

CA Automation Point

Administrator Guide

Release 11.4



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2012 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Automation Point
- CA Directory
- CA FAQS® for Automated System Operation for z/VSE (CA FAQS ASO for z/VSE)
- CA NSM
- CA OPS/MVS® Event Management and Automation (CA OPS/MVS)
- CA Remote Console™

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

Note: In PDF format, page references identify the first page of the topic in which a change was made. The actual change may appear on a later page.

- Updated the [Security Capabilities](#) (see page 115) section with single sign-on.
- Updated [Global and Local Session Settings](#) (see page 116) section.
- Updated [CA AP Remote Manager and Windows Security](#) (see page 116) section.
- Added the [Single Sign-on](#) (see page 120) section.
- Added the [Primary Control Mode](#) (see page 121) section.
- Updated the [Starting Remote Operations](#) (see page 124) with single sign-on.
- Updated the [Enabling Web MV](#) (see page 133) section.
- Updated the [Web MV Dialog](#) (see page 133) section.
- Updated the [Web MV Security](#) (see page 134) section.
- Updated the [Types of PPQs](#) (see page 201) section.
- Updated the [Using the TCP/IP Transport](#) (see page 207) section.
- Updated [Notification Website Application](#) (see page 266) sections.
- Updated the [Notification Manager Database Maintenance Commands](#) (see page 238) section.
- Updated the [Sample Code](#) (see page 297) section.
- Added the [Interacting with External Events Systems](#) (see page 351) sections.
- Updated the [Basic Privileges Mode](#) (see page 313) section.
- Added the [Define the RequestService Java API rootdir Property](#) (see page 353) section.

Contents

Chapter 1: Introduction	17
Notation Conventions	17
Chapter 2: Configuring CA Automation Point	19
Component Privileges	19
Requirement 1	20
Requirement 2	20
Configuration Manager	23
Wizard Interface.....	24
Expert Interface	25
Managing Your Site Configuration	28
File Directory Structure	28
How CA Automation Point Looks for Files	30
Managing Site Files.....	31
Export and Import Operations	32
Using Configuration Manager to Set Up Sessions	37
Creating a Session Definition Set	38
Activating a Session Definition Set	38
Enabling a Session or Function Window.....	38
Creating AXCREXX Sessions.....	39
Creating Windows Command Prompt Sessions	39
How You Create External Event Monitoring Sessions	40
Related Configuration Tasks	40
Chapter 3: Establishing Host Sessions	41
Communicating with the Mainframe	41
Communicating With Mainframe Host Using TN3270	41
Communicating with Midrange Host Using TN5250	42
Configuring SSL for CA Automation Point.....	42
Monitoring Connection Status.....	42
Automating the IPL Process for z/OS	43
Processor Consoles.....	43
Setting Up IPL Automation	44
Synchronizing Date and Time with the Mainframe Host.....	45
Setting the Local Server Clock.....	45
Establishing Cooperative Processing with Mainframe Host Automation	46

General Guidelines.....	46
Understanding Console Management.....	46
SYSPLEX, MCS, and RCS Consoles.....	47
JES3 Consoles.....	51
VSE Consoles.....	53
z/VM Consoles.....	55
z/OS Guest Consoles.....	58
MCS Session Automation and the OSA-ICC DHD Option.....	60

Chapter 4: Asynchronous Host Sessions 61

Asynchronous Sessions.....	61
Hardware Connections.....	61
How You Test Asynchronous Communications.....	62
How You Verify Message Processing.....	64
Asynchronous Telnet Sessions.....	64
How You Set Up Asynchronous Telnet Sessions.....	64
How You Test the Session.....	64
Asynchronous SSH Sessions.....	65
Security Features of the SSH-2 Protocol Implementation.....	65
Set Up Asynchronous SSH Sessions.....	66
Test the Session.....	67
Send Key Operations to Asynchronous Sessions.....	67
Manage Asynchronous Communication.....	67
The Status Line.....	68
Communication Problems.....	72
Asynchronous Console Management.....	72
Readlog Technique.....	74

Chapter 5: Managing Sessions Using CA Automation Point Windows 75

Using the Automation Point Desktop.....	75
Layouts.....	76
Understanding Function Windows.....	78
Window Menu Options.....	79
Selecting a Window.....	80
Closing a Window.....	80
Changing the Size of a Window.....	81
Displaying Multiple Windows.....	81
Moving Displayed Windows.....	81
Scrolling Information in Windows.....	82
Customizing Menus.....	82
Understanding the Menu System.....	83

Selecting Menus for the Function Windows	83
Selecting Menus for Session Windows	84
Menu Control Statements	85
Adding New Menu Options to Existing Submenus	90
Adding New Submenus to Existing Menus	92
Defining a New Menu	93
Customizing Screen Fonts	94
Understanding Screen Fonts	94
Changing the Font List on the Local Machine	95
Changing the Font List for the Remote Workstation	97
Issuing Commands	98
Command Dialog	98
Command Area	99
The Command Area State	99
Issuing a Console Command	100
Executing REXX Programs or Scripts	100
Stopping a REXX Program	101
Recalling Previously Entered Commands	101
Displaying Recalled Messages	101
Windows for Viewing Messages	102
Marking a Place in the List	102
Deleting a Recalled Message	103
Browsing Recalled Messages	103
Managing a Recalled Message List	103
Displaying Recent CA Automation Point Messages	103
Merging Messages from Managed Sessions	104
Displaying Log File Contents	105
Scrolling Through Log Contents	105
Displaying Notification Server and Notification Manager Messages	106
Displaying Diagnostic Information	106
Displaying Trace Information	106
Getting Information About CA Automation Point Messages	106
Displaying Graphs of System Information	106
Stopping CA Automation Point	107

Chapter 6: Viewing Remote Sessions 109

Overview	109
Remote Viewer	110
Web Message Viewer	112
Remote Viewer	112
Enabling CA Automation Point for Remote Viewing	112

Securing the Remote Viewer	114
Starting Remote Operations	124
Web Message Viewer.....	128
What is Web MV?	129
How Does Web MV Work?	130
Installing Web MV	132
Enabling Web MV.....	133
Web MV Security	134
Connecting to Web MV	135
Web MV Main Window	136
Viewing Messages with Web MV	139
The Built-in ALL Option.....	139
Messages Columns.....	140
Configuring Visual Attributes.....	142
Choosing the Right Remote Viewing Tool	145

Chapter 7: Writing Rules 149

Understanding the Rules Language.....	149
Types of Rules	149
Ways to Use Rules.....	150
Rules Keyword Summary	150
How CA Automation Point Processes Rules Keywords.....	155
Creating a Rules File	156
Using Variables in Rules.....	157
Environmental Variables.....	157
Status Variables.....	162
Specifying Text Strings in Rules.....	167
Specifying Character Strings	167
Enclosing Character Strings in Quotes.....	168
Writing Time Rules.....	168
Evaluating Time Rules.....	169
Evaluating Time Rules During Time Changes	170
Writing Message Rules.....	176
Writing Command Rules.....	176
Enabling a Rules File.....	178
Editing a Rules File.....	178
Replacing the Current Rules File Dynamically.....	179

Chapter 8: Configuring and Writing REXX 181

Using REXX Programs for Automation.....	181
Supplied Applications	182

Configuring Open Object REXX	183
Invoking REXX Programs.....	183
Accessing Various Command Environments Through REXX	184
Command Environments	184
Using the REXX ADDRESS Statement.....	185
How to Issue Command Processors	186
Issuing Command Processors from REXX Programs	186
Issuing Command Processors From Rules	187
Issuing Command Processors From Menus	187
Designing Portable REXX Programs.....	187
How Do CA Automation Point and REXX Communicate?	188
Understanding REXX Statement Processing	189
Referencing a Status Variable From REXX	189
Preventing REXX From Parsing a Statement Incorrectly.....	190
Understanding Global Variables	190
Understanding the Global Variable Environment	190
What are Volatile Variables?	191
What are Nonvolatile Variables?	191
What Is a Variable Group?.....	192
Naming GLV Variables	192
Assigning Values.....	193
Using the Global Variable Environment.....	193
Addressing the Global Variable Environment.....	194
Setting the GLV Path.....	194
Issuing GLV Commands	194
GLV Command Summary.....	194
GLV Return Code Variable RCs.....	196
Testing REXX Programs.....	196
Include a DEBUG Parameter in Your REXX Program.....	196
Run the REXX Program Outside CA Automation Point	197
View Multiple Windows During Debugging	198
Test Message-driven REXX Programs Locally	198
Binding a REXX Program to a CA Automation Point Session.....	198
How Does Binding Work in CA Automation Point?	199
Binding a Session.....	200
Unbinding a Session	200

