AF/OPERATOR and Subsystem Managers User’s Guide
Integrated Resource Manager

Version 500

GC32-9219-00

September 2002

Candle Corporation
201 North Douglas Street
El Segundo, California 90245
## Contents

*Preface* ........................................... 9  
  *About this Document* .......................... 9  
  *Prerequisites* ................................. 9  
  *Adobe Portable Document Format* .......... 10  
  *Introduction* .................................. 10  
  *Printing problems?* ......................... 10  
  *Documentation Conventions* ............... 12  
    *Introduction* ................................ 12  
    *Panels and figures* ....................... 12  
    *Revision bars* ............................... 12  
    *Variables and literals* ................... 12  
    *Symbols* ................................... 13  
  *Documentation Set* ......................... 14  
    *Introduction* ................................ 14  
    *Online Documentation* .................... 14  
    *Printed documentation* ................. 14  

*What’s New* .................................... 17  

*Chapter 1. AF/REMOTE Manager* .......... 19  
  *AF/REMOTE Manager Components* .......... 20  
    *Automation Application Tables* ........ 20  
    *Service Subroutines* ..................... 20  
    *AF/REMOTE System Variables* .......... 22  
    *AFR Command Syntax* ..................... 22  
    *AR Command On-line Help* .............. 28  

*Chapter 2. Peer-to-Peer Manager* ....... 29  
  *Components of the P2P* ................... 31  
    *Automation Application Tables* ........ 31  
    *Service Subroutines* ..................... 31
Contents

AF/OPERATOR System Variables ...................................... 32
Automation Table Component ........................................ 33
Introduction ......................................................... 33
Adding or Updating Links ............................................ 33
Control Options ..................................................... 34
Using the Service Subroutines ....................................... 36
Introduction ......................................................... 36
Starting the Service Subroutines .................................... 36
Controlling Peer-to-Peer Manager Resources ..................... 36
P2P Command Syntax ................................................ 37
Functions ............................................................ 38
Stop Options ......................................................... 39
Link Options ......................................................... 39
SMFID ................................................................. 40
Parms ................................................................. 40
Controlling Peer-to-Peer Links ....................................... 41
P2P Command On-line Help .......................................... 41

Chapter 3.  OMEGAMON Logon Manager ............................. 43
OMEGAMON Logon Manager Components .......................... 44
Automation Application Table ....................................... 44
Service Subroutines .................................................. 46

Chapter 4.  POVI Session Manager .................................... 49
AF/OPERATOR Manager Components ............................... 51
Automation Table ..................................................... 51
Automation Table ..................................................... 52
@OVICNTL - POVI Session Manager Control Table ............... 52
Service Subroutines .................................................. 53
Overview ............................................................ 53
Starting the Service Subroutines .................................... 53

Chapter 5.  Communication Manager ................................. 55
Communication Manager Components .............................. 57
ISPF User Interface .................................................. 57
Link Definition Panel: Field Descriptions ........................ 58
REXX Tables ........................................................ 64
Overview ............................................................ 64
## Contents

### Chapter 10. XRF Manager
- XRF Manager Components .......................................... 127
  - Automation Application Tables ................................. 127
  - Service Subroutines ............................................. 128
  - AF/OPERATOR System Variables .............................. 129
  - Automation Tables .............................................. 130
- Controlling XRF Manager Resources .......................... 132
  - Starting the Service Subroutines ............................ 132
  - Using the Service Subroutines ............................... 133
  - XRF Command Syntax ........................................ 133
  - Functions ..................................................... 135
  - XRF Command On-line Help ................................ 135

### Appendix A. Customer Support
- Introduction .................................................... 137
- Electronic support ............................................. 137
- Telephone support ............................................ 138
- Customer support locations and numbers ................. 138
- Incident documentation ..................................... 140
- Ensuring your satisfaction with customer support ....... 140
## List of Figures

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AFR Command Syntax</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>P2P Command Syntax</td>
<td>37</td>
</tr>
<tr>
<td>3</td>
<td>COM Manager Syntax</td>
<td>66</td>
</tr>
<tr>
<td>4</td>
<td>JES Command Syntax</td>
<td>85</td>
</tr>
<tr>
<td>5</td>
<td>XRF Command Syntax</td>
<td>133</td>
</tr>
</tbody>
</table>
About this Document

This guide provides instructions on how to use the Integrated Resource Manager (IRM) Version 500 AF/OPERATOR and Subsystem Managers.

This guide is directed to the systems operator(s) responsible for using the Integrated Resource Manager. It assumes that you have some previous experience in using mainframe software and knowledge of MVS systems and SMP/E.

Prerequisites

This guide assumes that AF/OPERATOR, or OMEGACENTER Gateway for MVS, and the Integrated Resource Manager has been SMP/E installed and that maintenance has been applied to bring them up to current levels of service. For information on installing AF/OPERATOR refer to Installing Candle Products on MVS (IC51-6057).

Refer to the AF/OPERATOR and the IRM Configuration and Customization Guides for additional information on installing and configuring AF/OPERATOR and IRM respectively.
Adobe Portable Document Format

Introduction

Candle supplies documentation in the Adobe Portable Documentation Format (PDF). The Adobe Acrobat Reader prints PDF documents with the fonts, formatting, and graphics as contained in the original document. To print a Candle document, perform the following:

1. Specify the print options for your system. From the Acrobat Reader Menu bar, select File > Print Setup... and make your selections. A setting of 300 dpi is highly recommended as is duplex printing if your printer supports it.

2. To start printing, select File > Print on the Acrobat Reader Menu bar.

3. On the Print popup, select one of the Print Range options for:
   - a single page
   - a range of pages
   - all of the document

(Optional)

4. To fit oversize pages to the paper size currently loaded on your printer, select the Shrink to Fit option.

Printing problems?

Your printer usually determines the print quality of your output. Sometimes printing problems can occur. If you experience printing problems, potential areas to check are:

- settings for your printer and printer driver. (The 300 dpi settings for both your driver and printer should be the same. A setting of 300 dpi is recommended.)

- the printer driver you are using. (You may need a different printer driver or the Universal Printer driver from Adobe. This free printer driver is available at www.adobe.com.)
Adobe Portable Document Format

- the halftone/graphics color adjustment for printing color on black and white printers. (Check the printer properties under Start > Settings > Printer. For more information, see the online help for the Acrobat Reader.)
- the amount of available memory in your printer. (Insufficient memory can cause a document or graphics to fail to print.)

For additional information on printing problems, refer to the documentation for your printer or contact your printer manufacturer.
Documentation Conventions

Introduction
Candle documentation adheres to accepted typographical conventions for command syntax. Conventions specific to Candle documentation are discussed in the following sections.

Panels and figures
The panels and figures in this document are representations. Actual product panels may differ.

Revision bars
Revision bars (|) may appear in the left margin to identify new or updated material.

Variables and literals
In examples of command syntax, uppercase letters are actual values (literals) that the user should type; lowercase letters are used for variables that represent data supplied by the user. Default values are underscored.

LOGON APPLID(cccccccc)
In the above example, you type \texttt{LOGIN APPLID} followed by an application identifier (represented by \texttt{cccccccc}) within parentheses. The number of characters indicates the maximum allowable length of the variable.

\textit{Note:} In ordinary text, variable names appear in italics.
Symbols

The following symbols may appear in command syntax.

Table 1.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The ‘or’ symbol is used to denote a choice. Either the argument on the left or the argument on the right may be used. For example: YES</td>
</tr>
<tr>
<td>[ ]</td>
<td>Denotes optional arguments. Those arguments not enclosed in square brackets are required. For example: APPLDEST DEST [ALTDEST] In this example, DEST is a required argument and ALTDEST is optional.</td>
</tr>
<tr>
<td>{ }</td>
<td>Some documents use braces to denote required arguments, or to group arguments for clarity. For example: COMPARE {workload} - REPORT={SUMMARY</td>
</tr>
<tr>
<td>_</td>
<td>Default values are underscored. For example: COPY infile outfile - [COMPRESS=YES</td>
</tr>
<tr>
<td>b</td>
<td>The symbol b indicates a blank space, when needed for clarity.</td>
</tr>
</tbody>
</table>
Documentation Set

Introduction

Candle provides a complete set of documentation for the Integrated Resource Manager. Each manual in this documentation set contains a specific type of information to help you use the product.

Candle welcomes your comments and suggestions for changes or additions to the documentation set. A user comment form, located at the back of each manual, provides simple instructions for communicating with Candle's Information Development department. You can also send electronic mail to UserDoc@candle.com. Please include the product name, version, and book title in the subject line. To order additional manuals, contact Candle Customer Support.

Online Documentation

All documents in the Integrated Resource Manager documentation set are available online.

Printed documentation

The documentation listed in the following table is available for the Integrated Resource Manager. To order additional product manuals, contact your Candle Support Services representative.
### Table 2. Integrated Resource Manager Documentation

<table>
<thead>
<tr>
<th>Document Number</th>
<th>Document Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMC99-4833</td>
<td>Candle Products Program Directory</td>
<td>Provides installation instructions and details all other installation consideration.</td>
</tr>
</tbody>
</table>
Summary

This section details the new features and enhancements incorporated into the latest release of the IRM AF/OPERATOR and Subsystem Managers User’s Guide Version 500. These changes reflect recent IRM enhancements.

These enhancements include

- The addition of the Communications Manager to the available Subsystem Managers.
- Identifying AF/REMOTE prohibited command combinations. These combinations appear before the AF/REMOTE command syntax figure on page 22.
Introduction

AF/REMOTE supports and enhances data center automation software such as AF/OPERATOR, availability management software such as OMEGACENTER Gateway for MVS (OG/MVS), and can be used as part of a complete service-level enterprise management package such as OMEGACENTER.

AF/REMOTE has the ability to communicate with IBM's NetView product through the AF/OPERATOR or OG/MVS NetView Interface. Using REXX, you can trap messages, parse relevant information, then take appropriate action. A set of REXX scripts, running under AF/REMOTE, traps messages from the AF/OPERATOR or OG/MVS MVS console. An optional response can be sent back to NetView.

The AF/REMOTE Manager is used to maintain the bi-directional communication link between AF/OPERATOR and AF/REMOTE and to provide AF/REMOTE with host services.

Chapter Contents

AF/REMOTE Manager Components ........................................... 20
Automation Application Tables ............................................. 20
Service Subroutines ............................................................. 20
AF/REMOTE System Variables ............................................. 22
AFR Command Syntax ......................................................... 22
AR Command On-line Help ................................................... 28
AF/REMOTE Manager Components

Automation Application Tables

To manage the AF/REMOTE to AF/OPERATOR communication link, the AF/REMOTE Manager uses information stored in two tables that are dynamically created by IRM and are not visible to the user. The tables are:

@AFRCNTL  AF/REMOTE Control Table. This table maintains the AF/REMOTE Console definitions.

@AFRSTAT  AF/REMOTE Session Data Table. This table maintains the AF/REMOTE session data between AF/OPERATOR and AF/REMOTE.

These tables are loaded into above-the-line storage at AF/OPERATOR and Automation Application initialization time.

Service Subroutines

The AF/REMOTE Manager service subroutines are AF/OPERATOR command file which perform the services needed to manage the AF/OPERATOR to AF/REMOTE communication link. The service subroutines is comprised of the following exec:

@AFRINIT  This exec initializes the AF/REMOTE Manager environment. It creates an MVS console command, called AFR, that allows you to perform various management functions on a single AF/REMOTE console or group of consoles. This exec invokes @AFRMAIN exec at AF/OPERATOR start-up to initialize and add all of the AF/REMOTE command traps controlled by the AF/REMOTE Manager.

@AFRMAIN  This exec processes AF/REMOTE Manager commands issued to manage AF/REMOTE consoles. @AFRMAIN is responsible for: validating and processing command text, responding to operator inquiries, and activating or inactivating AF/REMOTE consoles.
AF/REMOTE Manager Components

Starting the Service Subroutines
The AF/REMOTE Manager is automatically initialized at AF/OPERATOR initialization time by the Application Manager which will instruct the Table Manager to load the @APLAOCM Control Table into above-line-storage.

Using the Service Subroutines
The automation available with the AF/OPERATOR Manager is accessed via the KAUAINIT exec that is executed at IRM Initialization.

@AFRHELP
This exec provides the AF/REMOTE Manager on-line help facility.

@AFRPAGE
This exec is invoked by application error messages to page the application duty programmer. It calls AF/Remote to set off the page and wait for the call back.

@AFRXRB1
This exec is called as a Pre-recovery Exit for AF/Remote. It is used to bypass recovery of AF/REMOTE if the communication link is still active.

@AFRACTV
This exec is used to check the state of the AF/OPERATOR <= AF/REMOTE communication link. It will return a 1 if communication is active and 0 if not.

@AFRCALL
This exec is called by application REXX code as an interface to request and process an AF/Remote function. It will return a 1 if the requested function is successful and 0 if the function failed or timed out.
AF/REMOTE Manager Components

AF/REMOTE System Variables

The AF/REMOTE Manager creates the following AF/OPERATOR system variables to track AF/REMOTE status over time:

@AR#####F  AF/REMOTE Checkpoint Flags
@AR#####S  AF/REMOTE Communication State

AFR Command Syntax

The following table shows the AFR available commands. To protect against unintentional command errors, certain function/object/option combinations are prohibited. This includes:

- Functions INIT, START, STOP, RESTART, and SHUTDOWN with object ALL or * with the FORCE option.

**FIGURE 1. AFR Command Syntax**

<table>
<thead>
<tr>
<th>AFR function</th>
<th>object</th>
<th>option</th>
</tr>
</thead>
<tbody>
<tr>
<td>INIT</td>
<td>ALL</td>
<td>FORCE</td>
</tr>
<tr>
<td>START</td>
<td>*</td>
<td>MAINT</td>
</tr>
<tr>
<td>STOP</td>
<td>CONS_ADR</td>
<td>CHKP</td>
</tr>
<tr>
<td>RESTART</td>
<td>PATTERN</td>
<td></td>
</tr>
<tr>
<td>SHUTDOWN</td>
<td></td>
<td>CONSOLE=consid</td>
</tr>
<tr>
<td>EXCEPT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPDATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOGCHECK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOGSWITCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETTIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SETDATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SETTIME</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## AF/REMOTE Manager Components

<table>
<thead>
<tr>
<th>AFR function</th>
<th>object</th>
<th>option</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHOW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STATUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISPLAY</td>
<td></td>
<td>ACTIVITY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTIVITY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SESSION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HARDWARE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR_ACTIVE()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR_EMULATOR()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR_PING()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR_RESTART()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR_SCANSTRT()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR_SESSION()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR_START()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR_WHOAMI()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR_NUMBERS()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR_TITLES()</td>
</tr>
<tr>
<td>TBPUT</td>
<td></td>
<td>HWC_ACTIVE()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HWC_EMULATOR()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HWC_SESSION()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HWC_VENDOR()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REQ_COUNT()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REQ_CYCLE()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REQ_FAILED()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REQ_INVALID()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REQ_OK()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REQ_QUEUE()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REQ_REJECT()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REQ_UNAUTH()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REQ_WTOR()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FRMT(msg_frmt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SUBS(msg_subs)</td>
</tr>
</tbody>
</table>
### Functions

The function parameter is used to tell the AF/REMOTE Manager which action to take against a AF/REMOTE console. Many, but not all, functions have abbreviations. Acceptable abbreviations are noted with the function’s description.

<table>
<thead>
<tr>
<th><strong>AFR function</strong></th>
<th><strong>object</strong></th>
<th><strong>option</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>WTO</td>
<td>MSGID</td>
<td>ROUT(msg_rout)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DESC(msg_desc)</td>
</tr>
<tr>
<td>WTOR</td>
<td></td>
<td>CNID(msg_cnid)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CNNM(msg_cnnm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JOBN(msg_jobn)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JOBD(msg_jobd)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIME(msg_time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NSYS(msg_nsys)</td>
</tr>
</tbody>
</table>

**INIT**

Initializes and starts the communication link for the specified AF/REMOTE object. No abbreviations of the INIT keyword are acceptable.

**START**

Activates the AF/REMOTE communication link associated with the specified AF/REMOTE object name. No abbreviations of the START keyword are acceptable.

**STOP**

Inactivates the AF/REMOTE communication link associated with the specified AF/REMOTE object name. No abbreviations of the STOP keyword are acceptable.

**RESTART**

Re-starts the AF/Remote portion of the link. No abbreviations of the RESTART keyword are acceptable.

**SHUTDOWN**

Stops the communication link and shuts down the AF/REMOTE portion of the link. No abbreviations of the SHUTDOWN keyword are acceptable.
AF/REMOTE Manager Components

EXCEPT EXCEPT is used to temporarily except a link object from being managed on a system. This function will alter the in-storage copy of the AF/REMOTE Manager Control table to define the AF/REMOTE object with a KEYWORD of NONE. An excepted AF/REMOTE object will no longer be managed by the AF/REMOTE Manager. To re-enable AF/REMOTE Manager control of an excepted AF/REMOTE object, a 'AFR INIT object FORCE' command must be issued. Valid abbreviations are: EX and E.

RESET Resets the AF/REMOTE link re-start count. Valid abbreviations are: RES and R.

UPDATE Updates the AF/Operator link status table @AFRSTAT. Valid abbreviations are: UP and U.

LOGCHECK Checks the status of the AF/Remote log. No abbreviations of the LOGCHECK keyword are acceptable.

LOGSWITCH Shuts down & switches the AF/Remote logs. No abbreviations of the LOGSWITCH keyword are acceptable.

GETTIME Sets the AF/REMOTE PC date & time via ACTS. No abbreviations of the GETTIME keyword are acceptable.

SETDATE Sets the MVS system date via AF/REMOTE. No abbreviations of the SETDATE keyword are acceptable.

SETTIME Sets the MVS system time via AF/REMOTE. No abbreviations of the SETTIME keyword are acceptable.

DISPLAY Displays the requested AFR information via WTO to the MVS console (see Display Options below). The default display is the same as the STATUS display as described below. Valid abbreviations are: DIS and D.

STATUS Issues a WTO to the MVS console that indicates the AFR status for the specified resource. Valid abbreviation is: STAT.

SHOW Shows all AF/OPERATOR, AFR created, traps associated with a link. Valid abbreviations are: SH.

TBPUT AF/Remote Host service: Updates the status table with the specified value of the parameter used. No abbreviations of the TBPUT keyword are acceptable.
WTO
AF/Remote Host service: Issues a WTO on behalf of AF/REMOTE. No abbreviations of the WTO keyword are acceptable.

WTOR
AF/Remote Host service: Issues a WTOR on behalf of AF/REMOTE. No abbreviations of the WTOR keyword are acceptable.
Object Name
The AF/REMOTE object parameter is used to tell the AF/REMOTE Manager which AF/REMOTE object to manage. The aforementioned function will be performed for the AF/REMOTE objects specified with this parameter.

