmnDevTableEntry 4 }

os400CmnDevTextDescription OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "The long text description of the
                  communications device  "
     ::= { os400CmnDevTableEntry 5 }

os400CmnDevIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS read-only
     STATUS  mandatory
     DESCRIPTION "the index into the communications device table  "
     ::= { os400CmnDevTableEntry 10 }

END
```