Chapter 9: Using Program-to-Program Queues 201

Understanding PPQs	201
Types of PPQs	201
General Limitations	202

Memory Requirements	202
Design Guidelines.....	203
How to Issue PPQ Commands.....	204
Issuing PPQ Commands from REXX Programs.....	204
Issuing PPQ Commands from Rules.....	205
Issuing PPQ Commands from CA OPS/MVS	206
Configuring PPQs	206
Using the TCP/IP Transport.....	207
PPQSTAT.bat Program	207

Chapter 10: Using Notification Services 209

Introducing the VOX Environment.....	209
VOX Environment Capabilities	210
VOX Environment Features.....	211
Understanding the VOX Environment's Components	213
How the VOX Environment Components Work Together	214
Paging Capabilities	215
Batching Page Requests.....	216
Configuring Notification Services.....	217
Notification Server Options	217
VOX Client Options.....	218
Configuring the Dialogic Voice Card.....	218
Installing the Dialogic Software.....	219
Configuring Your Workstation for Voice Processing.....	220
Configuring Call Progress Analysis (CPA) Parameters	220
Inserting a New Named Set of CPA Parameters.....	221
Changing Individual CPA Parameters	222
Configuring Channel Groups.....	223
Configuring Voice Channels.....	224
The Voice Word Library.....	225
The Voxmaint Application.....	225
Configuring Answer Tree	226
Sample Answer Tree Configuration.....	228
Accessing Answer Tree	229
Configuring an Answer Tree Group.....	229
Configuring the Email Feature	231
Configuring the Text-to-Speech Feature	232
Configuring the AP Server for Text-to-Speech	233
Configuring a Text-to-Speech Client.....	233
Troubleshooting Text-to-Speech Notification.....	234
Two-Way Paging	235

Configure Two-Way Paging.....	236
The VOX Command Environment	237
What Is a VOX Command?.....	237
VOX Command Summary	238
Example VOX Environment Applications.....	244
Outbound Voice Notification with Input from Caller	245
Outbound Recording with Voice Message Delivery	249
Inbound Application with Input from Caller	255

Chapter 11: Using Notification Manager 261

Notification Manager Overview.....	261
Notification Strategy	262
Objects Used by Notification Manager.....	262
Notification Manager Scenarios.....	263
Notification Manager Database	265
Configuration Steps.....	265
Notification Website Application	266
Install the Notification Website Application on the Same Machine as the CA Automation Point Server	266
Install the Notification Website Application on a Non-CA Automation Point Machine	268
Optional Configuration Tasks.....	268
How You Create Or Update a Notification Manager Database on a Non-CA Automation Point Server.....	268
Configure Voice Notification and Call-in Features for Notification	269
Notification Website Search Capabilities.....	271
Security Options for Notification Manager.....	272
Notification Manager Concepts	275
Backup Methods	276
Overrides	277
Order of Events When Using Backups and Overrides.....	277
Forwarding.....	278
Availability	280
Groups	280
Broadcasting	281
REXX Programs.....	283
Basic Rules for Creating a Personal Schedule	293
Creating Your Own Invocation Programs	297
Sample Code	297
Architected Parameters.....	298
Method-specific Parameters.....	299
Runtime Parameters	300
Emergency Mode Processing.....	301
How You Populate the Notification Manager Database.....	304

How You Use the Notification Website	304
Connecting to the Notification Website	305
Overview of the Notification Website	305
Updating Notification Policies.....	307
Troubleshooting the Notification Website	307
Notifications Not Updated.....	307
Methods Not Updated.....	307
Notification Failure.....	308
REXX Program Not Included in Invocation List.....	308
Login Message.....	308
RC=6040 Error	309
Secure the Notification Website	309
User Authentication	310
User Authorization	312
Understanding Notification Manager Log Files.....	327
VOXNM.LOG	327
WEBNM.LOG	328
CAAPNFY.LOG	329
JolBeep Panel Emulation	330

Chapter 12: Managing CMOS Processors 333

CMOS Processors	333
Hardware Management Console (HMC).....	333
Service Elements (SE)	334
Communicating with CMOS Processors	334
Establishing Connectivity with the HMC/SE.....	336
Establishing Runtime Environment for HMC Console API	337
Using CA Automation Point to Manage CMOS Processors	338
APCMOSI REXX Program.....	339
Submitting HMC commands with APCMOSI REXX Program	339
APCMOSI Command Syntax.....	340
Hardware Access Facility (HAF).....	343
Configuring HAF	343
Monitoring HMC Messages with HAF.....	344
HAF Message Format	344

Chapter 13: Interacting with External Event Systems 351

Deploy and Configure CA Automation Point Web Services	351
Configure Web Services.....	351
Deploy the RequestService Client	352
Define the RequestService Java API rootdir Property	353

Configure Web Service Security	354
Create TLS certificates	355
Configure Tomcat for TLS	357
Configure a remote web service client for TLS.....	358
Using the CA NSM Interface	359
Understanding the ADDRESS TNG Environment	359
Configuring CA Automation Point CA NSM Objects Using Configuration Manager	364
Understanding the Event Traffic Controller	365
Communication Between Event Traffic Controller and Event Manager	365
Configuring the Event Traffic Controller.....	374
Troubleshooting the Event Traffic Controller	380
Using CA Automation Point to Monitor Windows Event Logs.....	388

Chapter 14: Using the CA OPS/MVS Interface 389

Sending Data from CA Automation Point to CA OPS/MVS	390
Sending Data From CA OPS/MVS to CA Automation Point.....	391
ADDRESS AP Host Command Environment.....	391
ADDRESS WTO Host Command Environment.....	391
Configuring the CA OPS/MVS Interface	392
Setting Up Communications to CA OPS/MVS	392
Defining CA OPS/MVS Nodes.....	397
Defining CA Automation Point Systems to CA OPS/MVS.....	398
Recycling the CA OPS/MVS Communications Interface	398
Command-level Security.....	399

Chapter 15: Using AS/400 Manager 401

What Is AS/400 Manager.....	401
AS/400 Manager Application Commands.....	401
AS/400 Manager Terminology	402
Connecting CA Automation Point and AS/400.....	402
AS/400 System Requirements.....	403
Installing and Configuring AS/400 Manager	403
Setting Up AS/400 for Communication with AS/400 Manager	403
Setting Up Your Workstation for Communication with AS/400.....	404
Customizing CA Automation Point Files to Install AS/400 Manager	405
Configuring AS/400 Manager.....	406
Creating a User Profile for CA Automation Point	407
Creating the 3270 Keyboard Map Routine in AS/400.....	409
Setting Up the Alert Function in AS/400.....	409
Understanding Configuration Settings	411
Managing Multiple AS/400 Systems	414

Choosing a Method for Connecting to Multiple Systems	414
Setting Up AS/400 to Connect to Multiple Stand-alone Systems.....	414
Accessing the New Systems.....	418
Test Starting AS/400 Manager	418
Start CA Automation Point.....	418
Designing Automation for AS/400 Manager.....	418
Defining Messages as Alerts in AS/400.....	419
Writing CA Automation Point Rules	419
Writing Automation REXX Programs	424
Operating AS/400 Manager.....	425
SUSPEND and RESUME	426
Displaying System and Configuration Data.....	427
Sending OS/400 Operator Line Commands	428
Unbinding Sessions.....	429
AS/400 Manual Session	429
Technical Information	429
Summary of the REXX Programs	430
Automation Session Programs.....	435
AS/400 Default Logon Screen	440

Chapter 16: Using the Plot Feature 441

Understanding the Plot Feature	441
Plotting a Graph	441
Summary of Steps for Plotting a Graph	441
Understanding the Elements of a Graph	442
Using the Plot Feature with REXX	443
Designing a Basic Graph	443
PLOT DEFINE GRAPH Command Options.....	444
Defining Graph Lines or Bars.....	451
Inserting Data into a Graph.....	452
Deleting Data from a Graph.....	453
Redrawing the Graph	455
Sample PLOT Statements in a REXX Program	455