**ALL**
The ALL keyword is used to specify all links defined to the system. All links known to the AFR will be interrogated, those that have been defined to this system via the AF/REMOTE Control table will be chosen. An asterisk (*) is a synonym for the ALL keyword.

**CONS_ADDR**
AF/Remote MCS Console address.

**PATTERN**
An AF/Remote MCS Console address pattern, using a '*' as a wild card character. Any console with a name that begins with the letters preceding the asterisk (*) and is defined to the system will be managed.

**MSGID**
Used only with WTO and WTOR functions.

Options:
- **FORCE**
  Force a link to be managed.
- **MAINT**
  Manipulate only those links associated with a skeleton/maintenance system.
- **CHKP**
  AFR Checkpoint Restart.
- **CONSOLE=cons**
  Console to receive AFR messages. May be specified as console ID, console address, or the keywords LOG (AO Message Log) or MLG (System Log).

Display Options

**ACTIVITY**
Displays link activity statistics.

**CONNECTIVITY**
Displays AF/Remote connectivity information.

**SESSION**
Displays link session information.

**HARDWARE**
Displays link hardware console information.
Note: The TBPUT, WTO, and WTOR functions are for AF/Remote internal use only. These functions can not be invoked from a general MCS console. TBPUT options are for internal use only. WTO/WTOR options are for internal use only.

AR Command On-line Help

On-line help for the AR Manager AR command is available anytime. To display on-line help for the AR command, enter the following command at the MVS operator console:

AR HELP

The AR Manager will then display the AR command syntax followed by a brief explanation of each of the keywords associated with the command.
Introduction

The AF/OPERATOR Peer-to-Peer (P2P) facility performs direct communication in an SNA network. Through Peer-to-Peer, separate CPUs running AF/OPERATOR may communicate bidirectionally. This bidirectional communication provides AF/OPERATOR with added flexibility in monitoring and controlling the console activities of multiple CPUs regardless of their physical location.

Peer-to-Peer can be used to route system messages or console commands from one system to another. More importantly, messages from different systems may be consolidated onto one system. The one master system may then be used to manage events on various local or remote systems by responding to routed messages and issuing commands that will be executed on the remote systems.

Peer-to-Peer thus expands AF/OPERATOR's capabilities by providing the basis for console consolidation and unattended remote operations.

Peer-to-Peer is a VTAM LU 6.2 APPC interface. AF/OPERATOR interfaces with VTAM to define, initiate and control communication links or sessions between AF/OPERATOR hosts in an SNA network. Once the sessions are activated, an AF/OPERATOR system can communicate with other AF/OPERATOR hosts using commands and keywords that conform to the LU 6.2 communications protocol.

AF/OPERATOR must supply VTAM with information about the AF/OPERATOR hosts in the form of VTAM definitions and customization options. This information is specified by VTAM APPLID and MODETAB definitions. The names defined in the APPLID and MODETAB definitions are
referenced by AF/OPERATOR P2P commands that initialize and start Peer-to-Peer operations.

The Peer-to-Peer facility itself is comprised of two components: the communications interface, COM1, and the links or sessions that use the COM1 interface. The COM1 interface must be active on all AF/OPERATOR hosts that will have links established prior to activating any of the links. The P2P is used to automate the activation and management of the AF/OPERATOR COM1 interface and the Peer-to-Peer links to other AF/OPERATOR hosts.

Chapter Contents

Components of the P2P ................................................. 31
Using the Service Subroutines ...................................... 36
Automation Application Tables ...................................... 31
Service Subroutines .................................................. 31
Automation Table Component ....................................... 33
Introduction .............................................................. 33
Adding or Updating Links ............................................. 33
Control Options .......................................................... 34
Using the Service Subroutines ...................................... 36
Starting the Service Subroutines .................................... 36
Controlling Peer-to-Peer Manager Resources ..................... 36
P2P Command Syntax .................................................. 37
Functions ................................................................. 38
Stop Options .............................................................. 39
Link Options .............................................................. 39
SMFID ....................................................................... 40
Parms ...................................................................... 40
Controlling Peer-to-Peer Links ...................................... 41
P2P Command On-line Help ......................................... 41
Components of the P2P

The P2P has 3 components: Automation Application tables, service subroutines, and AF/OPERATOR system variables.

Automation Application Tables

The P2P tables hold information used to establish sessions or links between any two AF/OPERATOR hosts. All of the tables are members of the Automation Application table dataset, hilev.AFOPER.USERTBL. The tables are created by the user by filling in the table fields with the appropriate information for each system. The section, Automation Table Component beginning on page 33, describes the P2P automation tables in detail.

Service Subroutines

The P2P service subroutines are AF/OPERATOR command files used to perform a variety of functions for the links controlled by the P2P:

- Start / Stop the P2P COM1 interface
- Start / Stop the links
- Display the status of the links

The service subroutines are comprised of the following execs:

@P2PINIT This exec initializes the P2P environment and creates an MVS console command, called P2P, that allows you to perform various global management functions on a single link or group of links.

@P2PMAIN This exec processes P2P commands to control the links.

@P2PDRVR This exec processes P2P commands to control the links.

@P2PMAIN is responsible for: determining if a given link is defined to a system, responding to operator inquiries, starting/stopping the P2P COM1 interface and determining the status of the necessary VTAM nodes.

@P2PDRVR is responsible for: processing the P2P link define, link delete, link start and link stop commands and invoking the monitor facility.
Components of the P2P

@P2PHELP This exec provides the P2P on-line help facility.

@P2PMON This exec is responsible for monitoring all links that should be active.

@P2PDISP This exec processes the P2P Display command.

@P2PCOM1 This exec is an external REXX function used to determine if the AF/OPERATOR COM1 interface is active.

@P2PLNKA This exec is an external REXX function used to retrieve selected status information for a Peer-to-Peer link.

AF/OPERATOR System Variables

The P2P creates the following AF/OPERATOR system variables to track P2P status over time:

@P2PCOM1 P2P COM1 interface flag
@P2PxxAP P2P VTAM APPLID flag
@P2PxxCD VTAM host and remote CDRM flag
@P2PxxDF P2P link defined flag
@P2PxxST P2P link started flag
Automation Table Component

Introduction

To manage the communication links under its control, the P2P uses information stored in a table created by the user. The table that contains the communication link management information is \( @P2PCNTL \) The P2P Control Table.

This table is a member of the Automation Application table dataset. The table is loaded into above-the-line storage at AF/OPERATOR and Automation Application initialization time. Anytime a P2P link is accessed, the appropriate information is retrieved from the table and used to determine how to manage the link.

Adding or Updating Links

To add a link to the P2P, add a row to the P2P Control table and fill in all of the fields in the table with the appropriate information. The next time AF/OPERATOR is started, the new link's information will be available to the P2P and it will begin to manage the link. To update information for a link, edit the P2P Control table and change whatever fields are necessary. The next time AF/OPERATOR is started the changes will be available to the P2P. To implement the changes while AF/OPERATOR is running, use the Table Manager REFRESH command to refresh the in-storage copy of the P2P Control table. This gives the P2P access to the new data. Then you can issue a P2P LINK STOP command and a P2P LINK START command, specifying the link name, to update an existing link. To initialize a new link, only a P2P LINK START command, specifying the link name, need be issued.
Control Options

@P2PCNTL - P2P Link Control Table

The P2P Link Control table is used to store information on which links are defined to which systems. When a link is accessed, by operator command or internally, information is retrieved from this table and used to determine if the link is defined to the system.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMFID</td>
<td>Specifies the SMF ID of the system where the AF/OPERATOR task resides for the link that is being defined.</td>
</tr>
<tr>
<td>P2P_#</td>
<td>Specifies a unique number for the link starting from 1000. This is a four character numeric field used by the Peer-to-Peer Manager when creating AF/OPERATOR variables and traps associated with the link. Link numbers need not be consecutive and are not used in any ordering scheme, but each link must have a unique number assigned to it. Link numbers are automatically assigned and cannot be modified.</td>
</tr>
<tr>
<td>P2P_ID</td>
<td>This field is used to identify the name of the link. Each Peer-to-Peer link must have a unique name or link ID. This is an eight character alphanumeric field, which must start with an alphabetic character. The name specified in this field must match the value specified for the LINKID(xxx) parameter in the OGPARMS member for the AF/OPERATOR task.</td>
</tr>
<tr>
<td>P2P_APPL</td>
<td>This field is used to identify the VTAM node name that defines the AF/OPERATOR Peer-to-Peer interface to VTAM for this link. This is an eight character alphanumeric field.</td>
</tr>
<tr>
<td>P2P_MODE</td>
<td>This field is used to identify the VTAM log mode table that contains the session parameters and characteristics for this link.</td>
</tr>
<tr>
<td>P2P_APPL</td>
<td>Specifies the VTAM application node name for the Peer-to-Peer facility for this system.</td>
</tr>
<tr>
<td>P2P_CMDRET</td>
<td>Specifies the maximum number of times the Peer-to-Peer Manager will attempt to define or start a link, in the form of a whole number. The Peer-to-Peer Manager will wait for the time period specified by the Command Interval field between each retry. The default number of retries is 10.</td>
</tr>
</tbody>
</table>
Automation Table Component

P2P_CMDINT  This field is used to specify the verification interval for Peer-to-Peer commands. When a command is issued to define or start a link, the Peer-to-Peer Manager will wait for the time limit specified here for verification that the command was successful. Enter the command verification interval in military time format. The default is 30 seconds, 00:00:30.

P2P_MONINT  This field is used to specify the time interval for monitoring the status of the link. At the end of each interval, a message is sent across the link to test connectivity and ensure that the link is active. Enter the monitoring interval in military time format. The default is 15 minutes, 00:15:00.

P2P_DESC  The Description Field is used to specify a one line, ‘plain English’ description of the resource for the user's reference. This information is displayed on the Peer-to-Peer Manager Link Selection panel.

P2P_ID  Specifies the link ID for this system. The link ID must be the same as the LSYSNAME specification in the AF/OPERATOR AFPARMxx member.
Using the Service Subroutines

Introduction
Once the automation table component of the P2P has been defined, the information needed to manage the links for an environment is available to the P2P service subroutines. To initiate the actual management of the links, the P2P service subroutines must be activated.

Starting the Service Subroutines
The P2P is automatically started at AF/OPERATOR initialization time by the Application Manager which executes the @P2PINIT exec. This command file initializes the P2P environment and sets the command traps for the P2P operator command. The STM executes @P2PMAIN to initialize the Peer-to-Peer COM1 interface and execute the @P2PDRVR exec to start the links that should be active for that system. The @P2PDRVR exec is executed asynchronously, via a TOD(*) trap, to start each defined link and set the trap to start the monitor facility for that link.

If AF/OPERATOR is already running, you can activate the P2P before the next AF/OPERATOR recycle by issuing the following commands from the MVS console:

```
APL INIT P2P
RCF START P2P FORCE
```

Note: This assumes that the P2P and STM tables have been pre-loaded into storage by the Table Manager. If the tables have not been automatically loaded, issue the appropriate Table Manager commands, from the MVS console, to load the tables before you initialize the P2P.

Controlling Peer-to-Peer Manager Resources
The P2P command provides an operator interface to give you additional control over the links defined to the P2P. It is used to communicate with AF/OPERATOR and the P2P service subroutines to control P2P links and display P2P information about the links from the MVS console.
For example, from the MVS operator console you can enter:

**P2P DISPLAY smfid**

at any time, to display the status of the link between this system and the system whose SMFID was specified in the above command.

An asterisk (*) is treated as a wild card character, which allows you to specify groups of like-named links. For example, the command:

**P2P DISPLAY S***

will display the status of all links that begin with the letter S.

**P2P Command Syntax**

**FIGURE 2. P2P Command Syntax**

<table>
<thead>
<tr>
<th>P2P function</th>
<th>option</th>
<th>smfid</th>
<th>parms</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOP</td>
<td>FORCE</td>
<td></td>
<td>TIMEOUT(nn)</td>
</tr>
<tr>
<td></td>
<td>IMMEDIATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINK</td>
<td>DEFINE</td>
<td>ALL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DELETE</td>
<td>smfid</td>
<td></td>
</tr>
<tr>
<td></td>
<td>START</td>
<td>pattern</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STOP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISPLAY</td>
<td>ALL</td>
<td>smfid</td>
<td>ACTIVE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pattern</td>
<td>INACTIVE</td>
</tr>
</tbody>
</table>
Functions

The function parameter is used to tell the P2P which action to take against a link or the Peer-to-Peer COM1 interface. Many, but not all, functions have abbreviations. Acceptable abbreviations are noted with the function’s description.

**START**

Activates the Peer-to-Peer COM1 interface. No abbreviations of the START keyword are acceptable.

**STOP**

Deactivates the Peer-to-Peer COM1 interface. When issued, if any received WTORs have not been replied to, the matches executing those WTORs will be cancelled. No abbreviations of the STOP keyword are acceptable.

**Note:** A P2P STOP command will be automatically issued anytime communication terminates due to network problems.

**LINK**

Defines, activates, terminates and deletes communication links among AF/OPERATOR hosts in a Peer-to-Peer network. Valid abbreviations are: L.

**DISPLAY**

Shows the current link status of the requested Peer-to-Peer link. Valid abbreviations are: DIS or D.

A link can be in one of the following states:

- **ACTIVE** - The link between this system and the system specified in the command is active. Commands and/or messages are being sent across the link.

- **INACTIVE** - The link between this system and the system specified in the command is inactive. Commands and/or messages are not being sent across the link. A P2P LINK START command should be issued to activate the link.

- **DEFINED** - The link between this system and the system specified in the command is defined. Commands and/or messages are not being sent across the link. A P2P LINK DEFINE command should be issued to activate the link.

- **UNDEFINED** - The link between this system and the system specified in the command is undefined. Commands and/or messages are not being sent across the link. A P2P LINK DEFINE command should be issued to define and activate the link.
Stop Options
The Stop options are used in conjunction with the P2P STOP command.

FORCE  Deactivates the Peer-to-Peer interface and any AF/OPERATOR operations using the interface. Without the FORCE option, P2P STOP deactivates the Peer-to-Peer interface but does not deactivate any AF/OPERATOR operations using the interface. Use when shutdown of AF/OPERATOR is prevented while link activities are in progress. No abbreviations of the FORCE option are acceptable.

IMMEDIATE Terminates the Peer-to-Peer interface after all active messages are sent and before new messages arrive. No abbreviations of the IMMEDIATE option are acceptable.

Note: Unless otherwise specified, the default option is IMMEDIATE.

Link Options
The Link options are used in conjunction with the P2P LINK command.

DEFINE  Defines a communication link between AF/OPERATOR systems through the Peer-to-Peer interface. Valid abbreviations are: DEF.

DELETE Deletes a communication link between AF/OPERATOR systems through the Peer-to-Peer interface. Valid abbreviations are: DEL.

START  Activates a Peer-to-Peer communication link or links. No abbreviations of the START option are acceptable.

STOP  Deactivates a Peer-to-Peer communication link or links. No abbreviations of the STOP option are acceptable.
Using the Service Subroutines

**SMFID**

The SMFID parameter is used to tell the P2P which link to manage.

**ALL**

The ALL keyword is used to specify all links defined to the system. All links known to the P2P will be interrogated, those that have been defined to this system via the Peer-to-Peer Control table will be chosen. A '*' is a synonym for the ALL keyword.

**SMFID**

An SMFID precisely as coded in the Peer-to-Peer Control table.

**PATTERN**

An SMFID pattern, using a '*' as a wild card character. Any link with a name that begins with the letters preceding the asterisk (*) and is defined to the system will be managed.

**Parms**

TheParms parameter is used to alter the normal P2P processing of the P2P STOP and DISPLAY functions.

**ACTIVE**

Displays only active links.

**INACTIVE**

Displays only inactive links.

**TIMEOUT(nn)**

The TIMEOUT parameter lets you specify a time limit (in seconds) for a P2P immediate stop command. If P2P does not terminate within the specified time frame, outstanding WTO and command traffic will be cancelled regardless of the work in progress.

*Note:* Unless otherwise specified, the default option is TIMEOUT(30).
Controlling Peer-to-Peer Links

The Peer-to-Peer interface is comprised of two components: the communications interface, COM1, and the links that use the COM1 interface for host-to-host communication. The COM1 interface must be started on all systems within the AF/OPERATOR Peer-to-Peer network prior to starting a link between any of the systems in the network. Once the COM1 interface has been started on all systems, the communication links between any two systems can be started. Each logical link, a link between any host and remote system, must be enabled on both the host and remote systems.

Should the link not be enabled on either the host or remote system, communication between the two systems will fail. In certain situations, it will be necessary to stop/start the Peer-to-Peer interface. Some of the likely situations would be: a system being removed from or added to the AF/OPERATOR Peer-to-Peer network or one of the systems within the network experiencing VTAM problems. In any case, all links to other systems from the removed, failing or additional system should be started or stopped.

The P2P commands that are necessary to accomplish the start or stop of the COM1 interface and the links are:

- **RCF START P2P**: This command will cause the STM to start and monitor the P2P interface.
- **P2P START**: This command will start the P2P interface external to the STM.
- **RCF STOP P2P**: This command will cause the STM to stop the P2P interface on the host system and all remote links connected to the host system.
- **P2P STOP**: This command will stop the P2P interface external to the STM.

**P2P Command On-line Help**

On-line help for the P2P Manager P2P command is available anytime. To display on-line help for the P2P command, enter the following command at the MVS operator console:

```
P2P HELP
```

The P2P Manager will then display the P2P command syntax followed by a brief explanation of each of the keywords associated with the command.
Using the Service Subroutines
Introduction

The AF/OPERATOR OMEGAMON™ interface is a component of AF/OPERATOR and uses the same virtual terminal pool as POVI. The AF/OPERATOR OMEGAMON interface allows you to gather a wide range of performance data on a system and send it to other applications or automation products. The two principal methods of using the interface is (1) trap OMEGAMON exceptions and (2) issue OMEGAMON commands. Exception analysis is an OMEGAMON feature that monitors predefined thresholds in a system. When a threshold is exceeded, an event called an exception, OMEGAMON displays the exception on the OMEGAMON console. A trap can be written that queries OMEGAMON at regular intervals for an exception and then takes actions when the trap executes. Example: trap an OMEGAMON CICS PAGE exception (page-in too high) and then build automation routines that reduce the system workload when the trap executes. The OMEGAMON Logon Manager is used to define in a structured table format, all of the logon definitions that are required by the AF/OPERATOR OMEGAMON interface. By using IRM to store and maintain the logon definitions it allows for the logon and logoff commands to be executed automatically based on the availability of the OMEGAMON address space in a pred/succ type relationship.

Chapter Contents

OMEGAMON Logon Manager Components ........................................... 44
Automation Application Table ......................................................... 44
Service Subroutines ................................................................. 46
Automation Application Table

@OM2CNTL - OMEGAMON Logon Control table

The OMEGAMON Logon Control table is used to store information on requirements for AF/OPERATOR logons to OMEGAMON and OMEGAVIEW tasks.

@OM2NAME

This field is used to specify the Logon Name field contains the name of the OMEGAMON Logon Manager resource. The name is defined when the resource is first added to the OMEGAMON Logon Manager.

The Logon Name is specified as follows:

- OMMVS An OMEGAMON MVS Logon
- OMCIC An OMEGAMON CICS Logon
- OMIMS The first OMEGAMON IMS Logon
- OMIMx Subsequent OMEGAMON IMS Logons (where x = 2,3, etc.)
- OMDB2 An OMEGAMON DB2 Logon
- OMVIEW An OmegaView Logon

AOSMFID

This field is used to specify the System and allows the user to define a unique SMFID where the logon is to be managed. This field is required.

@OM2SID

This field is used to specify the Link ID used to communicate with the system where the target OMEGAMON or OMEGAVIEW (the task coded in the OMEGAMON Task Name field) resides. The default is the local system link ID.

@OM2TASK

This field is used to specify the started task name of the target OMEGAMON with which the logon is to be established. This is a required field.

@OM2APL

This field is used to identify the VTAM node name of the target OMEGAMON. This is an 8 character alphanumeric field. This field is required.
**OMEGAMON Logon Manager Components**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@OM2APID</td>
<td>This field is used to specify a unique user defined name to be used in subsequent AF/OPERATOR commands to refer to the logon, such as the SHOW LOGONS command. This field is required.</td>
</tr>
<tr>
<td>@OM2UMX</td>
<td>This field is used to specify the maximum number of users who may log on concurrently to the target OMEGAMON. The default value is 10. Valid values are: 1-99.</td>
</tr>
<tr>
<td>@OM2USR</td>
<td>This field is used to specify the User ID that AF/OPERATOR has been assigned to logon to the specified OMEGAMON. This is a required field.</td>
</tr>
<tr>
<td>@OM2PSWD</td>
<td>This field is used to specify the password associated with the logon user ID. This field is not displayed. To enter or change the password, tab to this field and type in the password. The Password Verification panel will then be displayed to verify the password.</td>
</tr>
<tr>
<td>@OM2INT</td>
<td>This field is used to specify the verification interval for the Logon command. When the command is issued to initiate a logon, the Started Task Manager will wait for the time limit specified by this field for verification that the command was successful. Code the verification interval in military time format. The default is 5 minutes, 00:05:00.</td>
</tr>
<tr>
<td>@OM2LGNTASK</td>
<td>This field is used to Specify the logon task name that is defined to the Started Task Manager for this resource. This is a required field.</td>
</tr>
<tr>
<td>@OM2_DESCR</td>
<td>This field is used to specify the Description is a one line plain-English description of the logon defined by the user. The description is entered when the logon is first added to the OMEGAMON Logon Manager.</td>
</tr>
</tbody>
</table>
Service Subroutines

The OMEGAMON Logon Manager service subroutines are AF/OPERATOR command files which perform the services needed to manage the logons to the OMEGAMON and OMEGAVIEW tasks. The service subroutines are comprised of the following execs:

@OM2LOGF This exec is called from the @RCFDRVR exec and is used to set global variables prior to AF/OPERATOR logoff command for OMEGAMON MVS.

@OM2LOGN This exec is called from the @RCFDRVR exec and is used to set global variables prior to AF/OPERATOR logon command for OMEGAMON MVS.

@OM2VRA1 This exec is called from the @VTMMAIN exec and is used to issue a modify command to reconnect OMIIIEPZM after a VTAM failure.

@OM2XRBL This exec is called as a Pre-Recovery Exit for OMEGAMON MVS logons. It will issue a LOGOFF for the application prior to the recovery attempt to clear the old LOGON.

@OM2XSA1 This exec is called as a post-start exit for the OMIIIMANT job. It will start command for OMIIHDI and OMIIIEPZM tasks after successful completion of OMIIIMANT backup job.

@OM2XSB1 This exec is called as a pre-start exit for the OMIIIMANT job. It will stop OMIIHDI and OMIIIEPZM tasks prior to running OMIIIMANT backup job.
Starting the Service Subroutines

The OMEGAMON Logon Manager is automatically initialized at AF/OPERATOR initialization time by the Application Manager which will instruct the Table Manager to load the @OV2CNTL Control Table into above-line-storage.

Using the Service Subroutines

The automation available with the OMEGAMON Logon Manager is accessed via user exit points. Once all the required parameters are defined in the @OV2CNTL Control Table the next step is to define the required LOGON resource definition in the Started Task Manager to use the services of the OMEGAMON Logon Manager.

From the Started Task Manager main menu, select from the Action pull-down menu, option 9 Copy by Example. Locate and select, for this example a logon resource definition for OMEGAMON MVS, LOGONOMMVS, the name you select for the resource must match the name specified in the parameter @OM2TASK in the @OM2CNTL table. The predecessor list should at the very least include VTAM.

- Pre-Start Exit @OM2LOGN
- Pre-Recovery Exit @OM2XRBL
- Pre-Stop Exit @OM2LOGF

Save your definition and proceed to option 12.7 from the IRM ISPF interface main menu. From the Omegamon Lgn Manager Logon Definition panel enter ADD to create a new definition. For this example we will be defining a OMMVS definition. From the new definition panel enter the required values in the following fields:
Logon Name

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTAM Appl ID</td>
<td>System</td>
</tr>
<tr>
<td>Omegamon Task Name</td>
<td>Maximum Users</td>
</tr>
<tr>
<td>Application Name</td>
<td>Link ID</td>
</tr>
<tr>
<td>Task Name</td>
<td>Interval</td>
</tr>
<tr>
<td>User ID</td>
<td>Password</td>
</tr>
</tbody>
</table>

Refer to the @OM2CNTL Control table field descriptions earlier in the chapter for possible values to all of these fields. Once all values have been entered save your entry and exit the Interface.

To implement the changes while AF/OPERATOR is running, use the Table Manager REFRESH command to refresh the in-storage copy of the Started Task Manager and OMEGAMON Logon Manager tables. This provides the OMEGAMON Logon Manager access to the new data.
**Introduction**

POVI is an AF/OPERATOR automation interface with VTAM 3270 applications. POVI lets you automate interactions with VTAM 3270 applications and provides two methods:

- **POVI REXX functions**: POVI can be instructed to capture data by parsing an application's screen image and to insert data and variables into specific screen fields or cursor locations. After a REXX exec has been created using POVI REXX functions, you can instruct AF/OPERATOR to execute it automatically by a trap. AF/OPERATOR uses the POVI Playback Manager during REXX exec execution to accomplish your task.

- **POVI's Keystroke Recorder**: create scripts, sequences of keystrokes and variables. POVI can record everything you type as you "run through" a task and save this information for playback. After the script has been created you then need to create a REXX exec to execute it. When triggered by a trap, AF/OPERATOR executes the REXX exec that plays back your script and uses the Playback Manager to take the desired actions.

The POVI Session Manager is used to define in a structured table format, all of the session definitions that are required by the AF/OPERATOR POVI interface. By using IRM to store and maintain the session definitions it allows for the logon and logoff commands to be executed automatically.
Chapter Contents

AF/OPERATOR Manager Components ........................................ 51
Automation Application Table ............................................. 51
Automation Table .............................................................. 52
@OVICNTL - POVI Session Manager Control Table .................... 52
Service Subroutines ............................................................ 53
Overview ............................................................................. 53
Starting the Service Subroutines .......................................... 53
AF/OPERATOR Manager Components

Automation Application Table

To manage the POVI logons the POVI Session Manager uses information stored in a table created by the user. The table that contains the POVI Session information is:

@OVICNTL The POVI Session Manager Control Table

This table is a member of the Automation Application table dataset. The table is loaded into above-the-line storage at AF/OPERATOR and Automation Application initialization time. Any time AF/OPERATOR is started, the new POVI Session information will be available to the POVI Session Manager and will begin to manage the POVI logons.

To implement changes while AF/OPERATOR is running, use the Table Manager REFRESH command to refresh the in-storage copy of the POVI Session Manager Control Table. This gives the POVI Session Manager access to the new data.

Service Subroutines

The POVI Session Manager service subroutines are AF/OPERATOR command files that perform the services needed for the POVI Session logons. The majority of the POVI subroutines are used by the CICS Manager and are documented in the chapter on the CICS Manager. However, there are additional subroutines for the POVI Session Manager which are documented here:

@OVIVRA1 This exec is called from the @VTMMAIN exec and is used to recover POVI resources after VTAM node recovery.

@OVIXPW1 This exec is used as a stop WTOR exec and will be called if AFPOVI does not come down with the first stop command, issue another command to bring it down. If second stop works, bypass the STOP_VER WTOR.

@OVIXRB1 This exec is used as a pre-recovery used if VTAM task fails, stop and re-start AFPOVI.
@OVICNTL - POVI Session Manager Control Table

The POVI Session Manager Control Table is used to store information on each system's POVI Session logon parameters under the control of the POVI Session Manager. The POVI session logon automation is invoked when the Started Task Manager either starts, stops or attempts recovery a LOG@cicsname that has been defined to the POVI Session Manager.

- **POVI_TASK** Specifies the started task name of the target POVI task with which the session is to be established. This is a required field.

- **POVI_APPL** This field is used to identify the node name of the AF/OPERATOR POVI Session Interface as defined to VTAM. This is an 8 character alphanumeric field. This is a required field.

- **POVI_PFX** This field is used to specify the prefix of the logical unit names for the virtual terminal pool as defined to VTAM. This is a required field.

- **POVI_INT** This field is used to specify the verification interval for POVI Session commands. When a command is issued to define or start a session, the POVI Session Manager will wait for the time limit specified by this field for verification that the command was successful. Code the verification interval in military time format. The default is 1 minute, 00:01:00.

- **POVI_RET** Specifies the maximum number of times the POVI Session Manager will attempt to define or start a session, in the form of a whole number. The POVI Session Manager will wait for the time period specified by the Interval field between each retry. Valid values are: 1-999. The default number of retries is 4.

- **POVI_DESCR** The Session Description is a one line, 'plain-English' description of the session, defined by the user. The description is entered when the session is first added to the POVI Session Manager.
Service Subroutines

Overview

Once the automation table component of the POVI Session Manager has been defined, the information needed to manage the POVI Session logons is available to the POVI Session Manager service subroutines. To initiate the actual management of the POVI Session logons, the POVI Session Manager service routines must be activated.

Starting the Service Subroutines

The POVI Session Manager is automatically initialized at AF/OPERATOR initialization time by the Application Manager which will instruct the Table Manager to load the @OVICNTL Control Table into above-line-storage.

Using the Service Subroutines

The automation available with the POVI Session Manager is accessed via user exit points. Once all the required parameters are defined in the @OVICNTL Control Table the next step is to define the required Started task definition for each of the CICS regions in the Started Task Manager to use the services of the POVI Session Manager.

From the Started Task Manager main menu, use the 'copy by example' option and create a new definition for each of the desired CICS regions. Select any one task definition from the list of tasks that start with LOG@ and change each occurrence of the task name with the CICS name you wish to automate and save your entry.

To implement the changes while AF/OPERATOR is running, use the Table Manager REFRESH command to refresh the in-storage copy of the Started Task Manager and POVI Session Manager tables. This gives the POVI Session Manager access to the new data.
Communication Manager

Introduction

The Communication Manager is a facility used to define the parameters that are necessary for establishing the AF/Operator communication environment. Supported communication protocols for the connections are: APPC, TCP/IP and XCF. Once the Communication Manager definitions are created, Started Task Manager resource definitions can be created to manage the links and servers that comprise the communication environment.

Communication links allow AF/Operator to communicate with the following:

- A remote AF/OPERATOR
- AF/REMOTE OS/2
- AF/REMOTE NT

Peer-to-peer communication between AF/Operator systems can be established using APPC, XCF or TCP/IP protocols. An XCF connection can be established between two AF/Operator systems in a Sysplex before VTAM is available, or even if VTAM is unavailable. The APPC and TCP/IP protocols are both VTAM based connections.

Before proceeding further, the user should read the following sections in the AF/Operator Command Reference Manual, Version 320:

- Chapter 6 - Understanding the Communications Environment
- Chapter 7 - AF/OPERATOR Commands, COMM and LINK
Chapter Contents

Communication Manager Components ........................................ 57
  ISPF User Interface ..................................................... 57
  XCF Parameters ......................................................... 60
  TCP/IP Parameters ..................................................... 61
  APPC Parameters ....................................................... 62
  REXX Tables ............................................................. 64
    Overview .............................................................. 64
  Runtime Environment .................................................. 65
    Overview .............................................................. 65
    Adding or Updating Resources .................................... 65
    Controlling Communication Manager Resources ................. 65
  COM Command Syntax ................................................ 66
Communication Manager Components

The Communications Manager has several components: the ISPF User Interface, Automation Application Tables, Service Subroutines, and a Command Interface for controlling and managing the resources.

ISPF User Interface

Overview

Link and server resources are defined to the Communication Manager through the ISPF User Interface. Link and server definitions are stored in the Communication Control Table, @COMCNTL, which is a member of the table dataset, hilev.RKAUTBLR.

Adding or Updating Resources

Resources are added to the Communication Manager through the IRM ISPF User Interface. From the main panel of the IRM user interface, select option 12, AF/Operator Resources. Next, select option 9, Communication Manager Resource Definitions.

An existing resource definition may be updated by performing the following actions: position the cursor to the selection field of the desired resource, enter 'U', then press the Enter key. The Link Definition panel for the desired resource will be displayed.

A new resource definition may be added by one of the following methods:

- At the Command line prompt: enter 'ADD', then press the Enter key
- In the selection field of a resource definition, enter 'A' to add a new definition or 'C' to copy an existing definition, then press the Enter key
- From the Actions Pulldown Menu, select option 1, Add an item

The Link Add or Link Copy panel will be displayed.
Link Definition Panel: Field Descriptions

The following section describes the fields on the Link Add, Link Copy and Link Definition panels. REXX table name and column names corresponding to each field are included for reference.

**LinkID**

The LinkID field is used to identify the target application for this communication session and uniquely identifies the session. The linkid field can be a maximum of 8 characters. This is a required field.

*Note:* Linkid cannot start with the special character ‘#’. AF/OPERATOR reserves linkids beginning with a # in the range #0000001 to #9999999 for internal use only.

REXX table name is @COMCNTL; column name is LINKID.

**SMF ID**

The SMF ID Field is used to specify the system where the server will be started. Links to this server can be established from other systems.

This is a four character field, which may contain any EBCDIC character with the exception of an asterisk (*) or quotes (’, ’) and which must not begin with one of the Rexx Table special characters (+%/-). This is a required field.

REXX table name is @COMCNTL; column name is SMFID.

**Description**

The Description field is used to specify a one line, 'plain English' description of the resource for the user's reference. This information is displayed on the Communication Manager Link Selection panel. This is a required field.

REXX table name is @COMCNTL; column name is DESCRIPTION.
Communication Manager Components

**Protocol**

The Protocol field is used to specify the type of communication protocol for this session. Valid values are:

- XCF
- TCPIP
- APPC

This is a required field. An XCF connection is recommended for peer-to-peer communication within a Sysplex.

REXX table name is @COMCNTL; column name is PROTOCOL.

**Number**

The Number field is used to specify a unique number for the link starting from 1000. This is a four character numeric field used by the Communication Manager when creating AF/Operator variables and traps associated with the link.

Link numbers need not be consecutive and are not used in any ordering scheme, but each link must have a unique number assigned to it. Link numbers are automatically assigned and cannot be modified.

REXX table name is @COMCNTL; column name is NUMBER.

**NOID**

The NOID field is used to turn off AF/OPERATOR internal ID request processing. Specify NOID if you are defining a session to a distributed application, such as AF/REMOTE. This prevents an internal AF/OPERATOR record from flowing as part of the stream of outbound data.

This is a binary field (0,1); specify 1 to enable NOID, that is, to turn off AF/Operator internal ID request processing. Specifying 0 turns on internal ID request processing. This is an optional field. If not specified, the default is 0.

Note: At least one link must be defined with the NOID parameter disabled, or set to 0, to exploit IRM Sysplex functionality. It is recommended to define the link with a protocol of XCF.

Note: When an XCF link is defined with the NOID parameter enabled, or set to 1, AF/Operator will not issue error messages. IRM recovery processing will not function properly when this parameter is enabled.
**Communication Manager Components**

**No Heartbeat Check**  
The No Heartbeat Check field is used to enable or disable AF/OPERATOR to AF/OPERATOR heartbeat checking. Specify 0 if you are defining a session to a distributed application, such as AF/REMOTE.

This is a binary field (0,1); 1 specifies that heartbeat checking will be disabled. 0 specifies that heartbeat checking will be enabled. This is an optional field. If not specified, the default is 0.

REXX table name is @COMCNTL; column name is NOHRTBT.

**Drain**  
The Drain field is used to specify if queued requests should be allowed to complete before terminating the link.

This is a binary field (0,1); 1 specifies that queued requests will be allowed to complete before the link is terminated. Specifying 0 will immediately terminate the link; queued requests will not be allowed to complete. This is an optional field. If not specified, the default is 0.

REXX table name is @COMCNTL; column name is DRAIN.

**XCF Parameters**

**Group**  
The Group field is used to specify the XCF group name. The group name identifies with which group a session will be established.

This is a required field. The group field can be a maximum of 8 characters in length. Refer to the IBM Sysplex manuals for XCF Group naming conventions.

REXX table name is @COMCNTL; column name is GROUP.
Communication Manager Components

**Member**

The Member field is used to specify the XCF member name. The member name, along with group name, identifies the particular server with which communication will be established. Server names must be unique within an XCF Group.

This is a required field. The member field can be a maximum of 8 characters in length. Refer to the IBM Sysplex manuals for XCF Member naming conventions.  
REXX table name is @COMCNTL; column name is MEMBER.

**TCP/IP Parameters**

**Hostname**

The Hostname field is used to specify the IP address of the remote host you wish to establish a session with. Hostname must be used in conjunction with the Port Number field.

This is a required field. Hostname can be a maximum of 24 characters in length. It can be a dotted decimal IP address or a hostname that is defined to your network's domain name server. If hostname is a dotted decimal IP address, none of the four numeric levels of the address can exceed 255.  
REXX table name is @COMCNTL; column name is HOSTNAME.

**Port Number**

The Port Number field is used to specify the listening port of the remote host you wish to establish a session with. Port number must be used in conjunction with the hostname field.