Appendix A: Customizing Special CA Automation Point Files 457

Customizing Keyboard Parameter Files.....	457
Keyboard Mapping File.....	457
Customizing Your Keyboard File.....	467
Running the Keyboard Configuration Program	468
Using Key Operations in Rules, REXX Programs, and Scripts	471
Customizing Scan Code Files.....	471

Understanding Scan Codes	472
Scan Code Parameter Summary.....	472
Customizing Your Scan Code Files.....	472
Associating a Scan Code File with a Session.....	473
Customizing Script Files.....	474
Understanding How Script Files Automate Tasks.....	474
Types of Scripts	474
Specifying Scripts to Use.....	474
Creating a Script.....	476
Script Keyword Summary	476
Specifying Keystrokes.....	477
How to Start a Script	478
Starting Scripts From REXX Programs.....	479
Starting Scripts From Rules.....	479
Starting Scripts From Menus.....	479
Sample Script and Console Screen	480
Appendix B: Using the TPF - CA Automation Point REXX Interface	485
Overview	485
TPF Components.....	485
TPFCON.REX.....	485
TPFREAD.REX	485
axcrules.rul	486
Setup Requirements.....	486
On the TPF System	486
On the CA Automation Point Machine	486
Message Processing	487
Appendix C: TN3270 and TN5250 Considerations	489
The TN3270E Status Line	489
The TN5250 Status Line.....	490
Support for TN5250 Auto Sign-On	491
Recording a TN3270 or TN5250 Session.....	492
Index	493

Chapter 1: Introduction

The following topics are covered in this guide:

- Configuring CA Automation Point
- Establishing communication between CA Automation Point and 3270 hosts (using native TN3270) and 5250 hosts
- Establishing communication between CA Automation Point and asynchronous hosts
- Managing sessions using CA Automation Point windows
- Viewing remote sessions
- Writing rules
- Configuring and writing REXX
- Using program-to-program queues
- Using notification services (voice, paging, and e-mail)
- Using Notification Manager
- Managing CMOS processors
- Interacting with external event systems
- Using the CA OPS/MVSInterface
- Using AS/400 Manager
- Using the Plot feature
- Customizing special CA Automation Point files

Notation Conventions

This guide uses the following conventions:

- Text in *italic* indicates a variable or command for which you must supply a value.
- Keywords that appear in [] are optional.
- A vertical bar (|) means you must choose between mutually exclusive values.
- Keywords in all uppercase must be entered exactly as shown

Chapter 2: Configuring CA Automation Point

This chapter discusses Configuration Manager, and how you can set up the various CA Automation Point automation tools. It also describes how you can use Configuration Manager to set up sessions.

Component Privileges

When you run any component of CA Automation Point, you do so as a Windows user account. During configuration of CA Automation Point, you can specify user accounts for a number of components that are run automatically.

Each new version of Windows places an increased emphasis on running applications with the minimum possible privileges. By following this commonly recommended security practice, you can reduce your security vulnerability and thus minimize the possibility of a costly security breach.

Those CA Automation Point components that do not require the privileges of a Windows administrator can be run as a standard user. The following table identifies whether a given component can be run by a user who is a member of the Windows Administrators group or by a user with standard user rights. It also identifies any additional privileges that must be assigned to that user to accomplish certain operations.

Component	Run with administrator rights	Run with standard user rights
CA Automation Point Installation Program	Yes	No
Configuration Manager	Yes	No
Manage Site Files--Export/Import Utility	Yes	No
All CA AP Windows Services (Notification Server, PPQs, etc.)	Yes See Requirement 1 (see page 20)	No
Automation Point Desktop	Yes See Requirement 2 (see page 20)	Yes See Requirement 2 (see page 20)
VOX Client REXX programs	Yes	Yes

Component	Run with administrator rights	Run with standard user rights
Notification Manager REXX programs	Yes	Yes
PPQ REXX programs	Yes	Yes
Notification Website	Yes	Yes
Web Message Viewer	Yes	Yes
Remote Viewer	Yes	Yes
Speech Notification Client	Yes	Yes

Requirement 1

By default CA AP Windows services run as the LocalSystem account that has administrative privileges. If configured to run as another user, that account must have the following user rights:

- Belong to the Administrators group
- Act as part of the operating system
- Allow log on locally
- Log on as a batch job
- Log on as a service

Requirement 2

Your decision to run the CA Automation Point Desktop as an administrator or as a standard user may vary based on the Windows operating system that you run at your site.

Windows Server 2003

Starting the Automation Point Desktop using AP Autostart Manager or AP Remote Viewer

The user name specified in the "Automation Services Startup Options" dialog and/or the "Remote Viewing" dialog must be a member of the Administrators group.

Starting the Automation Point Desktop as an Application

The CA Automation Desktop is started as an application when you start it from the Windows Start Menu. In this situation, you can run the Automation Point Desktop as either an administrator or a standard user.