This is a required field. The port number field can be a maximum of 5 numeric characters and must be in the range of 1025 to 32767.  
REXX table name is @COMCNTL; column name is PORT_#.
**Recovery Wait**  
The Recovery Wait field is used to specify the amount of time the Communication Manager will wait before it initiates recovery for a TCP/IP server. The wait interval must be long enough to account for the TCP/IP Linger delay that is caused by the previous close of the server. Specifying a wait interval that is too short could result in a bind error during the server recovery process.

Specify the recovery wait interval in military time format. This is an optional field. If not specified, the default is 2 minutes, 00:02:00.

REXX table name is @COMCNTL; column name is TCP_WAIT.

**ASCII**  
This field is used to specify the type of text to be transferred for the negotiation phase of a TCP/IP connection. AF/OPERATOR will initiate the negotiation by sending an ID record. This option can be used to ensure that a distributed system will not have to handle EBCDIC data, thereby simplifying the process on that system.

This is a binary field (0,1); 1 specifies that ASCII text will be used 0 specifies that EBCDIC data will be used for initial contact. This is an optional field. If not specified, the default is 0.

*Note:* This parameter affects the first ID record sent from AF/OPERATOR during link startup. It does not necessarily determine the outcome of the negotiation process. The distributed system can still negotiate for the use of EBCDIC data.

REXX table name is @COMCNTL; column name is ASCII.

**APPC Parameters**

**TP Name**  
This field is used to identify the application to establish a session with by its APPC transaction program name.

This is a required field. TP name can be a maximum of 64 characters in length.

REXX table name is @COMCNTL; column name is TPNAME.
LLU Name

This field is used to specify the logical unit name that identifies the server. This is the NOSCHED type LU 6.2 applid defined to APPC/MVS and VTAM.

This is a required field. LLU name can be a maximum of 8 characters in length.

REXX table name is @COMCNTL; column name is LLUNAME.

PLU Name

This field is used to specify the APPC partner LU (logical unit) name for the application with which a session is to be established. This is the NOSCHED type LU 6.2 applid defined to APPC/MVS and VTAM.

This field is optional. If specified, PLU name can be a maximum of 17 characters in length.

REXX table name is @COMCNTL; column name is PLUNAME.

Symbolic Destination

This field is used to specify the APPC symbolic destination name for the application with which a session is to be established.

This field is optional. If specified, the symbolic destination name can be a maximum of 8 characters in length.

REXX table name is @COMCNTL; column name is SYMDEST.

Modename

This field is used to specify the VTAM log mode table entry definition used for VTAM LU 6.2 sessions. The log mode table contains the default session parameters and characteristics for this session.

This field is optional. If specified, the modename can be a maximum of 8 characters in length. If not specified, it defaults to the log mode in effect for the partner LU.

REXX table name is @COMCNTL; column name is MODENAME.
REXX Tables

Overview

Resource definitions that are created or updated through the Communication Manager ISPF User Interface are stored in REXX tables. The Communication Manager uses the information contained within these tables to manage its resources. The tables that contain this information for the Communication Manager are:

- @COMCNTL Communication Manager Control Table

Each table is a member of the table dataset, hilev.RKAUTBLR. At AF/Operator and IRM application initialization time, these tables are loaded from DASD into above-the-line storage. The in-storage tables contain the data that is used to manage the Communication Manager resources in the runtime environment.
Runtime Environment

Overview

The Communication Manager is enabled and initialized during AF/OPERATOR and IRM application initialization. During the application initialization process, application REXX tables and execs are loaded into storage, an operator interface is enabled, and application resources defined to the system are started.

Adding or Updating Resources

Through the ISPF user interface, add or update the desired Communication Manager resource. The updated definition is now contained in the REXXX table, @COMCNTL. To reflect this change in the runtime environment, the in-storage table must be updated. If this table is not updated, the change will not be available in the runtime environment. The in-storage table can be updated by one of the following methods:

- Recycle the AF/Operator started task. This will cause the updated REXXX table to be loaded into storage.

- To implement the changes while AF/Operator is running, use the Table Manager REFRESH command to refresh the in-storage table, i.e. to effect changes to the @COMCNTL table, issue the following command:

  TBL REFRESH @COMCNTL

Controlling Communication Manager Resources

The Communication Manager operator command is defined in the Command Prefix Field in the System Definitions selection. The default command prefix is COM.

The COM command provides an operator interface to give you additional control over the resources defined to the Communication Manager. It is used to communicate with AF/OPERATOR and to control and display information about the Communication Manager resources from the MVS console.

For example, from an MVS operator console you can enter:

COM DISPLAY LINK * TYPE=XCF

at any time, to display current status of all defined XCF links.
COM Command Syntax

FIGURE 3. COM Manager Syntax

<table>
<thead>
<tr>
<th>COM function</th>
<th>objects</th>
<th>resource</th>
<th>options</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>SERVER</td>
<td>resource</td>
<td>TYPE=type</td>
</tr>
<tr>
<td></td>
<td>LINK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HELP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Functions**

The function parameter, required, is used to tell the Communication Manager which action to take against a link or server. Many, but not all, functions have abbreviations. Acceptable abbreviations are noted with the function’s description.

- **DISPLAY** Display Link or Server status. Valid abbreviations are: DIS or D.
- **HELP** Displays the Communication Manager on-line help file on the operator console. No abbreviations of the HELP keyword are acceptable.

**Objects**

The objects parameter, required, is used to tell the Communication Manager the type of resource definition to manage. Many, but not all, objects have abbreviations. Acceptable abbreviations are noted with the objects’ description.

- **SERVER** Keyword to display server status. Valid abbreviations are: S.
- **LINK** Keyword to display link status. Valid abbreviations are: L.
COM Command Syntax

Resource

The resource parameter, required, is used to tell the Communication Manager which resources to manage. The aforementioned function will be performed for the resources specified with this parameter.

- **ALL**
  The ALL keyword is used to specify all resources defined to the system. All resources known to the Communication Manager will be interrogated, those that have been defined to this system via the Communication Control table will be chosen. An asterisk (*) is a synonym for the ALL keyword.

- **PATTERN**
  A resource name pattern, using an asterisk (*) as a wild card character. Any resource with a name that begins with the letters preceding the asterisk (*) and is defined to the system will be managed.

Options

The options, optional, are used to limit the Display output to a specified protocol.

- **TYPE=XCF**
  The TYPE=XCF keyword is used to limit the function to resources defined with a protocol of XCF.

- **TYPE=TCPIP**
  The TYPE=TCPIP keyword is used to limit the function to resources defined with a protocol of TCPIP.

- **TYPE=APPC**
  The TYPE=APPC keyword is used to limit the function to resources defined with a protocol of APPC.

- **TYPE=ALL or ***
  The TYPE=ALL or * keyword is used to display all servers or links, regardless of the defined protocol. If the TYPE= option is not specified, it defaults to ALL.
COM Command Syntax
Introduction

The JES Manager allows you to manage JES3 and all of its related resources, such as job classes, job groups, writers, JES lines, NJE nodes. The JES Manager can be used to automatically start and stop JES3 and any of its resources needed by Started Tasks or other JES applications running on your system. The JES Manager can be used to:

- Activate a JES resource such as a job class or job group before a Started Task is started.
- Activate a JES resource as an individual object.
- Activate all JES resources defined to a system when JES3 is started.

The JES Manager allows you to define default JES3 system specific startup parameters. The flexibility of the JES Manager object definitions allow you to manage many individual JES3 resources as a single entity. Each JES object can then be grouped with other similar objects or managed by itself. The JES Manager is initialized in a two step process. The basic JES Manager environment is created at AF/OPERATOR initialization time. The actual management of JES3 resources is enabled once JES3 itself has been started.

Note: The JES3 Job Class and JES3 Job Group Managers are separately billable items and can only be activated after obtaining the proper authentication from Candle Corporation.

Chapter Contents

JES Manager Components .......................................................... 70
Controlling JES Manager Resources ......................................... 85
JES3 Start Application .............................................................. 92
JES Manager Components

Automation Applications
The JES Manager tables hold information used to manage JES3 and all of its related resources. All of the tables are members of the Automation Application table dataset, hileu.RKAUTBLR. The tables are created by the user by filling in the table fields with the appropriate information for JES3 and related resources. The section, Automation Table Component beginning on page 73, describes the JES Manager automation tables in detail.

Service Subroutines
JES Manager service subroutines are AF/OPERATOR command files used to perform a variety of functions for the resources controlled by the JES Manager. The service subroutines are comprised of the following execs:

@JESINIT
This exec initializes the JES Manager environment and creates an MVS console command, called JES, that allows you to perform various management functions on a single JES3 Resource or group of JES3 Resources.

@JESMAIN
This exec processes JES Manager commands issued to manage JES3 resources. @JESMAIN is responsible for: validating and processing command text, determining if the specified JES3 resource is defined to the system, responding to operator inquiries, and starting/stopping JES3 resources.

@JESCLAS
This exec is called by @JESMAIN to start, stop, and monitor JES3 Job Classes.

@JESCNFM
This exec is called by @JESMAIN to confirm a remote JES3 Manager function request.

@JESCWTR
This exec is called by @JESMAIN to issue an "*I A D=WTR" command, traps the IAT8522 response message and cancels all writers that are in rescheduled status.

@JESDMSC
This exec is called by @JESMAIN to start, stop, and monitor JES3 DSPs.
### JES Manager Components

<table>
<thead>
<tr>
<th>EXEC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@JESDNCN</td>
<td>This exec is called by @JESMAIN to start, stop, and monitor JES3 NJECONS.</td>
</tr>
<tr>
<td>@JESDNJE</td>
<td>This exec is called by @JESMAIN to start, stop, and monitor JES3 NJE lines.</td>
</tr>
<tr>
<td>@JESDNSN</td>
<td>This exec is called by @JESMAIN to start, stop, and monitor JES3 NJERDR.</td>
</tr>
<tr>
<td>@JESFCIS</td>
<td>This exec is called by @JESMAIN to start, stop, and monitor JES3 CI FSS.</td>
</tr>
<tr>
<td>@JESGRUP</td>
<td>This exec is called by @JESMAIN to start, stop, and monitor JES3 Job Groups.</td>
</tr>
<tr>
<td>@JESDSTR</td>
<td>This exec is called by @JESMAIN to format a display of JES3 start information</td>
</tr>
<tr>
<td>@JESQLST</td>
<td>This exec is called by @JESMAIN to format a display of JES3 output queue information. Jobs in the output queue are displayed in descending use order. The default is all JES3 output classes.</td>
</tr>
<tr>
<td>@JESCALL</td>
<td>This exec is an external REXX function used by other Automation Applications to internally request JES services.</td>
</tr>
<tr>
<td>@JESOAMQ</td>
<td>This exec is called by @JESMAIN to delete highlighted non-WTOR messages from the JES3 operator action message queue.</td>
</tr>
<tr>
<td>@JESQLST</td>
<td>This exec is called by @JESMAIN to format a display of JES3 output queue information. Jobs in the output queue are displayed in descending use order. The default is all JES3 output classes.</td>
</tr>
<tr>
<td>@JESQMON</td>
<td>This exec Monitors JES3 output queue space via the response to the *I Q S command. Informs operator when used queue space exceeds threshold value. Optionally, calls JES QPURGE function, to relieve shortages.</td>
</tr>
<tr>
<td>@JESQPRG</td>
<td>This exec is called by @JESMAIN to purge sysout data in JES WTR and HOLD queues.</td>
</tr>
<tr>
<td>@JESSTRT</td>
<td>This exec starts JES3. Responds to WTORs and accounts for various JES3 start types. For the global processor, issues *S JSS and varies local processors online to the global. For further information on the @JESSTRT exec, refer to page 92.</td>
</tr>
</tbody>
</table>
JES Manager Components

@JESSWTR This exec varies a JES3 writer onto the global processor and starts the writer FSS.

@JESVARY This exec varies a JES3 device online or offline. Returns 1 if the *VARY command was successful and 0 if not.

@JESXPBS This exec is a Pre-stop Exit for SNA/RJP Writers. It is used to issue JES & JSX cleanup commands for an RJP writer.

@JESXPBW This exec is a Pre-stop Exit for JES3 Writers. It is used to set the @JOBNUM variable for the STM stop verification message, delete the IAT7089 Writer Job Number trap, and vary the writer offline prior to termination.

@JESXPWS This exec is a Stop WTOR Exit for SNA/RJP Writers. It is used to issue JES *FAIL to force RJP writer to terminate.

@JESXPWW This exec is a Stop-WTOR Exit for JES3 Writers. It is used to issue an MVS Cancel command to bring the writer down should the standard stop command fail.

@JESXSAS This exec is a Post-start Exit for SNA/RJP DSP failures. It searches for IAT2855 SNA/RJP open ACB failure WTOR, if found, reply RETRY to start SNA/RJP activity.

@JESXSBS This exec is a Pre-start Exit for SNA/RJP Writers. It is used to issue JES & JSX setup commands for an RJP writer.

AF/OPERATOR System Variables

The JES Manager creates the AF/OPERATOR system variables to track JES3 resources over time:

@JESCMNS JES3 Complex Main Names
@JESCSMF JES3 Complex SMF IDs
@JCss##F JES3 Job Class Checkpoint
@JCss##S JES3 Job Class Status
@JDSPxxN JES3 DSP Name
@JDSPxxS JES3 DSP Status
@JFxxxxF JES3 CI FSS Checkpoint
@JFxxxxN JES3 CI FSS Name
Automation Table Component

The JES Manager Control table is used to store information on the JES3 objects controlled by the JES Manager.

@JESCNTRL - JES3 Control Table

The JES Manager Control table is used to store information on the JES3 objects controlled by the JES Manager.

SMFID

The SMFID of the system on which to manage the JES3 started task. This is an 8 character alphanumeric field assigned by the user. Any alphanumeric character may be used for the SMFID with the exception of an (*) or (").

JES_TASK

This field specifies the actual started task name of the JES3 subsystem. This name should match the name of the JES3 task as is coded in the Started Task Manager. Usually the task name is JES3, however this field allows the user to override this value.
### JES Manager Components

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JES_INT</strong></td>
<td>This field is used to specify the verification interval for JES commands. When a command is issued, the JES Manager will wait for the time limit specified by this field for verification that the command was successful. Code the verification interval in the form: HH:MM:SS. The default is 30 seconds, 00:00:30.</td>
</tr>
<tr>
<td><strong>JSX_TASK</strong></td>
<td>This field is used to specify the JSX started task name you want the JES Manager to use when issuing commands to start or stop the main JSX printer address space as part of a pre-start and pre-stop exits for an SNA/RJP writer Started Task Manager definition.</td>
</tr>
<tr>
<td><strong>QS_MAX</strong></td>
<td>This is a numeric field and is used as the threshold value in determining when to inform the operator via a WTO that the current available spool space value is below the threshold value.</td>
</tr>
<tr>
<td><strong>QS_PRG</strong></td>
<td>This is a binary field (0,1). Specify 1 to have the JES manager issue the JES START QPURGE command if the available spool space is below the user-specified threshold (QS_MAX above). Specify 0 if you do not want the JES START QPURGE command issued.</td>
</tr>
<tr>
<td><strong>QS_PART</strong></td>
<td>This field is used to indicate the JES3 spool partition name. If there is no valid spool partition name then code an * in this field.</td>
</tr>
<tr>
<td><strong>CLS_OBJ</strong></td>
<td>The name of the JES Job Class object to manage. This is a 12 character alphanumeric field that defines the user assigned Job Class object name. Any alphanumeric character may be used for the name with the exception of an asterisk or double quotes. Object names need not be unique; an object name may be repeated if the System or CPU ID keys are used to uniquely identify the row. This is a required field.</td>
</tr>
</tbody>
</table>

### @JESCLAS - JES3 Job Class Table

The JES Manager Job Class table is used to store information needed by the JES Job Class Manager to manage an object.
**JES Manager Components**

**CLS_SYS**
The system field is used to identify the scope of the Job Class object definition. Code the DEFAULT keyword to enable the definition on all systems. Code an SMFID to enable the object definition only on a specific system. Each JES Job Class object must have a DEFAULT definition. A DEFAULT definition will automatically be added if one does not exist. If coding a DEFAULT definition that will not be used to manage the job class, code an asterisk in the Job Class State field.

**CLS_CPU**
This field is used to identify the scope of the Job Class object definition based on CPUID. Code an asterisk in this field if the definition is not dependent on CPUID. Code a CPUID to enable the object definition only on a specific system, running on a specific CPU. This field is used in combination with the System field; an SMFID must be coded in the System field before a CPUID will be recognized. A CPUID can be coded as either the full six character CPU ID or just the right-most five characters for those Job Class object definitions that are not LPAR dependent.

**CLS_NAME**
This field is used to specify the Job Class Name for the object definition. This is an 8 character alphanumeric field, as specified in the JES Initialization parameters. This is a required field.
The Keyword field is used as a generalized grouping mechanism to identify those objects that are defined to all systems, no systems, or manipulated internally by other Automation Applications. Valid values are:

- **ALL**: Specifies that the object is defined to all systems. The JES Manager will initialize this object on every system.

- **AUTO**: Specifies that the object is defined to all systems. Under most circumstances, the AUTO keyword is identical to the ALL keyword, except in the case of bringing up a maintenance system running in manual application processing mode. When running in manual mode, only those objects identified as AUTO will be managed. When running in automated mode, the JES Manager will initialize the object on every system.

- **NONE**: Specifies that the object is not defined to any system. The JES Manager will not initialize this object on any system.

**Note:** The object can be initialized via manual intervention by specifying the Force option on a JES INIT command. This keyword is used to identify those objects that should not be started automatically, but rather, are started on-demand via manual intervention.

- **MANUAL**: Specifies that the object is not to automatically start during JES Manager initialization. During initialization, or anytime a JES INIT CLASS or JES START CLASS command is processed, the JES Manager will bypass this object. An object with a Keyword of MANUAL will be managed only if a JES INIT CLASS command for the object itself is manually entered.
**CLS_KEY cont.**

- **INTERNAL** Specifies that the object will be initialized internally by another of the Automation Applications; the object will not automatically start during JES Manager initialization. An object defined as INTERNAL can only be accessed by the controlling application.

  **Note:** Objects defined with a Keyword of INTERNAL are hidden from normal JES Manager operations. INTERNAL objects cannot be accessed directly by JES Manager commands, unless the Force option of the JES command is used. This includes all JES Manager displays for the object. This is a required field.

**CLS_INT**

This field is used to specify the verification interval for JES Vary commands. When a command is issued to vary a job class on or off, the JES Manager will wait for the time limit specified by this field for verification that the command was successful. Code the verification interval in military time format. The default is 2 minute, 00:02:00.

**CLS_STATE**

Specifies the initial status of the job class. A value of 1 indicates the job class should be started during initialization. A value of 0 indicates it should be stopped. A value of (*) indicates the job class is not defined to this system and should not be managed.

**CLS_DESCR**

A one line description of the object. The object’s description field is used to provide a ‘Plain English’ description of the object for user reference. Up to 20 characters of free form text is allowed.
### @JESDMSC - JES3 DSP Control Table

The JES DSP Control table is used to store information on the JES3 DSPs controlled by the JES Manager.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>Card Readers</td>
</tr>
<tr>
<td>DR</td>
<td>Disk Readers</td>
</tr>
<tr>
<td>TR</td>
<td>Tape Readers</td>
</tr>
<tr>
<td>CI</td>
<td>This field is used to specify the maximum number of CI DSPs that can operate in the JES3 global address space at any time. The first subparameter (maxbatch) specifies the maximum number of CI DSPs that process batch jobs. The second subparameter (maxdemsel) specifies the maximum number of CI DSPs that process demand select jobs (that is, started tasks and TSO LOGONs).</td>
</tr>
<tr>
<td>POSTSCAN</td>
<td>This field is used to specify the maximum number of POSTSCAN DSPs that can operate in the JES3 global address space at any one time. The first subparameter (maxbatch) indicates the maximum number of POSTSCAN DSPs that can process batch jobs. The second subparameter (maxdemsel) indicates the maximum number of POSTSCAN DSPs that process demand select jobs (that is, started tasks and TSO LOGONs).</td>
</tr>
<tr>
<td>DISABLE</td>
<td>Disable</td>
</tr>
<tr>
<td>INTRDR</td>
<td>This field is used to define the maximum number of internal readers that can be active concurrently. Specify any value between 1 and 999 inclusive for the variable nnn.</td>
</tr>
<tr>
<td>PURGE</td>
<td>Job Purge</td>
</tr>
</tbody>
</table>
@JESDNJE - JES3 NJE Control Table

The JES NJE Control table is used to store information on the JES3 NJEs controlled by the JES Manager.

**NJ_E_INT**
This field is the amount of time IRM will wait for a response after attempting to retrieve NJE line definition(s) from table @JESDNJE. Specify in the form hh:mm:ss and within the range 00:00:01 to 24:00:00.

**RDR**
This field is used to specify the number of NJE RDRs that are active on this system.

**NJ_E.0**
This field is an array key that signifies the number of defined NJEs for the system. The default is 0.

**NJ_E_LN.n**
The NJE_LN.1 through NJE_LN.8 fields allow the user to specifically define up to 8 unique NJE line numbers for each SMFID.

**NJ_E_NM.n**
The NJE_NM.1 through NJE_NM.8 fields allow the user to specifically define up to 8 unique NJE line names for each SMFID.

**NJ_E_OP.n**
The NJE_OP.1 through NJE_OP.8 fields allow the user to specifically define up to 8 unique NJE options for each SMFID. Valid values are:

- LOG
- NORCV
@JESFCIS - JES3 CI FSS Control Table

The JES CI FSS Control table is used to store information on the JES3 CI FSS controlled by the JES Manager.

CI_NAME
This field is used to define the CIFSS task specific to this SMFID.

CI_DSPC
This field is used to specify the maximum number of CI DSPs that can operate in the C/I FSS address space at any time. The first subparameter (maxbatch) specifies the maximum number of CI DSPs that process batch jobs. The second subparameter (maxdemsel) specifies the maximum number of CI DSPs that process demand select jobs (that is, started tasks and TSO LOGONs).

CI_MAST
This field is used to specify the maximum number of JCL statements that can be processed concurrently by all CI DSPs in the C/I FSS address space. The value must be an integer between 0 and 99999999, inclusive.

CI_P
This field is used to specify a member of the procedure library for started task jobs, which contains a cataloged procedure for starting the FSS. The member must be in the procedure library defined by the STCPROC parameter of the STANDARDS statement, or in procedure library IATPLBST, if the STCPROC parameter is omitted.

CI_SYS
This field is used to specify the JES3 main on which the FSS is to operate. The name(s) must be the same as specified on the NAME parameter of the MAINPROC statement for the main.

CI_T
This field is used to specify whether or not the FSS terminates if the JES3 global terminates as the result of an *RETURN or *DUMP operator command.

CI_ST
This field is used to specify whether or not JES3 should start the FSS automatically when the main on which the FSS is to run is connected to the global. This parameter applies only to C/I FSSs.
@JESGRUP - JES3 Job Group Table

The JES Manager Job Group table is used to store information needed by the JES Job Group Manager to manage an object.

**GRP_OBJ**

The name of the JES Job Group object to manage. This is a 12 character alphanumeric field that defines the user assigned Job Group object name. Any alphanumeric character may be used for the name with the exception of an asterisk or double quotes. Object names need not be unique; an object name may be repeated if the System or CPUID keys are used to uniquely identify the row.

**GRP_SYS**

The System field is used to define the systems where the object will be managed. The DEFAULT keyword indicates the object will be managed on all systems. An SMFID signifies a single system where the object is to be managed.

**GRP_CPU**

The CPU field is used to define specific CPUIDs where the object is to be managed.

**GRP_NAME**

This field is used to specify the Group Name for the object definition. This is an 8 character alphanumeric field, as specified in the JES Initialization parameters. This is a required field.
**GRP KEY**

The Keyword field is used as a generalized grouping mechanism to identify those objects that are defined to all systems, no systems, or manipulated internally by other Automation Applications. Valid values are:

- **ALL**
  - Specifies that the object is defined to all systems. The JES Manager will initialize this object on every system.

- **AUTO**
  - Specifies that the object is defined to all systems. Under most circumstances, the AUTO Keyword is identical to the ALL keyword, except in the case of bringing up a Maintenance system running in manual application processing mode. When running in manual mode, only those objects identified as AUTO will be managed. When running in Automated mode, the JES Manager will initialize the object on every system.

- **NONE**
  - Specifies that the object is not defined to any system. The JES Manager will not initialize this object on any system.

**Note:** The object can be initialized via manual intervention by specifying the Force option on a JES INIT command. This keyword is used to identify those objects that should not be started automatically, but rather, is started on-demand via manual intervention.

- **MANUAL**
  - Specifies that the object is not to automatically start during JES Manager initialization. During initialization, or anytime a JES INIT GROUP or JES START GROUP command is processed, the JES Manager will bypass this object. An object with a Keyword of MANUAL will be managed only if a JES INIT GROUP command for the object itself is manually entered.

- **INTERNAL**
  - Specifies that the object will be initialized internally by another of the Automation Applications; the object will not automatically start during JES Manager initialization. An object defined as INTERNAL can only be accessed by the controlling application.
Note: Objects defined with a Keyword of INTERNAL are hidden from normal JES Manager operations. INTERNAL objects can not be accessed directly by JES Manager commands, unless the Force option of the JES command is used. This includes all JES Manager displays for the object. This is a required field.

GRP_INT
This field is used to specify the verification interval for JES Vary commands. When a command is issued to vary a job group on or off, the JES Manager will wait for the time limit specified by this field for verification that the command was successful. Code the verification interval in military time format. The default is 2 minute, 00:02:00.

GRP_STATE
Specifies the initial status of the job group. A value of 1 indicates the job group should be started during initialization. A value of 0 indicates it should be stopped. A value of (*) indicates the job group is not defined to this system and should not be managed.

GRP_COUNT
Specifies the number of initiators to assign exclusively to this group. This count defines the maximum number of jobs of this group that can be concurrently scheduled to a specified main.

GRP_ALLOC
Specifies when the execution resources are to be allocated to the group. Valid values are: DEMAND, DYNAMIC, IPL, or MANUAL.

GRP_UNALC
Specifies when the execution resources are to be released from the group. Valid values are: DEMAND, DYNAMIC, or MANUAL.

GRP_JSPAN
Specifies the number of jobs in this group to be examined in selecting a job to be scheduled. Valid values are: ALL or a decimal number from 1-32767.

GRP_BAR
Specifies a job priority barrier. All jobs in this group that has a priority equal to or above this barrier must be scheduled before any attempt is made to schedule jobs below this barrier. Valid values are: PRTY or a decimal number from 0-16.

GRP_DESCR
The Description is a one line plain-English description of the entry defined by the user. The description is entered when the entry is first added to the JES Job Group Manager.
@JESQPRG - JES3 Output Queue Mgmt Table

The JES Output Queue Management table is used to store information to manager jes3 output queues controlled by the JES Manager.

A-Z 0-9 The fields A-Z and 0-9 are used to define the disposition of held output in the format nnnD/H where nnn defines the minimum number of days (D) or hours (H) the network stream or sysout stream has been on the BDT queue.
Controlling JES Manager Resources

JES Command Syntax

FIGURE 4. JES Command Syntax

<table>
<thead>
<tr>
<th>JES function</th>
<th>object</th>
<th>options</th>
<th>jesopts</th>
</tr>
</thead>
<tbody>
<tr>
<td>INIT</td>
<td>CI</td>
<td>DSP_PARM</td>
<td></td>
</tr>
<tr>
<td>START</td>
<td>CIFSS</td>
<td>MAIN</td>
<td>MOD_CIF</td>
</tr>
<tr>
<td>STOP</td>
<td>CLASS</td>
<td>JOBCLASS</td>
<td>MAIN</td>
</tr>
<tr>
<td>MODIFY</td>
<td>CR</td>
<td>DSP_PARM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DISABLE</td>
<td>DSP_PARM</td>
<td></td>
</tr>
<tr>
<td>ACTIVATE</td>
<td>DR</td>
<td>DSP_PARM</td>
<td></td>
</tr>
<tr>
<td>INACTIVATE</td>
<td>GROUP</td>
<td>JOBGROUP</td>
<td>MAIN MOD_GRP</td>
</tr>
<tr>
<td>EXCEPT</td>
<td>INTRDR</td>
<td>DSP_PARM</td>
<td></td>
</tr>
<tr>
<td>NOOP</td>
<td>NJE</td>
<td>LINE</td>
<td>MOD_NJE</td>
</tr>
<tr>
<td></td>
<td>NJECONS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HELP</td>
<td>NJERDR</td>
<td>RDR_COUNT</td>
<td></td>
</tr>
<tr>
<td>SHOW</td>
<td>MSGCLEAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STATUS</td>
<td>POSTSCAN</td>
<td>DSP_PARM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PURGE</td>
<td>DSP_PARM</td>
<td></td>
</tr>
<tr>
<td>DISPLAY</td>
<td>QLIST</td>
<td>CLASS</td>
<td>LINE</td>
</tr>
<tr>
<td></td>
<td>QMON</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>QPURGE</td>
<td>CLASS</td>
<td>RETEN MOD_QUEUE</td>
</tr>
<tr>
<td></td>
<td>SNANJE</td>
<td>RDR_COUNT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>START</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TR</td>
<td>DSP_PARM</td>
<td></td>
</tr>
</tbody>
</table>

Functions
The function parameter is used to tell the JES Manager which action to take against a resource. Many, but not all, functions have abbreviations. Acceptable abbreviations are noted with the function’s description.

**INIT**
Reinitializes a resource defined to the JES Manager so that changes to the resource’s definition will take effect immediately without affecting the status of other resources. The resource will be started automatically if it should be running, and likewise, stopped if it should not be running at the current time. No abbreviations of the INIT keyword are acceptable. Initializing a resource causes native MVS command traps to be set, JES Manager resource status variables to be initialized and the resource to be started/stopped, if appropriate.

**START**
Starts a resource defined to the JES Manager immediately, rather than at the time specified in the resource definition. This function will not start a resource that is already active. No abbreviations of the START keyword are acceptable.

**STOP**
Stops a resource defined to the JES Manager immediately, rather than at the time specified in the resource definition. No abbreviations of the STOP keyword are acceptable.

**MODIFY**
Modifies an option associated with a JES Manager resource. Valid abbreviations are: MOD and F.

**ACTIVATE**
Changes a previously inactivated resource back to active status. This enables JES Manager control of a resource and actions can once again be performed for the resource. Valid abbreviations are: ACT and A.

**INACTIVATE**
Changes a resource to INACTIVE status. This effectively disables JES Manager control of the resource; no further action can be taken for this resource until it is reactivated. Valid abbreviations are: INACT and I.

**EXCEPT**
Temporarily except an object from this system.

**NOOP**
Perform no operation. NOOP is used to restore checkpointed control information, such as FORCE or EXCEPT, for an object.

**HELP**
Display online JES Manager help.

**SHOW**
Shows all AF/OPERATOR (JES Manager created) traps associated with the resource. Valid abbreviations are: SH.
Controlling JES Manager Resources

**STATUS**

Issues a WTO to the MVS console indicating the JES Manager status for the specified resource. Valid abbreviations are: STAT and S.

A resource can be in any one of the following states:

- **DEFINED** - The JES Manager knows about the resource, but does not know whether the resource is active or inactive.
- **INACTIVE** - The resource is not active.
- **ACTIVE** - The resource is active or has been enabled.
- **DE-ACTIVATED** - The resource is no longer under the control of the JES Manager. A JES INACTIVATE command has been issued for the resource.
- **UNKNOWN** - The resource's status is unknown. This is usually the case if the JES Manager has not fully initialized.

**DISPLAY**

Displays the requested JES Manager information via WTO to the MVS console. The default display is the same as the STATUS display as described above. Valid abbreviations are: DIS and D.

**Object**

The object name parameter is used to tell the JES Manager which object to manage. The aforementioned function will be performed for the resources specified with this parameter.

- **CI** - Converter/Interpreter
- **CIFSS** - CI Functional subsystem
- **CLASS** - Job class
- **CR** - Card reader DSP
- **DISABLE** - Disable DSPs
- **DR** - Disk reader DSP
- **GROUP** - Initiators
- **INTRDR** - Internal reader
- **NJE** - Network Job Entry links
- **NJECONS** - NJE console support
### Controlling JES Manager Resources

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NJERDR</strong></td>
<td>SNA/NJE reader DSPs</td>
</tr>
<tr>
<td><strong>MSGCLEAR</strong></td>
<td>Operator action message queue clearing</td>
</tr>
<tr>
<td><strong>POSTSCAN</strong></td>
<td>Postsan DSP</td>
</tr>
<tr>
<td><strong>PURGE</strong></td>
<td>Purge DSP</td>
</tr>
<tr>
<td><strong>QLIST</strong></td>
<td>Output queue job list+</td>
</tr>
<tr>
<td><strong>QMON</strong></td>
<td>Output queue utilization monitoring+</td>
</tr>
<tr>
<td><strong>QPURGE</strong></td>
<td>Output queue clearing+</td>
</tr>
<tr>
<td><strong>SNANJE</strong></td>
<td>SNA/NJE reader DSPs</td>
</tr>
<tr>
<td><strong>START</strong></td>
<td>Display start information++</td>
</tr>
<tr>
<td><strong>TR</strong></td>
<td>Tape reader DSP</td>
</tr>
</tbody>
</table>

**Note:** Those items marked with a + are START only Function/Object. Those items marked with a ++ are DISPLAY only Function/Object.

### JES Options

The options are to be used in conjunction with a specified object.

- **CLASS** Specify a sysout class
- **DSP_PARM** Parameters associated with a specified DSP. Valid parameters are:
  - **MC=nn or (nn,nn)** DSP count
  - **MAXASST=nn** JCL statements (STC)
  - **MAXJOBST=nn (JOB)** JCL statements (JOB)
  - **HOLD** Stop CI processing
  - **RELEASE** Start CI processing
- **JOBCLASS** Batch job class or classes
- **JOBGROUP** Batch job initiator group or groups.
- **LINE** NJE node or line name.
- **LINES** Maximum lines in DISPLAY
- **MAIN** System SMFID(s) or JES3 main name(s).
Controlling JES Manager Resources

MOD_CIFSS  CIFSS modify parameters. Valid parameters are:
- **DSPC=(nn,nn)**  DSP count
- **MAST=(nn,nn)**  JCL statements
- **P=procname**  PROCLIB option
- **SYS=system**  FSS system override
- **T=terminate_option**  Dump option
- **ST=Y/N**  Start option

MOD_GRP  Group modify parameters. Valid parameters are:
- **INIT=n**  Initiator count
- **ALLOC=n**  Allocation value
- **UNALLOC=n**  Unallocation value
- **JSPAN=n**  JSPAN value
- **BAR=n**  Barrier value

MOD_NJE  NJE modify parameters. Valid parameters are:
- **FORCE**  Reset network line(s).
- **HOLD/NOHOLD**  Hold/Nohold job transmission.
- **Path/p=node**  Route thru path=node.
- **PWCNTL/PC=key**  Job password control option.
- **R=##**  Line retry count (1-300).
- **TYPE=BSS/SNA**  Network protocol to use.
- **RCV/NORCV**  Receive / No receive
- **LOG/NOLOG**  Start/stop line tracing.

MOD_QUE  QPURGE modify parameters. Valid parameters are:
- **ID=userid**  Purge jobs for ID userid.
- **L=nnn**  Purge jobs with lines > nnn.
- **PG=nnn**  Purge jobs with pages > nnn.
- **DG=groupname**  Purge jobs with groupname
- **T=terminal_name**  Purge jobs for RJP terminal.
- **D=dest**  Purge jobs with specific destination.
- **F=form**  Purge jobs with specific form.
- **P=prty**  Purge jobs with specific priority.

RDR_COUNT  Count of NJERDR DSPs.
Controlling JES Manager Resources

RETENTION
Sysout retention period. Specified as number of hours/number of days, i.e. nH/nD.

SMFID
System SMFID or JES3 Main Name.

Standard Options
The standard options are used to alter the normal JES Manager processing of a function for a resource.

FORCE
The FORCE option is used to force a resource that is not defined to a system onto that system. The FORCE option can also be used to force a JES Manager function to complete. Specifying force will override all JES Manager checks and pre-requisites. Valid abbreviations are: F.

The FORCE option can be used to:

- Override table definitions and force a resource to run on a system where it is not defined to run.
- Cancel outstanding JES Manager function requests for the resource.

Note: If a resource is forced onto a system via a JES INIT or JES START command, the JES Manager will update the in-storage copy of the Task Control table and define the resource to the system. Any time after this, you need not use the FORCE option on the JES command to manage the resource.

This control definition change is checkpointed and will remain in effect until the next IPL or until AF/OPERATOR is started in manual mode between system IPLs. After a system IPL, the definition for the resource will revert to that specified in the Task Control Table.

MAINT
The MAINT option is used to manage only those resources that have been defined with a KEYWORD of AUTO. This is a special option used to IPL a maintenance system. Valid abbreviations are: M.

CALL TYPE=ct
Internal option that is used to specifically declare the REXX call type for a cross system function request.
Controlling JES Manager Resources

The CONSOLE option is used to specify the console to receive JES Manager messages for a function. Specify either a valid console ID or the device address of the console for the console_id argument. There are no abbreviations for the CONSOLE option.

CONFIRM

Confirms remote function requests. Send confirmation to system SYSID.