In addition to the 'privileges required for standard users', which are listed below, if you are running the Automation Point Desktop as a standard user on Windows Server 2003, you must also assign the user Read permission for the following Windows registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib
```

A standard user does not have this permission by default on Windows Server 2003. You can assign Read permission to this registry key using the apAddRegRead program.

Windows Server 2008

Running the Automation Point Desktop as a Standard User

Typically, it is easiest to configure the Automation Point Desktop as a standard user and assign to that user the 'privileges required for standard users', which are listed below. Such a configuration works when the Automation Point Desktop is started from the AP Autostart Manager, the AP Remote Viewer, or the Windows Start Menu.

Autostarting the Automation Point Desktop as an Administrator

You can start the Automation Point Desktop from the Windows Start Menu as an administrator with no special configuration. However, if you choose to start the Automation Point Desktop, from the AP Autostart Manager or the AP Remote Viewer as a user who is a member of the Administrators group (but who is not the built-in Administrator), you must re-configure a value in the Windows User Account control (UAC) security settings.

Beginning with Windows Vista, Microsoft has included an enhanced security feature called User Account Control (UAC). A primary function of this UAC component is to prompt the user for acknowledgment when a program elevates into a privileged state. Because the Automation Point Desktop can require one or more advanced privileges, UAC displays an interactive prompt for acknowledgment before allowing execution of the Automation Point Desktop. No interactive prompt displays when you start the Automation Point Desktop using either the AP Autostart Manager or the AP Remote Viewer. Therefore, the prompt goes unanswered and the application appears to hang. The application can then only be stopped using the Windows Task Manager. You can prevent this hang by reconfiguring a value in the Windows UAC security settings.

To reconfigure UAC so that the AP Autostart Manager and the AP Remote Viewer can start the Automation Point Desktop as an administrator

1. Log in as an administrator.
2. From a Windows command prompt, type secpol.msc and press Enter.
3. If a UAC prompt displays, click Continue.
The Local Security Policy dialog displays.
4. Navigate to the following tree item located in the left pane of this dialog: Security Settings, Local Policies, Security Options.
5. To run the Automation Point Desktop using the built-in 'Administrator' account, find the following security policy in the right pane of this dialog:
'User Account Control: Admin Approval Mode for the Built-in Administrator account'.
The Windows default value for this UAC setting is 'Disabled'. If that is your current value, no change is required to run the Automation Point Desktop as the built-in Administrator.
6. To run the CA Automation Point Desktop using a member of the local 'Administrators' user group (who is not the built-in 'Administrator'), find the following security policy in the right pane of this dialog:
'User Account Control: Run all administrators in Admin Approval Mode'.
The Windows default value for this UAC setting is 'Enabled'. This value must be changed to 'Disabled' to run the Automation Point Desktop as an administrator other than the built-in Administrator.
Important: Setting this value to 'Disabled' reduces the overall security of your system. For this reason, only choose this option if your corporate policy prevents you from running the Automation Point Desktop as a standard user.
7. Double-click the appropriate security policy entry, and make sure the Disabled radio button is selected.
8. Click OK on the Security Policy dialog to save your changes (if any).

Note: This configuration setting will affect all applications.

Privileges Required for Standard Users

Regardless of operating system, if your site uses the following features and chooses to run the Automation Point Desktop as a standard user, you must assign the following rights to that standard user:

If you do one of the following:

- Send or receive messages from CA NSM Event Manager
- Send or receive messages from CA OPS/MVS

then the CA Automation Point user must have the following user right:

- Create global objects

If you monitor the Windows Security Event log, the CA Automation Point user must have the following rights:

- 'Manage auditing and security log' user right
- Read permission for the following Windows registry key:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security

You can assign Read permission to this registry key with the apAddRegRead program.

Configuration Manager

Configuration Manager is a graphical user interface you can use to configure various CA Automation Point functions and services.

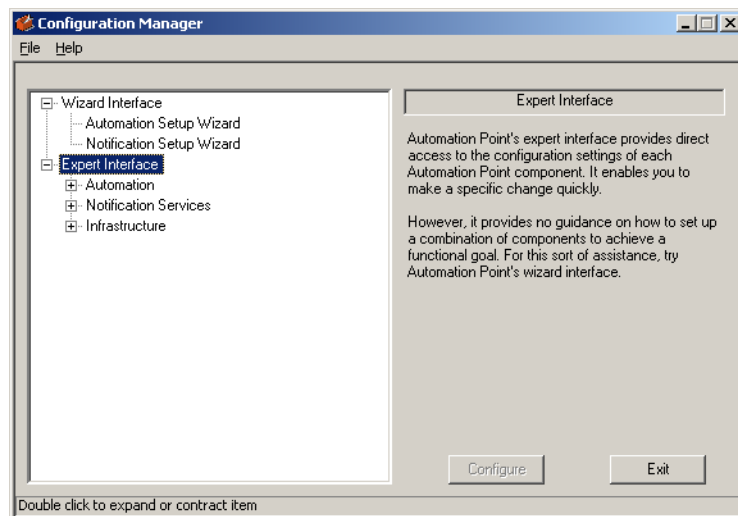
To start Configuration Manager, from the Start menu, choose Programs, CA, CA Automation Point, Configuration Manager.

Configuration Manager is composed of two interfaces:

- The **Wizard Interface** allows you to functionally configure the automation facilities and notification services.
- The **Expert Interface** allows you to access the settings of the CA Automation Point components. You can use this interface to drill down and quickly make changes to specific features.

Note: We recommend that you use the Wizard Interface to perform the initial configuration of essential facilities and services. However, you may want to use the Expert Interface to make smaller modifications. For example, if you have already used the Wizard Interface for initial configuration and only want to modify a specific setting, you may find the Expert Interface more expedient.

The following illustration shows the Configuration Manager expandable tree view with the interfaces and their components:



The following sections provide a more detailed description of the Configuration Manager interfaces.

Wizard Interface

The Configuration Manager Wizard Interface provides two intelligent graphical setup wizards, the *Automation Setup Wizard* and the *Notification Setup Wizard*, to guide you through a functionally complete configuration of the CA Automation Point automation facilities and notification services.

The following sections describe the function of each setup wizard.

Automation Setup Wizard

The Automation Setup Wizard guides you step-by-step through the configuration of the key components of an automation server. This setup wizard explains the relevance of each component to automation system operations, helping you determine whether a component is required for the automation needs of your site.

Examples of tasks that the Automation Setup Wizard lets you configure include:

- Automatic startup
- Scripts
- Session definition sets

- Rules
- Remote viewing
- Web message viewing
- Program-to-Program Queues
- Interface to CA NSM
- Interface to CA OPS/MVS
- Interface to HAF
- Windows Event Logs

Click Next on the Automation Setup Wizard to perform the next recommended task, and click Finish when you are through with your configuration.

Notification Setup Wizard

The Notification Setup Wizard guides you step-by-step through the configuration of the key facilities of Notification Services. This setup wizard explains the relevance of each facility, helping you determine whether a facility is required for the notification needs of your site.

Examples of tasks that the Notification Setup Wizard enables you to configure include:

- Voice card installation
- Notification Startup Options
- E-mail Notification Facilities
- Notification Manager Database

Expert Interface

The Configuration Manager Expert Interface provides direct access to the settings of each CA Automation Point component. The CA Automation Point components follow:

- Automation
- Notification Services
- Infrastructure

You can drill down to your particular area of interest, access each component's underlying facilities and make a specific change, add more sessions, add more rules, and so on.

Note: The Expert Interface *does not* provide guidance on how to set up a combination of components to achieve a functional goal. For this type of assistance, refer to the Wizard Interface.

The following sections describe the three components of the Expert Interface.

The Automation Component

The Automation component of the Expert Interface allows you to configure the facilities that control the automated system operations of your site. With the Automation component, you can configure all of the facilities presented by the Automation Setup Wizard, and other less commonly modified features. By navigating through an expandable tree structure, you can drill down to a particular feature of interest and modify its configuration settings.

The facilities that you can configure using the Automation component of the Expert Interface include:

- Automation Services Startup Options
- Scripts
- Session definition sets
- Rules
- REXX
- Automation Point Desktop settings
- Message logging settings
- Non-volatile status variables
- Remote viewing
- Web message viewing
- Events interface
- AS/400 Manager

The Notification Services Component

The Notification Services component of the Expert Interface allows you to configure the facilities used to provide notification capabilities for your site. With the Notification Services component, you can configure all of the facilities presented by the Notification Setup Wizard, and other less commonly modified features. By navigating through an expandable tree structure, you can drill down to a particular feature of interest and modify its configuration settings.

The facilities that you can configure using the Notification Services component of the Expert Interface include:

- Notification Services startup options
- Paging
- Voice
- Email
- Notification Manager

The Infrastructure Component

The Infrastructure component of the Expert Interface allows you to configure the underlying facilities used in support of both your Automation and Notification facilities.

The facilities that you can configure using the Infrastructure component of the Expert Interface include:

- Error Tracing
- Debugging
- Program-to-program queues (PPQs)
- Services
- Diagnostic reports
- Java/Tomcat Installation

Managing Your Site Configuration

CA Automation Point supplies you with a rich set of extensions that you can use to configure and manage your site's data to suit your needs.

- Use Configuration Manager to configure, view, and edit settings saved in user data files that customize your site. Configuration Manager saves the user data files in a directory structure separate from the directory where CA Automation Point-distributed files are installed.
- Use the Export and Import Utility to backup, export, or import the full set of Configuration Manager user data files and accompanying user supplied files residing in the Site directory. Configuration files exported from the local CA Automation Pointserver can be imported for use by a different CA Automation Point server. Likewise, configuration files exported from another CA Automation Point server can be imported for use by the local CA Automation Point server. Use the Export and Import Utility to facilitate upgrading the operating system or base hardware for a preconfigured CA Automation Point server.