JES Command On-line Help

On-line help for the JES Manager JES command is available anytime. To display on-line help for the JES command, enter the following command at the MVS operator console:

JES HELP

The JES Manager will then display the JES command syntax followed by a brief explanation of each of the keywords associated with the command.
Overview of Operation

The JES3 Start Application is used to automate the start of JES3. The application is implemented as a start command procedure executed by the Started Task Manager. The activities of the application are controlled by information stored in a table created by the user. The table that contains the JES3 start information is:

@JESSTRT  The JES3 Start Table

The table is a member of the Automation Application table dataset, hilevel.RKAUTBLR. The table is loaded into above-the-line storage at AF/OPERATOR and Automation Application initialization time.

When an RCF START ALL or RCF START JES3 command is received, information is retrieved from this table by the @JESSTRT service routine and used to determine how to start JES3.

The JES3 start procedures will vary depending on the type of IPL requested. The possible IPL types are: Maintenance Mode, Automated, and Checkpoint Restart. The JES3 Start Application performs the following processes for each of the possible IPL types:

Maintenance Mode JES3 Start

Issue a WTO to the operator console, which displays the default JES3 start parameters and allows for overrides to the start information. Three options are available at this point:

- Reply A to abort the automated start of JES3.
- Reply C to continue the automated start of JES3, by AF/OPERATOR, with the default start parameters.
- Reply to the WTO with override information for the start command, start type and/or Inish deck parameters. The format of the reply is as follows:
  - Start command
    
    R 00,START=new_start_command
– Start type

   **R 00,TYPE=new_start_type**

– Inish deck

   **R 00,INISH=new_inish_deck**

Only one override at a time may be specified to the WTOR. When an override reply is received, the WTOR will be re-displayed with the new information. If more than one override is necessary, continue replying to the WTOR, until all of the necessary overrides are in place. The WTOR will continue to be displayed until either an A or C is received.

Once the necessary overrides are in place, reply C to the WTOR to initiate the start of JES3.

**Note:** If the abort option, A, was chosen, JES3 will need to be started manually.

**Automated Mode JES3 Start**

If the continue option, C, was chosen for the maintenance mode WTOR or if this is an automated mode IPL, the JES3 Start Application will proceed to start JES3:

- Determine if this system is a Global or a Local processor by interrogating the Automation Application JES3 processor definition variable, @APLJES, and the specified JES3 start type. If the information does not match, a WTOR will be issued to the operator requesting the correct JES3 processor definition.

- Determine if the system has been IPL'ed:
  - If the system has been IPL'ed and this is a Global processor, the application will validate the JES3 start type. If the provided start type is incorrect, the default start type of Hot, H, will be used.
  - If this is not an IPL and this is a Global processor, the application will ensure that the start type is either Hot, H, or Hot with Analysis, HA. If the provided start type is incorrect, the default start type of Hot, H, will be used.

- Set the traps for JES3 start WTORs.

- Issue the JES3 start command.
JES3 Start Application

- Reply with the defined JES3 Main name.
- Reply with the specified JES3 start type.
- Reply with the specified JES3 Inish deck member.
- Reply 'U' for cold start confirmation, if necessary.
- Reply 'DONE' for Local processors disabled, if necessary.
- Reply 'CONTINUE' for allocation failures, if necessary.
- Issue the JES3 *S JSS command.
- Check for the JES3 start verification message.
  - If this is a JES3 Local processor that has been started prior to global processor initialization, JES3 will issue a WTOR to allow Local start retry. The JES3 Start Application recognizes this WTOR and will suspend the start of the Local processor for 2 minutes and then retry the Local start. If the Local start is still unsuccessful, the application will issue a warning message to the operator.
  - If the start verification is not received on either the Global or Local processor, the Started Task Manager will issue a WTOR informing the operator that the JES3 start has failed.
  - If the start verification is received and this is a JES3 Global processor, the JES3 Start Application will issue the JES3 vary on-line commands for all of the defined Local processors.
- Set the JES3 Main name and JES3 complex information variables.

Checkpoint Restart JES3 Start

For a checkpoint restart of AF/OPERATOR and the Automation Applications, the JES3 Start Application will start JES3 as previously described for an automated start. The only difference for a checkpoint restart is that all previously entered JES3 start override parameters will be restored from checkpointed AF/OPERATOR system variables. This means, that if the defined JES3 processor definition, start command, start type, or Inish deck was dynamically changed the last time AF/OPERATOR was running, the change will be restored and still be in effect for the current start of JES3.
Enabling the Application

To add a system to the JES3 Start Application, add a row to the @JESSTRT table, identifying the system by SMF ID, and fill in all of the fields in the table with the appropriate information. Then, specify:

```
EXEC @JESSTRT "JES3"
```

in the START_CMD field for the JES3 resource in the Started Task Manager Task Start Table. The next time AF/OPERATOR is started, the new information will be available to start JES3.

To update information for the JES3 Start Application, edit the table and change whatever fields are necessary. The next time AF/OPERATOR is started the changes will be available.

To implement the changes while AF/OPERATOR is running, use the Table Manager REFRESH command to refresh the in-storage copy of the JES3 Start Table, and if necessary, the Started Task Manager Task Start Table. This will give the Started Task Manager and the JES3 Start Application access to the new data that will be in effect for the next start of JES3.
Control Options

@JESSTRT - JES3 Start Table

The JES3 Start Application control information is stored in the JES3 Start Table, @JESSTRT. The following describes the control information held in the table:

SMFID

The SMFID of the system on which to manage the JES3 start. This field is the table row name. This is an 8 character alphanumeric field assigned by the user. Any alphanumeric character may be used for the SMFID with the exception of an asterisk (*) or double quotes (").

JES_MAIN

The JES3 Main name as specified in the JES3 Inish deck for the system.

JES_INISH

Specify the suffix (last 2 characters) of the member name for the JES3 Inish deck to be used at JES3 initialization. Unless overridden during a maintenance mode IPL, this will be the default JES3 Inish deck.

START_TYPE

Specify the JES3 start type. See the IBM manual: MVS/ESA Operations: JES3 Commands, SC23-0074, for valid JES3 start types. Unless overridden during a maintenance mode IPL, this will be the default JES3 start type.

JES_START_CMD

Specify the JES3 start command. Unless overridden during a maintenance mode IPL, this will be the default JES3 start command.

JES_INT

The interval specified in military time format, HH:MM:SS., to wait for the JES3 start verification message. The default is 2 minutes, 00:02:00.

LOCAL.0

This field is an array key that specifies the number of defined Local processors for this Global processor. The default is zero (0).

LOCAL.n

Specify the JES3 Main name for all Local processors. Code an asterisk (*) in all unused fields.
RCF Command Overrides

The JES3 start overrides described for a maintenance mode IPL, can also be passed to the JES3 Start Application as part of a Started Task Manager RCF command for JES3:

- `RCF START JES3 ADDPARM(START=new_start_command)`
- `RCF START JES3 ADDPARM(TYPE=W)`
- `RCF START JES3 ADDPARM(TEMPTYPE=HA)`
- `RCF START JES3 ADDPARM(INISH=VM)`
- `RCF START JES3 ADDPARM(START=new_start_command TEMPTYPE=HA)`

The following options can be passed to the JES3 Start Application via the RCF command:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>START=start_command</code></td>
<td>Override start command for JES3</td>
</tr>
<tr>
<td><code>TYPE=type</code></td>
<td>Override start type for JES3.</td>
</tr>
<tr>
<td><code>TEMPTYPE=type</code></td>
<td>Override start type for JES3. The difference between TYPE and TEMPTYPE is</td>
</tr>
<tr>
<td></td>
<td>that the override start type specified by the TEMPTYPE option will not be</td>
</tr>
<tr>
<td></td>
<td>checkpointed; the start type will only be in effect for the current start</td>
</tr>
<tr>
<td></td>
<td>of JES3.</td>
</tr>
<tr>
<td><code>INISH=deck</code></td>
<td>Override Inish deck for JES3</td>
</tr>
</tbody>
</table>
Introduction

The TSO Manager is used to manage the pre-stop activities associated with stopping TSO.

Chapter Contents

TSO Shutdown Application ........................................ 100
Enabling the Application ........................................ 102
TSO Command On-line Help ..................................... 104
TSO Shutdown Application

The application is implemented as a Started Task Manager pre-stop exit. The activities of the application are controlled by information stored in the table created by the user. The table that contains the control information is:

@TSOSTOP The TSO Stop Table

The table is a member of the Automation Application table dataset, hilevel.RKAUTBLR. the table is loaded into above-the-line storage at AF/OPERATOR and Automation initialization time.

When an RCF STOP ALL or RCF STOP TSO command is received, the Started Task Manager executes the pre-stop user exit for TSO called, @TSOSTOP. The @TSOSTOP service routine retrieves information from the TSO Stop Table, which controls the pre-stop services performed by the TSO Shutdown Application.

The TSO Shutdown Application performs the following services prior to TSO shutdown:

1. Determines if there are users currently logged onto TSO.
2. If there are no users logged onto TSO, the TSO shutdown process will continue uninterrupted. If there are users logged on to TSO, the application will:
   - Enable the TSO CONTINUE and TSO STOP operator commands and issue a console message informing the operator that users are being warned of a pending TSO shutdown.
   - Set TSO USERMAX to zero to prevent further TSO logons.
   - Issue shutdown warning messages to the users. The warning messages inform users that TSO is being shutdown and requests them to logoff. The warning message is designed so that as many messages as needed can be sent to the TSO user community. For each warning message, the following process takes place:
     - Warning message is issued noting the current time and time remaining until shutdown.
     - The TSO Shutdown Application then waits for the specified warning interval to elapse.
– Once the interval has elapsed, the TSO Shutdown Application determines if the operator has issued an override command:

3. If an operator has issued the TSO CONTINUE command, the warning process will be stopped and the TSO Shutdown Application will proceed to abort the TSO shutdown.

4. If an operator has issued the TSO STOP command, the warning process will be stopped and the TSO Shutdown Application will proceed to shutdown TSO immediately.

5. If no override command has been received, the warning process will continue until all warning messages have been issued.

6. After the user warning messages have been issued, the operator warning message will be deleted from the console and the operator override commands will be disabled.

7. If the operator has overridden TSO shutdown, the TSO Shutdown Application will:
   ■ Inform the users that TSO services will continue.
   ■ Reset the TSO USERMAX to once again allow TSO logons.
   ■ Issue a stop abort message to the operator console and return to the Started Task Manager signalling that TSO shutdown should be aborted.
   ■ Otherwise, TSO shutdown procedures will continue.

8. Set a trap to reply to the TSO shutdown WTOR with the designated response.

9. Return to the Started Task Manager to continue TSO shutdown.
Enabling the Application

To add a system to the TSO Shutdown Application:

1. Add a row to the @TSOSTOP table, identifying the system by SMFID
2. Fill in all of the fields in the row with the appropriate information.
3. Specify @TSOSTOP in the STOP_PRE field for the TSO resource in the Started Task Manager Task Stop Table.

The next time AF/OPERATOR is started, the new information will be available to stop TSO.

To update information for the TSO Shutdown Application, edit the table and change whatever fields are necessary. The next time AF/OPERATOR is started the changes will be available.

To implement the changes while AF/OPERATOR is running, use the Table Manager REFRESH command to refresh the in-storage copy of the TSO Stop Table and, if necessary, the Started Task Manager Task Stop Table. This will give the Started Task Manager and the TSO Shutdown Application access to the new data which will be in effect for the next TSO shutdown.
**TSO Shutdown Application**

### @TSOSTOP - TSO Stop Table

The TSO Shutdown Application control information is stored in the TSO Stop Table, @TSOSTOP.

The following describes the control information held in the table:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SMFID</strong></td>
<td>The SMFID of the system on which to manage TSO shutdown. This field is the</td>
</tr>
<tr>
<td></td>
<td>table row name. This is an 8 character alphanumeric field assigned by the</td>
</tr>
<tr>
<td></td>
<td>user. Any alphanumeric character may be used for the name with the exception</td>
</tr>
<tr>
<td></td>
<td>of an (*) or (&quot;).</td>
</tr>
<tr>
<td><strong>TSO_MAX</strong></td>
<td>The TSO USERMAX value as specified in SYS1.PARMLIB. This value is used by</td>
</tr>
<tr>
<td></td>
<td>the application to reset the TSO USERMAX after a TSO CONTINUE command has</td>
</tr>
<tr>
<td></td>
<td>been received. An (*) can be specified, which will cause the application to</td>
</tr>
<tr>
<td></td>
<td>reset the TSO USERMAX to the value it had prior to being set to zero.</td>
</tr>
<tr>
<td><strong>TSO_WARN</strong></td>
<td>The number of user warning messages that will be issued prior to TSO</td>
</tr>
<tr>
<td></td>
<td>shutdown. This must be a whole number. The default is 5.</td>
</tr>
<tr>
<td><strong>TSO_WINT</strong></td>
<td>The amount of time between each user warning message, in the form HH:MM:SS.</td>
</tr>
<tr>
<td></td>
<td>The default is one minute, 00:01:00.</td>
</tr>
<tr>
<td><strong>TSO_WAIT</strong></td>
<td>The amount of time to wait for commands to complete, in the form HH:MM:SS.</td>
</tr>
<tr>
<td></td>
<td>The default is one minute, 00:01:00. This value is used for the verification</td>
</tr>
<tr>
<td></td>
<td>interval when issuing TSO Modify commands to reset TSO USERMAX.</td>
</tr>
<tr>
<td><strong>TSO_REPL</strong></td>
<td>The reply to be issued to the WTOR when users are logged on at TSO shutdown.</td>
</tr>
<tr>
<td></td>
<td>Valid values are:</td>
</tr>
<tr>
<td></td>
<td>- <strong>SIC</strong> - Cancels active users normally. Users receive any messages</td>
</tr>
<tr>
<td></td>
<td>queued for them and TSO performs its normal termination process for the</td>
</tr>
<tr>
<td></td>
<td>users.</td>
</tr>
<tr>
<td></td>
<td>- <strong>FSTOP</strong> - Forces immediate cancellation of all active users. Users do</td>
</tr>
<tr>
<td></td>
<td>not receive messages queued for them. TSO will not perform normal</td>
</tr>
<tr>
<td></td>
<td>termination processing.</td>
</tr>
</tbody>
</table>
**TSO Command On-line Help**

On-line help for the TSO Manager TSO command is available anytime. To display on-line help for the TSO command, enter the following command at the MVS operator console:

**TSO HELP**

The TSO Manager will then display the TSO command syntax followed by a brief explanation of each of the keywords associated with the command.
Introduction

The CICS Manager is extended automation for CICS regions defined as resources in the Started Task Manager. To use the services of the CICS Manager the appropriate user exits for pre/post start, pre/post stop, recovery exits must be coded, as well as an entry for each of the CICS regions you want the CICS Manager to monitor and control. The CICS Manager can be used to:

- Manage CICS POVI logon for CICS
- Provides Escalated Shutdown with Options
- Provides Standard Routines for CICS Maintenance
- JPU/E -Journal Processing
- Transaction Dump Processing
- Automatic VTAM Node Recovery

The CICS Manager is initialized in a two step process. The basic CICS Manager environment is created at AF/OPERATOR initialization time. The actual management of CICS regions is enabled once the specified CICS regions have been started.

Note: CICS Manager is a separately billable item and can only be activated after obtaining the proper authentication from Candle Corporation.
Chapter Contents

CICS Manager Components .................................................. 107
Automation Application Table .............................................. 107
Service Subroutines .......................................................... 107
ISPF User Interface ........................................................... 109
Controlling CICS Manager Resources ................................. 114
CICS Manager Components

The CICS Manager has three components. They are:

- Automation application table
- Service subroutines
- ISPF User Interface

Automation Application Table

To manage the CICS POVI logons, provide escalated shutdown capabilities, CICS maintenance routines, journal and dump processing, and to provide VTAM node recovery the CICS Manager uses information stored in a table created by the user. The table that contains the CICS region information is:

@CICCNTL CICS Manager Control Table

This table is a member of the Automation Application table dataset, hilev:RKAUTBLR. The table is loaded into above-the-line storage at AF/OPERATOR and Automation Application initialization time. Any time AF/OPERATOR is started, the new CICS information will be available to the CICS Manager.

To implement changes while AF/OPERATOR is running, use the Table Manager REFRESH command to refresh the in-storage copy of the CICS Manager Control Table. This gives the CICS Manager access to the new data.

Service Subroutines

The CICS Manager service subroutines are AF/OPERATOR command files that perform the services needed to provide the extended services for the CICS regions. The CICS Manager service subroutines are comprised of the following execs:

@CICDEQ This exec is used to DEQUEUE an application-generated enqueue in case of code failure.
CICS Manager Components

@CICDJOB  This exec is called from the DFHDU0303I Message Manager Trap to Process CICS transaction dumps. It will copy the dump to sysout class to be archived to SAR.

@CICEXIT  This exec is called by application REXX code to invoke a CICS user exit and return the result of the exit to the caller.

@CICLOGF  This exec is called by application REXX code to logoff a CICS POVI session. Returns 1 if logoff was successful and 0 if not.

@CICLOGN  This exec is called by application REXX code to logon to a CICS region via POVI supplying the AF/OPERATOR CICS userid and password. Returns 1 if the logon was successful and 0 if not.

@CICMLGF  Called by application REXX code to logoff a CICS region from an MCS console. Returns 1 if the logoff is successful and 0 if not.

@CICMLOG  Called by application REXX code to logon a CICS region from an MCS console. Returns 1 if the logon is successful and 0 if not.

@CICPLGN  This exec is called by the Started Task Manager to stop the CICS POVI logon prior to the stopping of the CICS region.

@CICPOVI  This exec is called by application REXX code to issue a CICS command via POVI. Returns 1 if the command was successfully processed and 0 if not.

@CICPURG  This exec is used by the CICS Manager to display and purge all suspended transactions.

@CICSLGN  This exec is called by the Started Task Manager to start the POVI logon to a CICS region after a successful start of the CICS region.

@CICVRA1  This exec is called by the VTAM Manager to issue a CEMT SET VTAM OPEN after a VTAM node recovery attempt for a CICS region.
CICS Manager Components

@CICXPA1 This exec is called by the Started Task Manager as a CICS Post-Stop Exit. It will call a CICS region specific Post-Start exit.

@CICXPB1 This exec is called by the Started Task Manager as a CICS Pre-Stop Exit. It is used to logon to CICS region in preparation for shutdown. It will set a trap for purge processing, call a CICS region specific Pre-Stop exit, and stop the POVI logon.

@CICXPW1 This exec is called by the Started Task Manager as a CICS Stop-WTOR Exit. It is used if the region does not come down within the stop verification interval to inactivate the ACB defined in the @CICCNTL table. It will call a region specific Stop_WTOR exit. If the region is hung, it will issue an MVS cancel against the CICS region and restart it and then stop the region normally.

@CICXSA1 This exec is called by the Started Task Manager as a CICS Post-Start Exit. It is used to logon for post-start processing, then logoff. It will post CA7 requirement for a job, start a POVI logon, and call a region specific Post-Start exit.

@CICXSB1 This exec is called by the Started Task Manager as a CICS Pre-Start Exit. If specified, it will call a region specific Pre-Start exit.

ISPF User Interface

The CICS ISPF User interface panels are used to define and update information on each CICS region under the control of the CICS Manager. The extended CICS automation is invoked when the Started Task Manager either starts, stops or attempts recovery of a CICS region that has been defined to the CICS Manager.

Region_Name The name of the CICS region to manage. This field is an eight character alphanumeric field that defines the CICS region to the CICS Manager. It must match the resource name of the Started Task Manager task name entry for the CICS region.
**Description**
The region description is a one line, ‘plain English’ description of the region defined by the user. The description is entered when the region is first added to the CICS Manager.

**Alias**
The CICS region Alias field is used to specify an alias for the name of the region. This is a 12 character alphanumeric field, assigned by the user. The alias can be used to identify the region by a name that is easier to remember than its defined regional name, such as, defining an alias of INFO for an Infosys region named CICSxxxx. It can be used to address a CICS region using the IRM command interface.

If there is no alias for the region, the region name can be entered in this field. The region alias field is required.

**Number**
Specifies a unique number for the region starting from 1001. This is a four character numeric field that is used by the CICS Manager to create names for the AF/Operator variables and traps associated with the region.

The region numbers need not be consecutive and are not used in any ordering scheme, but each region must have a unique number assigned to it. Region numbers are automatically assigned and cannot be modified.

**VTAM Appl ID**
This field is used to identify the node name of the CICS region as defined to VTAM. An AF/Operator system or global variable can be coded in this field by prefacing the AF/Operator variable name with an ampersand (i.e. &AOVARNAM). This is an 8 character alphanumeric field.

This field is optional.
CICS Manager Components

**Console Logon**
This field is used to specify whether or not a logon is required to issue CICS commands, including the shutdown command. If security is not turned on, logging on to the region is not required. CEMT commands can be issued directly via the modify command. If security is turned on, the region logon is required to issue CEMT commands.

This is a binary field (0,1). Specify 1 to enable the region logon. Specifying 0 indicates region logon is not required. This is an optional field. If not specified, the default is 0.

**User ID**
Specifies the TSO userid that will be used by AF/OPERATOR to logon to the CICS region. An AF/OPERATOR system or global variable can be coded in this field by prefacing the AF/OPERATOR variable name with an ampersand (i.e., &AOVARNAM).

This field is required if the Console Logon field is enabled.

**Password**
Specifies the logon password for the CICS region.

This field is not displayed. To enter or change the password, tab to this field and type in the password. When the changes are saved, the password verification panel will be displayed.

This field is required if the User ID field is not blank.

*Note:* To clear an existing password, enter a single asterisk in the field, press ENTER, and save the update.

**Reply Wait**
Specifies the amount of time the CICS Manager will wait for a response to a logon request for a CICS region. This field is in military time format and is required. If not specified, the default is 2 minutes: 00:02:00.
Logon Wait
Specifies the amount of time the CICS Manager will wait for a response from the CICS region indicating that the user ID’s password was accepted. This field is in military time format and is required. If not specified, the default is 5 minutes: 00:05:00.

POVI Logon
The POVI Logon field is used to specify if a VTAM logon is to be initiated from AF/OPERATOR to a CICS region. If enabled, a continuous session is available for user automation applications. The logon is automatically established after successful region startup and automatically terminated prior to region shutdown.

This is a binary field (0,1). Specify 1 to enable the logon function. Specify 0 to disable the logon function. This field is optional. If not specified, the default is 0.

POVI ID
Specifies the POVI ID of the CICS region. Used in the POVI logon command to identify the CICS logon session. The POVI ID may also be used in the IRM command interface to address a CICS region.

This is an optional field.

Purge Transactions
The Purge Transactions field is used to specify if hung transactions are to be purged during the shutdown process for the region. If enabled, all non-system transactions with a status of 'Hung' will be purged if the region issues the message indicating shutdown cannot complete due to outstanding transactions.

This is a binary field (0,1). Specify 1 to enable transaction purging. Specify 0 to disable the function. This is an optional field. If not specified, the default is 0.

Cancel Region
The Cancel Region function provides an alternate stop process for a region that will not shutdown. If enabled, as part of the STOP-WTOR exit processing, the region will be cancelled, restarted and stopped.

This is a binary field (0,1). Specify 1 to enable the cancel region feature. Specify 0 to disable the feature. This field is optional. If not specified, the default is 0.
The CA7 Region Up Post feature is designed to provide an interface between the Started Task Manager and CA7 batch job processing, such as batch jobs to open files, start transactions or printers. If the feature is enabled, a CA7 dataset requirement will be posted once the Started Task Manager receives the start verification message for the region. The dataset requirement must in the following form:

`SCS.AFOPER.<regname>.UP.G0000V00.`

This is a binary field (0,1). Specify 1 to enable the feature. Specify 0 to disable the feature. This is an optional field. If not specified, the default is 0.

**Pre-Start Exit**
This field identifies the command file name of the region specific pre-start exit. An AF/Operator system or global variable, which equates to the exit command file name, can be coded in this field by prefacing the AF/Operator variable name with an ampersand (i.e. &AOVARNAM). If the region has no pre-start exit, leave this field blank.

**Post-Start Exit**
This field identifies the command file name of the region specific post-start exit. An AF/Operator system or global variable, which equates to the exit command file name, can be coded in this field by prefacing the AF/Operator variable name with an ampersand (i.e. &AOVARNAM). If the region has no post-start exit, leave this field blank.

**Start-WTOR Exit**
This field identifies the command file name of the region specific start-WTOR exit. An AF/Operator system or global variable, which equates to the exit command file name, can be coded in this field by prefacing the AF/Operator variable name with an ampersand (i.e. &AOVARNAM). If the region has no start-WTOR exit, leave this field blank.
### Pre-Stop Exit
This field identifies the command file name of the region specific pre-stop exit. An AF/Operator system or global variable, which equates to the exit command file name, can be coded in this field by prefacing the AF/Operator variable name with an ampersand (i.e. &AOVARNAM). If the region has no pre-stop exit, leave this field blank.

### Post-Stop Exit
This field identifies the command file name of the region specific post-stop exit. An AF/Operator system or global variable, which equates to the exit command file name, can be coded in this field by prefacing the AF/Operator variable name with an ampersand (i.e. &AOVARNAM). If the region has no post-stop exit, leave this field blank.

### Stop-WTOR Exit
This field identifies the command file name of the region specific stop-WTOR exit. An AF/Operator system or global variable, which equates to the exit command file name, can be coded in this field by prefacing the AF/Operator variable name with an ampersand (i.e. &AOVARNAM). If the region has no stop-WTOR exit, leave this field blank.

## Controlling CICS Manager Resources
Once the automation table component of the CICS Manager has been defined, the information needed to manage the enhanced automation capabilities for the CICS region is available to the Started Task Manager. To initiate the actual management of the CICS regions, the CICS Manager must be activated.

### Initializing the CICS Manager
Ensure the enable and initialize flags are on, set to 1, for the CICS Manager in the Application Manager System Definition for this system. If the flags are turned on, the CICS Manager will be automatically initialized at AF/OPERATOR initialization time. As part of the initialization process, the CICS Manager Control table, @CICCNTL, will be loaded into above-the-line storage.
Using the Service Subroutines

The extended automation available with the CICS Manager is accessed via user exits. Once all the required parameters are defined to the CICS Manager, the exits that control the features can be coded in the proper fields in the Started Task Manager definition for the region.

From the Started Task Manager main menu, update each of the desired CICS regions and enter the desired user exits in the fields indicated:

- Pre-Start Exit @CICXSB1
- Post-Start @CICXSA1
- Pre-Stop Exit @CICXPB1
- Post-Stop @CICXPA1
- Stop-WTOR @CICXPW1

To implement the changes while AF/OPERATOR is running, use the Table Manager REFRESH command to refresh the in-storage copy of the Started Task Manager and CICS Manager tables. This allows the Started Task and CICS Managers access to the new data.
CICS Manager Components
Introduction

The IMS Start Application is used to automate the start of IMS. The application is implemented as a start command procedure executed by the Started Task Manager.

Note: This is a separately billable item and can only be activated after obtaining the proper authentication from Candle Corporation.

Chapter Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS Manager</td>
<td>118</td>
</tr>
<tr>
<td>IMS Start Application</td>
<td>118</td>
</tr>
<tr>
<td>Control Options</td>
<td>123</td>
</tr>
<tr>
<td>IMS Command On-line Help</td>
<td>124</td>
</tr>
</tbody>
</table>
IMS Manager

IMS Start Application

The activities of the application are controlled by information stored in a table created by the user. The table that contains the IMS start information is:

@IMSSTRT  The IMS Start Table

The table is a member of the Automation Application table dataset, hilevel.RKAUTBLR. The table is loaded into above-the-line storage at AF/OPERATOR and Automation Application initialization time.

When an RCF START ALL or RCF START <IMS Jobname> command is received, information is retrieved from this table by the @IMSSTRT service routine and used to determine how to start IMS.

The IMS start procedures vary depending on the IMS start type requested. The possible IMS start types are: AUTO or MANUAL. The IMS Start Application performs the following processes for each of the possible IMS start types

Manual Mode IMS Start

Issues a WTOR to the operator console, which displays the default IMS start parameters, for example:

ASOA401I  IMS Automated start up overridden
ASOA401I  IMS Default start information is:
ASOA401I  Start command =
            S  DBM02CTL,AUTO=Y,MSDBREC=N,HSBID=1
ASOA401I  Start parms  = *
ASOA401I  Start type   = AUTO
ASOA401I  XRF start type = BOTH
ASOA401I  MSDB recovery = N
xx ASOA401A Enter A to Abort, C to Continue start up, or IMS start overrides (PARMS=, TYPE=, XRF=, MSDBREC=)
Three options are available. They are:

- Reply A to abort the automated start of IMS.
  
  **Note:** If the abort option, A, is chosen, IMS will need to be started manually.

- Reply C to continue the automated start of IMS, by AF/OPERATOR, with the default start parameters.

- Reply to the WTOR with override information for the start parms, start type, XRF type or MSDB recovery parameters.

Only one override at a time may be specified to the WTOR. When an override reply is received, the WTOR will be re-displayed with the new information. If more than one override is necessary, continue replying to the WTOR, until all of the necessary overrides are in place. The WTOR will continue to be displayed until either an A or C is received.

The format and reply options are as follows:

**StartParms**

Specify any valid symbolic defined in the IMS proc. This is not a required field. For example, you may enter:

```
R xx,PARMS=SSN=DSN1
```

**StartType**

Specify the IMS start type. This is a required field. The options are:

- **AUTO**
  
  Valid only on XRF Active system. May not be specified for an XRF Alternate system. Default start type unless otherwise specified.

- **BACKUP**
  
  Valid only on XRF Alternate system. No other start type will be accepted for an XRF Alternate system.

- **BUILDQ**
  
  Valid only on XRF Active system. May not be specified for an XRF Alternate system.

- **COLD**
  
  Valid only on XRF Active system. May not be specified for an XRF Alternate system.
**IMS Manager**

**MANUAL**
AF/OPERATOR will not reply to the IMS WTOR with a restart command. The operator must reply to the WTOR with a valid restart option. Valid only on the XRF Active system.

For example, you may enter:

R xx,TYPE=BUILDQ

---

**XRF Type**
Specify the XRF system type. This is a required field. The options are:

**ACT**
Start an XRF Active system only. May not be specified on an XRF Alternate system.

**ALT**
Start an XRF Alternate system only. No other XRF type will be accepted for an XRF Alternate system.

**Note:** Unless the XRF Active system is currently running, the Alternate system will abend.

**BOTH**
Start an XRF complex: Active and Alternate system. Default XRF type unless otherwise specified.

**NONXRF**
Start a non-XRF IMS system. Valid only if this IMS is non-XRF capable.

For example, you may enter:

R xx,XRF=ACT

---

**MSDB Recovery**
Specify the MSDB recovery option. This is a required field. The options are:

**Y**
Perform MSDB recovery.

**N**
Do not perform MSDB recovery. Default MSDB recovery option unless otherwise specified.

**N/A**
MSDB recovery option is not enabled.

For example, you may enter:

R xx,MSDBREC=Y
Once the necessary overrides are in place, reply C to the WTOR to initiate the start of IMS.

**Automated Mode IMS Start**

If the continue option, C, was chosen for the manual mode WTOR or if this is an automated start of IMS, the IMS Start Application will proceed to start IMS as follows:

Determine the XRF type:

- If this is not an XRF system, then proceed with normal IMS start-up.
- If this is an XRF active:
  - Set the default XRF type to BOTH.
  - Validate the HSBID. If the last active HSBID does not match, issue an error message and re-display the start parameters.
  - If the active is already running, abort the start-up.
  - Update the XRF file.
  - Set @APLXRF=ACT.
  - If this is an XRF alternate:
    - If the active system is not running, then abort the start-up.
    - Update the XRF file.
    - Set @APLXRF=ALT
    - Notify the active system that the alternate is available.

Set the traps for IMS start WTORs.

- @I####BQ - Performs IMS BUILDQ Start
- @I####BK - Performs IMS BACKUP Start
- @I####CL - Performs IMS Cold Start

Set the traps for DBRC signon failures.

- @I####EO - Performs IMS ERE OVERRIDE Start
- @I####RT - Sets Restart Terminated Flag

Issue the IMS start command.
Reply with the requested IMS start type.
Check for the IMS start verification message.

**Enabling the Application**

To add a region to the IMS Start Application, add a row to the @IMSSTRT table, identifying the control region name and the SMFID it runs on, then, fill in all of the fields in the table with the appropriate information. Then, specify:

```bash
EXEC @IMSSTRT "jobname option"
```

in the START_CMD field for the IMS resource in the Started Task Manager Task Start Table. The next time AF/OPERATOR is started, the new information will be available to start IMS.

To update information for the IMS Start Application, edit the table and change whatever fields are necessary. The next time AF/OPERATOR is started, the changes will be available.

To implement the changes while AF/OPERATOR is running, use the Table Manager REFRESH command to refresh the in-storage copy of the IMS Start Table, and if necessary, the Started Task Manager Task Start Table. This will give the Started Task Manager and the IMS Start Application access to the new data which will be in effect for the next start of IMS.
Control Options

@IMSSTRT - IMS Start Table

The IMS Start Application control information is stored in the IMS Start Table, @IMSSTRT. This is a keyed table with the keys being: JOBNAME, SMFID, and XRF Type. The service routine, @IMSSTRT, searches this table for the specified configuration, based upon the jobname specified on the RCF command, the SMFID of the system (AOSMFID), and the XRF type (@APLXRF). If no match is found, an error message will be displayed. The following describes the control information held in the table:

**JOBNAME**

The jobname of the IMS control region as specified in the Started Task Manager. This field is the first key in the table.

**SMFID**

The SMFID of the system on which to manage the IMS start. This field is the second key of the table. This is an 8 character alphanumeric field assigned by the user. Any alphanumeric character may be used for the SMFID with the exception of an (*) or (").

**XRF_TYPE**

This is the XRF type of the system. Valid types are: ACT, ALT, or N/A. specify ACT for the XRF Active system, ALT for the XRF Alternate system and N/A for a non-XRF capable system. This field is the last key of the table.

**XRF_HSBID**

This is the HSBID of the system. Valid HSBIDs are 1 or 2. Code an (*) if this is a non-XRF capable system.

**XRF_LINK**

This is the P2P link ID to the partner XRF system. Code an (*) if this is a non-XRF capable system or if a P2P link does not exist.

**START_TYPE**

This is the default IMS start type. Valid start types are: AUTO, BACKUP, BUILDQ, COLD, or MANUAL. This is a required field.

**MSDBREC**

This is the MSDB recovery option. Valid options are: Y or N. Code an (*) in this field if the MSDB recovery option is not enabled.
RCF Command Overrides

The IMS start overrides described for a manual mode start-up, can also be passed to the IMS Start Application as part of a Started Task Manager RCF command for IMS:

- `RCF START IMS PARM(jobname AUTO)`
- `RCF START IMS PARM(jobname MANUAL)`
- `RCF START IMS ADDPARM(MANUAL)`

IMS Command On-line Help

On-line help for the IMS Manager IMS command is available anytime. To display on-line help for the IMS command, enter the following command at the MVS operator console:

- `IMS HELP`

The IMS Manager will then display the IMS command syntax followed by a brief explanation of each of the keywords associated with the command.
Introduction

The XRF Manager is used to automate the IMS/XRF takeover process. Its main purpose is to automate the UNLOCK reply and perform workload movement following an XRF takeover. It performs these tasks through a set of message traps, which detect XRF takeovers.

The XRF Manager does not make any takeover decisions itself, but rather waits for IMS to begin the takeover process, before taking any action. Once the takeover has begun, the XRF Manager monitors the status of both the failing active and new active systems, through the use of a shared DASD PDS, which the XRF Manager maintains on each system.

From information stored in the PDS, the XRF Manager can determine if I/O prevention has successfully completed on the failing system, and thereby complete the UNLOCK of the new active. If, however, the XRF Manager is unable to determine the status of the failing system within 2 minutes, AF/Remote is called to reset the failing LPAR. Upon successful completion of the system reset, the UNLOCK is performed on the new active.

After completion of the XRF takeover, the XRF Manager performs the tasks necessary to relocate the XRF workload from the failed system to the new active. This workload movement includes such tasks as DB2, tape drives, and the start-up of a new alternate.

Note: This is a separately billable item and can only be activated after obtaining the proper authentication from Candle Corporation.
## Chapter Contents

- XRF Manager Components .............................................. 127
- Automation Application Tables ....................................... 127
- Service Subroutines .................................................... 128
- AF/OPERATOR System Variables ...................................... 129
- Automation Tables ....................................................... 130
- Controlling XRF Manager Resources ................................. 132
  - Starting the Service Subroutines .................................. 132
  - Using the Service Subroutines ..................................... 133
  - XRF Command Syntax ................................................. 133
- Functions ................................................................. 135
- XRF Command On-line Help ........................................... 135
XRF Manager Components

The XRF Manager has three components. They are:

- Automation Application tables
- Service subroutines
- AF/OPERATOR system variables.

Automation Application Tables

To manage the XRF takeover process, the XRF Manager uses information stored in a table created by the user. The table that contains the XRF information is @XRFCNTL The XRF Manager Control Table.

This table is a member of the Automation Application table dataset. The table is loaded into above-the-line storage at AF/OPERATOR and Automation Application initialization time. When the XRF Manager system is initialized, the appropriate information is retrieved from the table and used to set the XRF message traps and allocate the XRF status file.

To add an XRF system to the XRF Manager, add a row to the XRF Manager Control table and fill in all of the fields in the table with the appropriate information. The next time AF/OPERATOR is started, the new information will be available to the XRF Manager.

To update information for an XRF system, edit the XRF Manager Control table and change whatever fields are necessary. The next time AF/OPERATOR is started the changes will be available to the XRF Manager.

To implement the changes while AF/OPERATOR is running, issue an APL INSTALL XRF command which will refresh the in-storage copy of the XRF Manager Control table and re-initialize the XRF Manager.
**Service Subroutines**

The XRF Manager service subroutines are AF/OPERATOR command files which perform the services needed to manage XRF systems. The XRF Manager service subroutines are comprised of the following execs:

- **@XRFINIT**
  This exec initializes the XRF Manager environment and creates an MVS console command, called XRF, that allows you to display information about a single XRF system or group of systems.

- **@XRFMAIN**
  This exec processes XRF Manager commands issued to manage XRF systems.
  @XRFMAIN is responsible for: validating and processing command text, determining if the specified SMFID is defined as an XRF system, responding to operator inquiries, and activating or inactivating XRF automation.

- **@XRFHELP**
  This exec provides the XRF Manager on-line help facility.

- **@XRFCIMS**
  This exec issues an RCF KILL for the IMS control region during a takeover.

- **@XRFMGRP**
  This exec performs workload movement after an XRF takeover.

- **@XRFNALT**
  This exec performs start-up of the new alternate system after an XRF takeover.

- **@XRFPTKO**
  This exec performs post-takeover functions.

- **@XRFP2PC**
  This exec provides support for IMS message region mirroring.

- **@XRFTKOC**
  This exec indicates that takeover has completed.

- **@XRFREAD**
  This exec reads data from the XRF file.

- **@XRFUNLK**
  This exec replies to the UNLOCK WTOR. It monitors the XRF file to determine if I/O prevention has completed. It also calls AF/Remote, if necessary, to reset the failing system's LPAR.

- **@XRFWRIT**
  This exec writes data to the XRF file.

- **@XRFXPA1**
  This exec is the XRF control region post-stop exit.

- **@XRFXPA4**
  This exec is called by @RCFDRVR as an IMS Database Post-stop Exit for XRF. It is used to verify IMS database de-allocate message is received from the XRF alternate system.
XRF Manager Components

@XRFXPB1 This exec is the XRF control region pre-stop exit.

@XRFXPB4 This exec is called by @RCFDRVR as an IMS Database Pre-stop Exit for XRF. It is used to set a message trap to capture IMS database de-allocate message from the XRF alternate system.

@XRFXRB1 This exec is the XRF control region pre-recovery exit.

@XRFXRB2 This exec is the XRF message regions pre-recovery exit.

@XRFXRB3 This exec is called by @RCFDRVR as an IMS Message Region Pre-recovery Exit. It is used to bypass recovery of the IMS message regions if IMS/XRF takeover is in-progress.

@XRFXSA1 This exec is the XRF control region post-start exit.

AF/OPERATOR System Variables

The XRF Manager creates the following AF/OPERATOR system variables:

@XRFALT XRF Alternate System Status Flag
@XRFDBRC Name of DBRC Region
@XRFDB2 Name of DB2 Region
@XRFDD0 Number of members in XRF file
@XRFDD1 Member name of IMS on this system
@XRFDD2 Member name of IMS on the other system
@XRFDD3 Member name of CPU on this system
@XRFDD4 Member name of CPU on the other system
@XRFDD5 Member name of the last active HSBID
@XRFDLI Name of DLI Region
@XRFFILE Dataset Name of XRF File
@XRFHSB HSBID of this system
@XRFIMS Name of IMS Control Region
@XRFLINK P2P link ID to the other system
@XRFNALT New Alternate Start Time
The Application Manager creates the following XRF system variable:

**@XRFSALT** Alternate Start Option Flag

The Application Manager creates the following XRF system variable:

**@APLXRF** Specifies the type of IMS/XRF system. Valid values are:
- **ACT** - This is an IMS/XRF Active system.
- **ALT** - This is an IMS/XRF Alternate system.
- **N/A** - This is not an IMS/XRF system.

**Automation Tables**

**@XRFCNTL - XRF Manager Control Table**

The XRF Manager Control table is used to store information about the XRF systems. The table is keyed by SMFID, therefore, there must be one unique entry per system per IMS Control Region.

- **HSBID** — This field identifies the XRF HSBID. Valid values are: 1 or 2. This is a required field.
- **CTL_JOB** — This field is the jobname of the IMS control region. This is a required field.
- **DBRC_JOB** — This field is the jobname of the DBRC region. This is a required field.
- **DL1_JOB** — This field is the jobname of the DL1 region. This is a required field.
- **DB2_JOB** — This field is the jobname of the DB2 region. This is an optional field. Code an asterisk (*) in this field if there is no DB2 associated with this IMS.
- **DB2_IRLM** — This field is the jobname of the IRMLM address space. Code an asterisk (*) in this field if there is no DB2 associated with this IMS.
- **NEW_ALT** — This field is the re-start delay interval for the new alternate. Specify in the form HH:MM:SS and within the range 00:00:01 to 24:00:00. Default is 00:15:00.
**XRF Manager Components**

**INT_MGRP**
This field is the amount of time IRM will wait for a response from a command issued to another system through Peer to Peer services. Specify in the form hh:mm:ss and within the range 00:00:01 to 24:00:00. Default is 00:15:00.

**P2P_ID**
This field is the P2P link ID to the partner XRF system. Code an asterisk (*) if this is a non-XRF capable system or if a P2P link does not exist between the XRF systems.

**MEMBER.0**
This field is the number of members in the XRF file. This is a required field.

**MEMBER.1**
This field is the member name for this IMS system. This should be the same as the IMS ACB name (i.e. S21IMS). This is a required field.

**MEMBER.2**
This field is the member name for the other IMS system. This should be the same as the IMS ACB name (i.e. S06IMS). This is a required field.

**MEMBER.3**
This field is the member name for this CPU. This should be the same as the IMS ACB name, except that the last 3 characters should be CPU (i.e. S21CPU). This is a required field.

**MEMBER.4**
This field is the member name for the other CPU. This should be the same as the IMS ACB name, except that the last 3 characters should be CPU (i.e. S06CPU). This is a required field.

**MEMBER.5**
This field is the member name for the last active HSBID. This must be specified as LAHSBID.

**XRF_FILE**
This field is the dataset name of the XRF PDS. This is a required field.

**MSG_LIST**
This field is the STM List name for all IMS message regions and IMS related tasks. The list name is used for start and stop processing. Code an asterisk (*) in this field if there is no STM list.

**NOW_LIST**
This field is the name of the IMS list of tasks as defined to STM that will be associated with the move from one system to another. Code an asterisk (*) in this field if there is no RCF list.
Controlling XRF Manager Resources

Once the automation table component of the XRF Manager has been defined, the information needed to manage XRF systems for an environment is available to the XRF Manager service subroutines.

To initiate the actual management of XRF systems, the XRF Manager service subroutines must be activated.

Starting the Service Subroutines

The XRF Manager is automatically initialized at AF/OPERATOR initialization time by the Application Manager which executes the @XRFINIT exec. This command file initializes the XRF Manager environment and creates the XRF console command.

If AF/OPERATOR is already running, you can activate the XRF Manager before the next AF/OPERATOR recycle by issuing the following Application Manager commands from the MVS console:

**APL INSTALL XRF**
Using the Service Subroutines

The XRF command provides an operator interface to give you information about the XRF systems defined to the XRF Manager.

For example, from the MVS operator console you can enter:

**XRF DISPLAY ***

at any time, to get the current status of the XRF complex.

XRF Command Syntax

**FIGURE 5. XRF Command Syntax**

<table>
<thead>
<tr>
<th>XRF function</th>
<th>dd_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELP</td>
<td></td>
</tr>
<tr>
<td>DISPLAY *</td>
<td>Defaults to all members</td>
</tr>
<tr>
<td>ddname</td>
<td>Specific member name</td>
</tr>
</tbody>
</table>

XRF Status Indicators

- **STARTACT** - The active system is starting
- **STARTALT** - The alternate system is starting
- **ACTIVE** - The active system is running
- **BACKUP** - The alternate system is running
- **TAKE0A** - Takeover in progress - Old Act
- **TAKE0B** - Takeover in progress - New Act
- **TAKE0C** - Takeover complete - New Act
- **IOPREV** - I/O prevention in progress - Old Act
- **IOT** - I/O toleration in effect - New Act
### XRF function dd_name

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACT-IOT</td>
<td>TKO complete/IOT in effect</td>
</tr>
<tr>
<td>UNLOCK</td>
<td>Unlock complete/TKO in progress</td>
</tr>
<tr>
<td>IOFREE</td>
<td>I/O prevention complete</td>
</tr>
<tr>
<td>FAILED</td>
<td>IMS has abended</td>
</tr>
<tr>
<td>RESET</td>
<td>System reset in progress</td>
</tr>
<tr>
<td>RESETOK</td>
<td>System reset complete</td>
</tr>
<tr>
<td>RESETFAIL</td>
<td>System reset failed</td>
</tr>
<tr>
<td>WAITALT</td>
<td>Active is waiting for Alternate shutdown</td>
</tr>
<tr>
<td>STOPACT</td>
<td>The active system is stopping</td>
</tr>
<tr>
<td>STOPALT</td>
<td>The alternate system is stopping</td>
</tr>
<tr>
<td>DOWN</td>
<td>IMS is down</td>
</tr>
</tbody>
</table>

### XRF Takeover Reason Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>VTAM Termination</td>
</tr>
<tr>
<td>70</td>
<td>Surveillance Timeout (RDS, LOG, LNK)</td>
</tr>
<tr>
<td>08</td>
<td>/SWITCH Command</td>
</tr>
<tr>
<td>04</td>
<td>IRLM Failure</td>
</tr>
<tr>
<td>02</td>
<td>ESTAE exit entered</td>
</tr>
</tbody>
</table>
Functions

The function parameter is used to tell the XRF Manager which function to perform. Many, but not all, functions have abbreviations. Acceptable abbreviations are noted with the function's description.

**HELP**
Displays the XRF Manager on-line help file on the operator console. No abbreviations of the HELP keyword are acceptable.

**DISPLAY**
Issues a WTO to the MVS console that displays the contents of the specified XRF file member. The default display is all members. Valid abbreviations are: DIS and D.

XRF Command On-line Help

On-line help for the XRF Manager XRF command is available anytime. To display on-line help for the XRF command, enter the following command at the MVS operator console:

**XRF HELP**

The XRF Manager will then display the XRF command syntax followed by a brief explanation of each of the keywords associated with the command.
Introduction

Candle Corporation offers a comprehensive maintenance and support plan to ensure you realize the greatest value possible from your Candle software investments. We have more than 200 technicians worldwide, committed to providing you with prompt resolutions to your support requests.

Customer Support hours of operation are from 5:30 A.M. – 5:00 P.M., Pacific Time. In the event of an after-hours or weekend emergency, Candle’s computerized call management system ensures that a technician will return your call within one hour. For customers located outside of North America, after-hours and weekend support is provided by Candle Customer Support locations in the United States.

Electronic support

Candle provides information and support services through

- Candle’s home page at [www.candle.com](http://www.candle.com). You can use the Candle Worldwide Web Site to
  - open problem records
  - access maintenance information
  - order products or maintenance
  - access IBM compatibility information
  - download fix packs for distributed products
  - read news and alerts
  - scan a list of scheduled Candle education classes
- Candle Electronic Customer Support (CECS), an electronic customer support facility. You can access this facility using the IBM Global Network. You can use CECS to:
Both CECS and the Candle Worldwide Web Site are available 24 hours a day, 7 days per week.

**Telephone support**

Our support network consists of product specialists who work with you to solve your problem.

Candle uses an on-line problem management system to log and track all support requests. Your support request is immediately routed to the appropriate technical resource.

When you call to report a problem, please have the following information:

- your Candle personal ID (PID) number
- the release level of the Candle product
- the release level of IBM or other vendor software
- identifying information and dates of recently applied maintenance to your Candle product or IBM product
- a detailed description of the problem (including the error message) and the events preceding the problem
- a description of any unusual events that occurred before the problem

**Customer support locations and numbers**

To contact a Customer Support representative, refer to the following list. While these phone numbers were accurate at the time this document was published, the current numbers can be found on the Candle Web site, [www.candle.com](http://www.candle.com), under Customer Support.
<table>
<thead>
<tr>
<th>Office</th>
<th>Telephone</th>
<th>FAX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>North America</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(800) 328-1811</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(310) 535-3636</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(310) 727-4204</td>
<td></td>
</tr>
<tr>
<td><strong>Europe</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Belgium/Luxembourg</td>
<td>+32 (0) 3 270 95 60</td>
<td>+32 (0) 3 270 95 41</td>
</tr>
<tr>
<td>France</td>
<td>+33 (0) 1 53 61 60 60</td>
<td>+33 (0) 1 53 61 06 16</td>
</tr>
<tr>
<td>Germany/Switzerland/Austria</td>
<td>+49 (0) 89 54 554 333</td>
<td>+49 (0) 89 54 554 170</td>
</tr>
<tr>
<td>Italy – Freephone</td>
<td>800 780992</td>
<td></td>
</tr>
<tr>
<td>Netherlands</td>
<td>+31 (0) 30 600 35 50</td>
<td>+31 (0) 30 600 35 10</td>
</tr>
<tr>
<td>Scandinavia</td>
<td>+46 (0)8 444 5940</td>
<td>+46 (0)8 623 1855</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>+44 (0)161 437 5224</td>
<td>+44 (0)161 437 5225</td>
</tr>
<tr>
<td>(Southern Europe, Middle East and South Africa Agents call United Kingdom)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Asia Pacific – English Hub</strong></td>
<td></td>
<td>+61 2 9954 1818</td>
</tr>
<tr>
<td>Australia</td>
<td>+61 2 8912 9898</td>
<td></td>
</tr>
<tr>
<td>Hong Kong</td>
<td>800 908 457</td>
<td></td>
</tr>
<tr>
<td>India</td>
<td>+61 2 8912 9898</td>
<td></td>
</tr>
<tr>
<td>Indonesia</td>
<td>0018 03061 2061</td>
<td></td>
</tr>
<tr>
<td>Malaysia</td>
<td>1800 803 459</td>
<td></td>
</tr>
<tr>
<td>New Zealand</td>
<td>0800 449 596</td>
<td></td>
</tr>
<tr>
<td>Philippines</td>
<td>1800 1612 0096</td>
<td></td>
</tr>
<tr>
<td>Singapore</td>
<td>800 616 2075</td>
<td></td>
</tr>
<tr>
<td>Thailand</td>
<td>0018 00612 1045</td>
<td></td>
</tr>
<tr>
<td><strong>Asia Pacific – Japanese Hub</strong></td>
<td>+81 3 3595 7150</td>
<td>+81 3 3595 7110</td>
</tr>
<tr>
<td><strong>Asia Pacific – Korean Hub</strong></td>
<td>+82 2 552 8744</td>
<td>+82 2 552 8746</td>
</tr>
<tr>
<td><strong>Asia Pacific – Mandarin Hub</strong></td>
<td>+88 62 2739 3223</td>
<td>+88 62 2378 5993</td>
</tr>
<tr>
<td><strong>Asia Pacific – e-mail address: <a href="mailto:ap_support@candle.com">ap_support@candle.com</a></strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
When your local support office is unavailable, you can contact Candle’s North America support center. If USADirect® service is available in your country, use the 800 telephone number. If USADirect® service is not available, ask your international operator for assistance in calling Candle’s local (310) number.

**Incident documentation**
You may be asked to send incident documentation to the Candle Customer Support Center. On the outside of all packages you send, please write the incident number given to you by the Customer Support representative.

Send tapes containing the incident information to the following address, unless directed otherwise by your Customer Support representative:

*Candle Customer Support*  
Candle Support Center, *Incident number*  
201 North Douglas Street  
El Segundo, California 90245

Send all other relevant documentation, such as diskettes or paper documentation, to the address provided by your Customer Support representative.

**Ensuring your satisfaction with customer support**
Candle Customer Support is committed to achieving high customer satisfaction ratings in all areas. These include

- connecting you to a support representative promptly
- providing you with the appropriate fixes
- answering support questions
- filling your shipping orders
- supplying documentation

If you have a concern that has not been resolved to your satisfaction, you can open a complaint ticket. All tickets are logged and tracked to ensure responsiveness and closure. Using the ticket information, a manager will contact you promptly to resolve your problem.
Symbols

@IMSSTRT  123
@JESCLAS  74
@JESCNTL  73
@JESDMSC  78
@JESDNJE  79
@JESFCIS  80
@JESGRUP  81
@JESQPRG  84
@OM2CNTL  44
@XRFCNTL  130

A

About this Document  9
Adobe Portable Document Format  10
AF/REMOTE Manager  19
AO System Variables
  JES Manager  72
  XRF Manager  129

C

Candle Electronic Customer Support (CECS)  137
Candle home page  137
Candle Internet site  137
Candle Worldwide Web site  137
CICS
  Controlling Resources  114
  Manager Components  107
  Service Subroutines  107
Command Syntax
  JES  85
  JES Manager  85
  XRF  133
Communication Manager  55
  Components  57
customer support
  electronic support  137
  Internet  137
  locations  138
  numbers  138
  telephone support  138

D

Documentation
  supplied format  10
  Documentation Conventions  12
  Documentation Set  14

H

home page
  Candle  137

I

IMS
  Automated Mode Start  121
  Command Overrides  124
  Control Options  123
  Manual Mode Start  118
  MSDB Recovery  120
  Start Application  118
  Start Parms  119
  Start Type  119
  XRF Type  120
Internet site
  Candle  137
  ISPF User Interface  109

J

JES
  Automation Applications  70
  Automation Table Component  73
  Command Syntax  85
  Controlling Resources  85
Index

Functions  85
Manager Components  70
Object  87
Options  88
Service Subroutines  70
Standard Options  90
JES Manager  69

O
OMEGAMON Logon Manager  43
Online Documentation  14
On-line Help
    IMS Command  124
    JES Command  28, 91, 104
    P2P Command  41
    XRF Command  135

P
Panels and figures  12
Peer-to-Peer Manager  29
POVI Session Manager  49
Prerequisites  9
Printed documentation  14
Printing problems  10

S
Symbols  13

T
TSO Manager  99
typographical conventions  12

V
Variables and literals  12

W
Worldwide Web site
    Candle  137

X
XRF
    Automation Application Tables  127
    Automation Tables  130
    Command Syntax  133
    Controlling Manager Resources  132
    Functions  135
    Manager Components  127
    Service Subroutines  128, 132