- Use the Import Sessions Utility to import or export the user data files that define just your session definition sets. Importing and exporting session definition sets from one CA Automation Point server to another allows you to move session definitions without manually reconfiguring them.

File Directory Structure

The CA Automation Point installation program prompts you to supply the following:

- A "Product installation folder" where you want the program to be installed. This folder is referred to in this guide as *installDir*.

For a new installation, the default *installDir* path is %PROGRAMFILES%\CA\CA Automation Point (%PROGRAMFILES(X86)%\CA\CA Automation Point on a 64-bit system)
- A "User files folder", where you want all site-specific files to be stored. This folder is referred to in this guide as *installUserDir*.

For a new installation, the default *installUserDir* path is %ALLUSERSPROFILE%\CA\CA Automation Point.

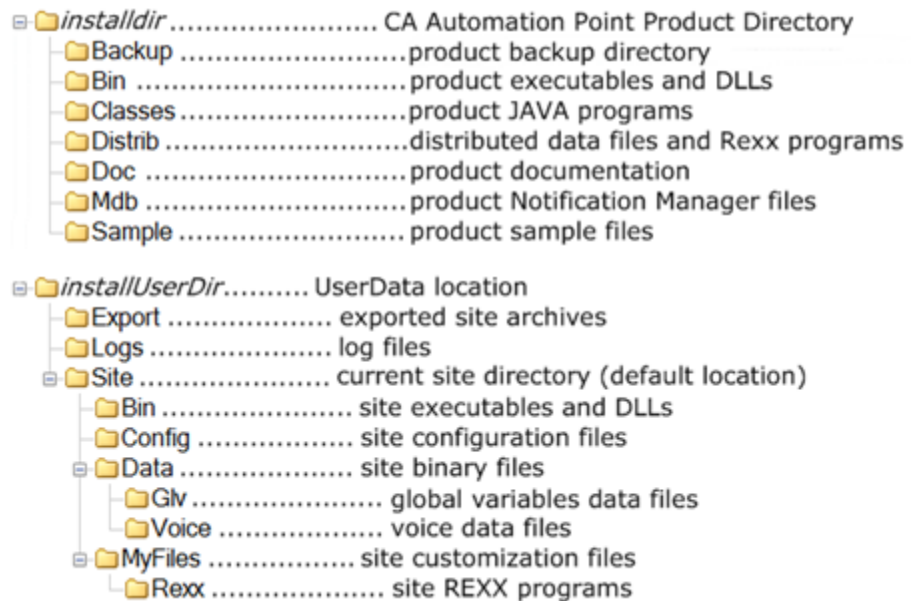
System variables pointing to defined system folders (%var%) are present on all supported operating systems; however, their exact location may vary based on the version of the operating system and its settings.

Your current configuration and user files are located in the Site directory (referred to in this guide as *siteDir*). During installation of a new release, a shortcut to this directory is placed in the CA Automation Point folder in your Start menu.

Environment variables that point to the above-mentioned directories are created at installation time:

- Product installation folder (*installDir*): %AP_HOME%
- User files folder (*installUserDir*): %AP_DATA%
- Site folder (*siteDir*): %AP_SITE%

Following diagrams shows detailed directory structure:



Note that all CA Automation Point-distributed files reside in directories under the installation directory. The site-specific files you create reside in directories under the Site directory. Site folder subdirectories include the following:

Bin

Contains site-specific and customer-created CA Automation Point-related executables and DLLs. Executables and DLLs for product patches and test fixes should not be put in this directory.

Config

Contains configuration files that you can modify using CA Automation Point tools. These include user interface layouts, the PPQ hosts file, the remote view hosts file, the session definition sets, and various binary files generated by Configuration Manager.

Data

Contains site-specific and customer-created CA Automation Point-related binary files. The GLV subdirectory contains the data files used to store the values of non-volatile GLV variables. The Voice subdirectory contains voice files such as VOX files, voice library files, and voice files created during the execution of the Notification Manager component.

MyFiles

Contains customization files that you can modify using a generic editor. These files include rules files, script files, keyboard parameter files, scan code files, screen fonts files, menu files, and log format files. The REXX subdirectory holds site-specific REXX programs.

Other directories that have a fixed location under *installUserDir* are as follows:

Logs

Contains various output log files and log files moved from previously exported sites (in separate subfolders)

Export

Contains exported site configuration archives and a backup of the current site configuration archive (created automatically before any import operation is performed).

How CA Automation Point Looks for Files

CA Automation Point looks for many customization files, both when it starts and during operation. It distributes these files, which are stored in the *installDir*\Distrib directory, for use in initial site setup. You can create your own such files, which are stored in the %AP_SITE%\MyFiles directory, to extend the use of the product. CA Automation Point uses a special search order to support this directory structure.

The search order for site-customizable files such as rules and scripts follows:

1. Site customization files directory, the %AP_SITE%\MyFiles subdirectory
2. Product support files directory, the *installDir*\Distrib directory

The search order for REXX programs follows:

1. Site REXX files directory, the %AP_SITE%\MyFiles\REXX subdirectory
2. Product support files directory, the *installDir*\Distrib directory

Managing Site Files

You can use the CA Automation Point Export and Import Utility to help you manage your site files and make backup copies regularly. It is important that you conduct ongoing backups of your CA Automation Point directory so that you can recover your configuration settings and notification policies in the event of hardware failure.

You can use the Export and Import Utility to do the following:

- Restore configuration settings that were lost due to an unforeseen event. For example:
 - Your disk crashes and you want to restore it from a CD.
 - You accidentally change your configuration settings and you want to return to your original settings.
- Save multiple configuration settings based on the needs of different end users. For example, there could be different settings for the following shifts, eliminating the need to change the settings when a new shift begins:
 - Day shift
 - Night shift
 - Weekend shift

- Import a configuration from a different CA Automation Point installation, including a configuration from a previous currently supported release.

After you perform such an import, you can view an report in HTML format that shows you the following:

- Imported settings that were automatically changed
- Settings that you may need to manually change so that they work properly in the new CA Automation Point installation

You can execute the Export and Import Utility from Configuration Manager or from the command line. Executing the utility from Configuration Manager is referred to as *interactive* mode. Executing the utility from the command line is referred to as *unattended* mode.

Note: Because the Export and Import Utility requires administrator rights, it is not generally possible to execute it from a rules file. If you try to run the utility from a rules file, the application fails on Windows 2003 Server or displays UAC interactive prompt on Windows 2008 Server. However, if Automation Point Desktop is already running with administrator rights, the Export and Import Utility executed by a rule will already have the required privileges, and performs the required operation.

Export and Import Operations

Export

You can use the Export Settings function to copy all configuration settings into an archive you specify. These site archives reside in the installUserDir\Export directory, which is created by CA Automation Point. The export can be completed without shutting down any CA Automation Point applications or services.

All files are compressed and stored in this archive. To create different configurations for use under various circumstances, specify different archive names. These archives become standby site settings that you can later import into your active site directory.

You can export configuration settings using Configuration Manager (by choosing Manage Site Files from the File menu), from a rules file (if Automation Point Desktop is running with administrator rights), or a from a Windows command prompt window.

You can also transfer exported configuration settings from one CA Automation Point server to another CA Automation Point server where you can import them using the same Export/Import utility.

1. On the server from which you want to transfer exported configuration settings, locate the exported site archives in the %AP_DATA%\Export directory.
2. Copy the exported site archives to any removable storage device (such as USB flash drive).
3. On the target CA Automation Point server, copy the site archives from the removable storage device to the %AP_DATA%\Export folder.
4. From Configuration Manager, run Manage Site Files.

The imported configurations are displayed in the Standby Site Archives list of the utility dialog.

If you need to import any CA Automation Point r11.2 configuration settings that you previously exported by previous version of this utility to a CA Automation Point r11.4 server, follow these steps:

1. If you saved the previously exported sites to a removable storage device, connect the device to the CA Automation Point r11.4 server.
2. Locate the root folder of the exported site (for example, Site_Aug17) on the removable storage device.
3. Copy this root folder to the %AP_DATA%\Export directory of the target server.

The resulting folder structure could be, for example,
%AP_DATA%\Export\Site_Aug17\MyFiles

4. Run the Export and Import Utility by selecting Manage Site Files option from the File menu of Configuration Manager.

The copied site directories are automatically compressed into archives and you should see them in the Standby Site Archives list.

5. Select the desired configuration settings in the Standby Site Archives list, and click Import to Site.

The selected settings are imported to your Site.

Import

You can use the Import Settings function to switch from using the active site settings to the previously archived site settings you specify, and to restart all CA Automation Point services to reflect new configuration. You can import configuration settings using the Export and Import utility, which you can launch from Configuration Manager or from a Windows Command Prompt window.

The import process works as follows:

1. Before starting the import operation, all CA Automation Point services are stopped
2. The import operation automatically backs up current site files to an archive named "Site_autobackup" and stores the archive in the *installUserDir*\Export directory. You can use this backup to switch back to the previous state before performing an import operation.
3. The site directory is cleared and the selected archived site files are extracted to that directory.
4. All CA Automation Point services are restarted as specified by the imported configuration settings.
5. A successful import operation generates a report named *importReport.html* and stores it in the Logs directory. This report shows the following:
 - Settings that were automatically changed (an import from previous release of CA Automation Point)
 - Settings that you might need to change if the previous configuration came from a different CA Automation Point installation on different machine.

Manage Site Files Using Configuration Manager

To manage your site files from Configuration Manager, choose Manage Site Files from the File menu.

The Manage Site Files dialog appears, displaying the current active site directory and a list of standby site directories that are not currently in use, including ones that were previously exported. You can use this dialog to perform the following actions:

- Select from or sort the list of standby directories.
- Make a backup copy of the active site configuration.
- Switch to a standby site configuration. This automatically backs up the current site configuration to a fixed archive named "Site_autobackup," then extracts the former standby site archive into the active site directory.
- Delete a standby site archive.

For more information about this dialog, see the dialog Help.

Manage Site Files Using the Command Line Interface

In the unattended mode, you must specify which operation you want to perform and supply additional parameters to complete the operation successfully. If you do not specify any command line parameters, the application starts in interactive mode.

The command line syntax is as follows:

Export:

```
expimpcfg.exe -e [folderpath] [-n archivename] [-d description]
```

Import:

```
expimpcfg.exe -i [folderpath] -n archivename [-s]
```

Other:

```
expimpcfg.exe [-v| -ver| -?| -h| -help]
```

Parameters can be specified with a preceding '-' or '/' character.

/e* [*folderpath*] or ***-e* [*folderpath*]*

Specifies an export operation. If you do not specify the optional *folderpath*, the exported archive will be stored in the standard Export folder (specified by %AP_DATA%\Export).

/i* [*folderpath*] or ***-i* [*folderpath*]*

Specifies an import operation. If you do not specify the optional *folderpath*, the operation looks for the site archive in the standard Export folder (specified by %AP_DATA%\Export)

/n* *archivename* or ***-n* *archivename**

Specifies the name of the archived site on which you want to perform either export or import operation.

If you specify */e*, this parameter indicates the name of an archive (without extension) into which the current site files will be exported. If this parameter is omitted the application uses the default archive name "AP_Site_backup".

Note: If the target site archive already exists, it will be overwritten without notification.

If you specify */i*, this parameter indicates the name of an archive (without extension) from which site files are extracted and copied to the current site folder. This parameter with its value is required for an import operation.

/d description or -d description

Supplies an additional description of the currently exported site. Use only with the /e parameter (export operation).

/s or -s

Specifies that the configured CA Automation Point services are not to be started after the import operation completes. This parameter is used only with the /i parameter (import operation). You can manually start services later using the Services dialog of Configuration Manager or by restarting the computer

/v or -v

/ver or -ver

Displays version information and exits

Note: You can view these parameter descriptions from a Windows Command Prompt window by entering one of the following commands:

```
expimpcfg.exe /? or -?  
expimpcfg.exe /h or -h  
expimpcfg.exe /help or -help
```

To view the status of the operation, and the status messages it generates, see the ExplmpCfg.log file.

Example

To export configuration settings to an archive named AP_BACKUP, enter the following command:

```
expimpcfg.exe -e -n AP_BACKUP
```

Return Code

If the operation ends successfully, the application returns a 0. Otherwise, it returns a 1. You can check the return code by running the expimpcfg.exe executable within an MS-DOS batch script (.bat), then checking the %ERRORLEVEL% variable.

For more information about the process and possible cause of failure, see the ExplmpCfg.log file in the %AP_DATA%\Logs directory.

Automating the Backup Process Through Rules

If Automation Point Desktop is running with administrator rights, you can back up your configuration settings automatically by writing a time rule. For example:

```
TIME(02:00), DOSCMD(expimpcfg.exe /e /n DailyBackup /d "Backup generated by time rule")
```

This example backs up the site configuration to the archive named DailyBackup at 2:00 a.m. each day and includes a description.

For more information about writing time rules, see the chapter "[Writing Rules](#) (see page 149)".

Import Sessions Utility

You may want to move only your session definitions from one machine to another, and leave the remainder of your destination machine's configuration unchanged. Such an operation can be useful when preparing a disaster recovery machine or when load-balancing your sessions across multiple CA Automation Point servers. You can use the Import Sessions utility program to accomplish this task. You can launch the Import Sessions program from the Configuration Manager menu item: File, Import Sessions. You can also run the Import Sessions program directly by running the program `importSess.exe`.

For detailed information about the operation of the Import Sessions program, go to Help, Usage on the Import Sessions menu.

Using Configuration Manager to Set Up Sessions

To configure sessions for CA Automation Point to monitor, start Configuration Manager, choose Expert Interface, Automation, and click Session Definition Sets. Using this entry point, you can perform the following tasks:

- Configure CA Automation Point sessions and function windows.
- Define multiple session definition sets.
- Define communication options necessary to connect to a monitored session.

Creating a Session Definition Set

In the Session Definition Sets dialog a default session definition set is created for you. To modify this session definition set and create new ones, use the right-click options on the Session Definition Set tree. Double-click an item to view or modify your session definition sets.

Default values and temporary names are provided on the Session Definition Sets dialogs to help you with your initial configuration. When you configure your session definition sets, you must customize these values for your site, and enter values for options that have no preset values. If you omit entering a value for an option that is required, you are reminded before you leave the dialog.

The following items in the session definition set tree allow you to access specific options for configuring a session definition set. These settings collectively configure the CA Automation Point operating environment.

- Rules Settings
- Global Session Settings
- Overriding REXX Settings
- Sessions
- Function Windows

Note: For details about each option on the Session Definition Sets dialogs, see the dialog-specific Help. The Help for the Session Definition Sets dialog explains the use of the right-click options on the Session Definition Set tree.

Activating a Session Definition Set

To activate a session definition set, right-click name of the session definition set and choose Set As Active from the drop-down menu. Only one session definition set can be active at a time. The active session definition set is loaded the next time CA Automation Point starts up.

Enabling a Session or Function Window

To enable a session or function window, check the appropriate box in its definition. Session and function windows must be defined to the active session definition set and be enabled before they will appear on the Automation Point Desktop.

Creating AXCREXX Sessions

Standard output stream for REXX programs started within the CA Automation Point environment or from the Automation Point Desktop is displayed in the AXCREXX window. You can define this window as a session on the Automation Point Desktop so that you can view it remotely using the Remote Viewer.

To create an AXCREXX session

1. In the Session Definition Sets dialog in Configuration Manager, add a new session.
2. Choose the Default Console Type.
3. Choose the VIO terminal.
4. Specify AXCREXX as the session name.
5. Customize your session settings in the Windows Command Prompt Session Settings dialog as follows:
 - a. Select the viont.scn key codes file.
 - b. Set the window height to 25.
 - c. Set Send Delay to 0.

In the local session settings, ensure that web viewing is *not* enabled for the session.

Creating Windows Command Prompt Sessions

A Windows Command Prompt session is a terminal emulator session to the local machine that appears on the Automation Point Desktop. From the Automation Point Desktop session window, you can type commands to the CA Automation Point server machine and view its output. You can do this locally, or you can do it remotely through the use of the Remote Viewer.

Windows Command Prompt sessions are non-automated sessions; however, you can use the CA Automation Point ADDRESS command environments to programmatically control the session and perform automation tasks for character-based applications running locally out of the command prompt window.

To create a Windows Command Prompt Session

1. In the Session Definition Sets dialog in Configuration Manager, add a new session.
2. Uncheck the Automate Session checkbox.
3. Choose the Default Console Type.
4. Choose the VIO terminal.
5. Specify a unique session name.

6. Customize your session settings in the Windows Command Prompt Session Settings dialog as follows:
 - a. Select the viont.scn key codes file.
 - b. Set the window height to 25
 - c. Set Send Delay to 0.

In the local session settings ensure that web viewing is *not* enabled for the session.

How You Create External Event Monitoring Sessions

CA Automation Point has the ability to monitor events from external event systems such as CA NSM Event Manager, CA OPS/MVS, the Hardware Management Console for mainframe CMOS processors, and Windows event logs. The sessions required to monitor events from these sources are generated automatically when the appropriate event interface is configured using Configuration Manager. To configure the appropriate event interface, choose Expert Interface, Automation, Events Interface, and then select the event interface that you want to configure (CA NSM Event Manager, CA OPS/MVS, HAF, or Windows Event Logs).

Because these types of sessions are automatically generated, you are not able to view or modify these sessions using the Session Definition Sets dialog. If you want to view or modify the session definition for a specific event interface, you can choose the Session Definition button located on the corresponding event interface configuration dialog.

Related Configuration Tasks

As you create and configure your sessions and set your communication settings, it will also be necessary to perform other configuration tasks, including the following:

- Configure global Automation Point Desktop settings
- Configure message logging settings
- Edit REXX programs and set other REXX-related options
- Create and specify prefixes for non-volatile status variables
- Edit and configure options related to script files

You can access these items using Configuration Manager by clicking the appropriate item under Expert Interface, Automation. Instructions and HTML help are provided with the interface.

Chapter 3: Establishing Host Sessions

This chapter describes how you can establish communication between CA Automation Point and mainframe or midrange hosts using TN3270 or TN5250 connections.

Communicating with the Mainframe

The maximum number of combined 3270 and 5250 sessions that can be configured is 64. The maximum number of sessions you are allowed (by license) depends on the particular interface license you have purchased.

Communicating With Mainframe Host Using TN3270

Before you can establish communications between CA Automation Point and your IBM or IBM-compatible mainframe host, you must configure a communication server that supports TN3270 (such as OSA-Integrated Console Controller (OSA-ICC) or IBM 2074 Console Controller) to host either operating consoles or VTAM applications using TN3270E connections through IBM 3278/3279 model 2,3,4, or 5 display terminals.

Note: For a 3270 session defined to CA Automation Point to properly connect to a z/OS console, the console must also be defined to the mainframe host. A z/OS console is defined on the mainframe host in member CONSOLxx in the SYS1.PAEMLIB library, where the xx in the CONSOLxx is the value of the CON parameter in member IEASYS00.

TN3270E terminal emulation is native to CA Automation Point. Using Configuration Manager, you must also configure corresponding TN3270E connection settings and terminal display emulation for each mainframe host session you wish CA Automation Point to establish. Console type, when applicable, must also be specified for each defined 3270 session to enable built-in console drivers to manage the session accordingly.

Note: For faster message processing and better action message handling by CA Automation Point, we recommend that you configure 3270 console sessions for IBM 3278/3279 model 4 terminal emulation.

Communicating with Midrange Host Using TN5250

Before you can establish communications between CA Automation Point and your IBM System i host (system i is also designated as iSeries or i5/OS or AS/400), you must configure a communication server that supports the TN5250 terminal communications protocol on that host.

Notes:

- For IBM iSeries connections, CA Automation Point supports the 5292 terminal type only. This provides a 24x80 screen size.
- Sessions of this type are not automatable within CA Automation Point.

Configuring SSL for CA Automation Point

Secure Socket layer (SSL) is a protocol for securing data transfer by applying cryptographic techniques for data authentication and encryption. CA Automation Point installs and uses an internal CA product called CA Socket Adapter, an OpenSSL application, to implement support for SSL. Under CA Automation Point, CA Socket Adapter is configured to use the Anonymous Diffie-Hellman (ADH) key exchange. The base Diffie-Hellman algorithm for key exchange is used without authenticating the remote TN3270 or TN5250 server. Therefore, certificates are not required to establish a secure connection. Data exchange is still encrypted according to cipher specifications negotiated between remote server and CA Socket Adapter.

Automation Point Desktop sessions with 3270 or 5250 connections may be configured to run using SSL. To configure, use Configuration Manager to enable SSL when specifying connection options for a 3270 or 5250 session definition.

Monitoring Connection Status

The last line of each session window displays connection-related status information. For a description of the status codes for TN3270E and TN5250 sessions, see the Appendix, "[TN3270 and TN5250 Considerations](#)" (see page 489)."

You can reference the status information programmatically using CA Automation Point's built-in ADDRESS AXC command environment. Issue the GETSCRN command to capture and save the session's screen image into REXX variables for automated processing. For more details, see the *Command and Keyword Reference Guide*.

Automating the IPL Process for z/OS

CA Automation Point can initiate and complete the IPL process. The product initiates the IPL through the APCMOSI.REX console session and then automates the early messages that appear during the IPL process through the z/OS system (or MCS) console session.

Processor Consoles

The Hardware Management Console (HMC) is the GUI application that acts as the processor console for CMOS mainframe processors. It runs on a Linux workstation. In addition to providing GUI dialogs, the CMOS hardware vendors provide, as part of the HMC software, an application programming interface (API) for performing IPL activities. This API is called from REXX.

CA Automation Point is not designed to automatically navigate through and interact with a series of HMC GUI dialogs. Therefore, it supports automating the IPL activities for a CMOS processor through REXX programming calls to the APIs.

CA Automation Point provides a REXX program named APCMOSI.REX, which can be automatically executed in a managed session from a menu on the Automation Point Desktop or from a time rule. After CA Automation Point initiates the IPL through APCMOSI.REX, z/OS begins issuing IPL-related messages to the first active MCS console that it finds. By controlling this console, the product can respond to each message automatically.

For information on APCMOSI.REX, see the chapter "[Managing CMOS Processors](#) (see page 333)."

Setting Up IPL Automation

To set up IPL automation, you configure your host system so that it sends its IPL-related messages to an MCS console monitored by CA Automation Point. Because z/OS sends IPL messages to the first active console that it finds, you need to define (or set during system generation) the console with console ID 01 as a CA Automation Point-controlled session.

To set up IPL automation

1. Using Configuration Manager, go to Session Definition Sets and define a 3270 host connection to the IPL console for the mainframe host:
 - a. Define a session with console type MCS and enable it for automation.
 - b. Designate the defined session as the IPL console, enabling CA Automation Point to recognize IPL messages and direct them to rules for processing:
 1. Click Customize Scripts.
 2. Select IPL for Console Type/State.
 3. Click Add.
 4. The IPL state is added to the MCS session definition.
 - c. Click Customize Session Settings, then specify your connection settings.
2. Write specific CA Automation Point rules to do the following:
 - Initiate the IPL
 - Reply to IPL-related messages

The sample rules file, named `axcrules.rul`, contains CA Automation Point rules for automating responses to the IPL-related messages that are sent to the MCS console. You may need to customize the rules for your site.

You can initiate the IPL process in any of the following ways:

- **A menu item**--For information on initiating APCMOSI.REX from a menu item, see the section on customizing menus in the chapter "[Managing Sessions Using CA Automation Point Windows](#) (see page 75)."
- **The Host Command Area**—From any CA Automation Point function window with a Host Command Area, you can initiate the APCMOSI.REX manually by entering the name of the REXX program to start.
- **A time rule**--For information on initiating APCMOSI.REX from a time rule, see the chapter "[Writing Rules](#) (see page 149)."
- **A command rule**—For information about initiating APCMOSI.REX from a command rule, see the chapter, "[Writing Rules](#) (see page 149)."

Synchronizing Date and Time with the Mainframe Host

You can enable CA Automation Point rules to trigger events or actions by time of day according to the local server. Before activating automation, we recommend that you synchronize the local server clock with mainframe host clocks for all configured 3270 sessions.

Setting the Local Server Clock

Before setting the local server's internal clock, we recommend that you exit from CA Automation Point completely and restart it *after* you have set the clock.

If you must change the local server clock while CA Automation Point is running, remember that the CA Automation Point internal timing operations are not dependent on the clock. However, several environmental variables (&DAY, &DATE, &LDATE, &JULDATE, &LJULDATE, and &TIME) and time rules still reference the local server clock.

Date and Time Environmental Variables

If you have written rules that are dependent on the &DAY, &DATE, &LDATE, &JULDATE, &LJULDATE, and &TIME environmental variables, you can change the clock while CA Automation Point is running without affecting automation.

Time Rules

If you have written time rules, exit from CA Automation Point completely before changing the clock; otherwise, your time rules may execute erratically. Because of the complex relationship between time rules and the local server clock, the effect on automation at your site is unpredictable.

Note: Changing just the date has no effect on time rules.

Establishing Cooperative Processing with Mainframe Host Automation

CA Automation Point can act as a server to mainframe automation products such as CA OPS/MVS. To establish cooperative processing, you need to set up a few rules in each product.

When CA Automation Point wants to request a service from a mainframe automation product, it sends a command and receives one or more messages as an acknowledgment. When a mainframe automation product wants to request a service from CA Automation Point, it sends a message and receives one or more commands as an acknowledgment. The mainframe automation product needs to know which consoles CA Automation Point controls so that it can send a service request message to the correct console.

CA provides the AXCCOMM application, which sets up and dismantles the logical host-workstation connections. It also handles error conditions that may arise when other applications "sign in" to receive a certain type of data from the host and "sign out" when they no longer need to receive the data. The application works with the AXC HERE and AXC GONE commands in the default CA Automation Point state and pause scripts. For more information, see the AXCCOMM.txt file.

General Guidelines

When setting up and configuring your system, follow these guidelines:

- Suppress much of the message traffic to the master console by either suppressing the message entirely or by using route codes to display the message only on the consoles that need to see it. This is especially important for highlighted messages. We recommend that you suppress 90% of your message traffic.
- Monitor the status of the WTO buffer for master consoles and clear the buffers if they begin to fill up. Sample rules are available for CA OPS/MVS for this purpose. See the axcrules.rul file.

Understanding Console Management

CA Automation Point directly automates certain types of consoles by using built-in console drivers. The console type you choose in the Console Type field when you are defining your session determines how the session is managed.

This section discusses the how CA Automation Point manages various session types, and how it handles messages.

SYSPLEX, MCS, and RCS Consoles

SYSPLEX, MCS, and RCS are keywords used in the session definition to specify the kind of session that CA Automation Point is to automate.

- SYSPLEX—Master consoles receiving messages from multiple systems
- MCS—Master consoles receiving messages from only one system
- RCS—CA Remote Console sessions

Automating SYSPLEX, MCS, and RCS

CA Automation Point uses a similar method to automate SYSPLEX, MCS, and RCS sessions. (JES3MCS is a synonym for SYSPLEX.) Use the red command area or Command dialog to manually enter commands into an automated session.

Keep the following points in mind:

- Do not manually issue K commands (such as k e,n) that directly affect the messages on the emulation window while automation is active in the session. Other K commands that do not affect the messages (such as K Q and K E,D) are acceptable.
- Do not create a display area while automation is active in the session. Doing so produces unpredictable results.
- By default, keyboard input is inhibited while automation is active on these consoles.

Determining the State of a Session

When automating a session, CA Automation Point examines the screen contents to determine the state of the session. If CA Automation Point cannot identify the current state of the session, the console is considered to be in the INIT state and the INIT script is initiated (if the INIT state has been specified). The purpose of the INIT script is to put the console in a state that CA Automation Point can recognize.

A state script is an interpreted CA Automation Point file that sets the console to the proper mode for automation. CA supplies sample state scripts with CA Automation Point. You can use the sample files as they are or edit them as needed.

The purpose of the state script is to put the console in non-delete mode with the appropriate message format required for rules processing. A state script determines the console state by looking for the appropriate console-specific message on the status line. If found, the state script issues the console command to set the console characteristics required for use by CA Automation Point.

If the console will also receive IPL messages and you want to process the IPL messages through rules:

- Specify IPL in the Console Type/State field on the Customize Scripts dialog.
- In your 3270 session settings, set the number of lines on the IPL screen in the IPL Window Height field.

Console Processing for SYSPLEX, MCS, and RCS

All SYSPLEX, MCS, and RCS sessions should have an INIT script specified for the session. When CA Automation Point recognizes the SYSPLEX, MCS, or RCS state, the appropriate state script is started.

SYSPLEX Consoles

CA Automation Point recognizes a SYSPLEX session by the IEE612I or IEE512I message. The state script puts the console in non-delete mode and formats the messages to include time stamps, session name, and job names. The state script also deletes any existing display areas and issues a message telling CA OPS/MVS that CA Automation Point is active on this console. (See the AXCCOMM application.) CA provides a sample SYSPLEX.scr file with CA Automation Point. For more information on AXCCOM, see the AXCCOMM.txt file.

A message on a SYSPLEX console may look like this:

```
*09.28.38 X2345678 SYSABC *093 IEF355A INITIATOR TERMINATED, RESTART
```

When processing this message, CA Automation Point sets the following environmental variables as follows and passes the message to rules for processing:

Variable	Is Set to This Value
&HOSTTIME	09.28.38
&JOBNAME	JOB45678
&JES3NAME	SYSABC
&REPLYID	093
&MSG	IEF355A INITIATOR TERMINATED, RESTART
&WORD1	IEF355A
&WORD2	INITIATOR
&WORD3	TERMINATED,
&WORD4	RESTART

MCS Consoles

When you configure z/OS, define the device addresses to be controlled by CA Automation Point as an MCS console. If you plan to use CA Automation Point to reply to IPL messages, define or set the first console (console ID 01) as a CA Automation Point-controlled session.

CA Automation Point recognizes an MCS session by the IEE612I or IEE512I message. The state script puts the console in non-delete mode and formats the messages to include time stamps and job names. The state script also deletes any existing display areas and issues a message telling CA OPS/MVS that CA Automation Point is active on this console. (For more information on AXCCOM, see the AXCCOMM.txt file.) CA provides a sample mcs.scr file with CA Automation Point.

Specify the consoles that are to receive the IPL messages by running the z/OS configuration program with the following statements:

```
IODEVICE ADDRESS=xxx, ...
NIPCON=(xxx, ...other consoles... )
```

z/OS sends IPL messages to the first active console defined on the NIPCON parameter.

You can also set up alternate configurations and request one of them during IPL. For example, you may want to control your data center locally on weekdays and remotely on the weekend (for remote IPL). The following example shows one way that you can set up alternate configurations for CA Automation Point:

For normal operations— Set up the standard configuration using console 01, a local 3279, as the first NIP console.

For remote operations— Set up the alternate configuration using console 02 connected to CA Automation Point as the first NIP console.

A message on an MCS console might look like this:

```
*09.28.38 X2345678 *093 IEF355A INITIATOR TERMINATED, RESTART
```

When processing this message, CA Automation Point sets the following environmental variables as follows, and passes the message to rules for processing:

Variable	Is Set to This Value
&HOSTTIME	09.28.38
&JOBNAME	JOB45678
&REPLYID	093
&MSG	IEF355A INITIATOR TERMINATED, RESTART

Variable	Is Set to This Value
&WORD1	IEF355A
&WORD2	INITIATOR
&WORD3	TERMINATED,
&WORD4	RESTART

RCS Consoles

CA Automation Point recognizes a CA Remote Console session by the RCS5663 or RCS5664 message. The state script puts the console in non-delete mode and formats the messages to include time stamps and job names. The state script also deletes any existing display areas and issues a message telling CA OPS/MVS that CA Automation Point is active on this console. CA provides a sample rcs.scr file with CA Automation Point. For more information on AXCCOM, see the AXCCOMM.txt file.

Note: If your RCS console operates with anything other than 20 lines on the screen, you must change the value of the SEG parameter (on the KEY statement) in the RCS script. This is the default KEY statement:

```
KEY=(K S,DEL=N,SEG=20,CON=N,MFORM=(T,J)@E)
```

RCS messages are processed similarly to MCS messages.

Message Handling for SYSPLEX, MCS, and RCS Consoles

When CA Automation Point encounters a SYSPLEX, MCS, or RCS session in non-delete mode, it reads the first line on the console. If the message does not have a time stamp, CA Automation Point deletes the line. This process continues until CA Automation Point finds a time-stamped message on the first line of the session.

When CA Automation Point reads a message, it processes the message by assigning values to environmental variables, building a PREFIX for the message and processing the message through rules.

An environmental variable is a CA Automation Point variable that contains information about the current system environment when a message is issued. For example, the variable could store the current time or the job ID associated with the message.

Action Message Handling for SYSPLEX, MCS, and RCS Consoles

An *action message* is a highlighted message on a sysplex, MCS, or RCS console. If CA Automation Point encounters an action message, it performs additional processing, including:

- Displaying the message in the action message area of the Merged Message window
- Displaying the message in the action message window
- Deleting the non-highlighted messages above the action message so that all action messages float to the top of the screen. (If the highlighted messages fill more than half the screen, CA Automation Point begins deleting the highlighted messages that are near the top of the screen.)

When an action message receives a reply, the action message indicator changes. CA Automation Point recognizes the change and removes the action message from the action message area of the Merged Message window and from the action message window.

If the message receives a reply after it clears from the console window, CA Automation Point does not recognize that it should clear the message. The message remains in the action message area until it scrolls off the screen, and it remains in the action messages area of the Merged Message window until it either scrolls off the screen or someone manually deletes it. CA Automation Point periodically issues a command to delete messages from the terminal session.

Important! When a line on the screen does not begin with a time stamp, CA Automation Point assumes that the line is a continuation of the previous message. CA Automation Point concatenates the two lines and processes them as one message. If a message is longer than two lines, CA Automation Point processes the first line through rules, and processes each additional line using the characteristics of line one for display, printing, and logging.

Note: Using a Model 4 console improves action message handling.

JES3 Consoles

CA offers the following recommendations for automating JES3 consoles:

- For **JES3 LOCAL consoles**, the host route the messages to the MCS operator console, and then have CA Automation Point automate the MCS console. This scenario is possible with all current versions of JES3.
- For **JES3 GLOBAL consoles**, route the messages to the MCS operator console and have CA Automation Point automate the console as type SYSPLEX.

Message Handling for JES3 Consoles

To determine whether a session is running on a JES3 console, CA Automation Point looks for the screen to display IAT7185 in column 1 and S.P. in column 70. CA supplies a JES3 state script in the active rules file JES3.scr that slows the JES3 console's scroll rate to allow CA Automation Point to keep up with it. To use the script, you need to write this rule:

```
TIME(00:00) EVERY(1 MINUTE) MATCHLIM(1)
SCRIPT((JES3.SCR) SESSION(sessname))
```

Note: CA Automation Point does not run the JES3 state script automatically. You must run it through a rule.

If you have defined a pause script for this session and you pause automation—either globally or for this session—CA Automation Point executes the pause script. (A pause script returns the console to the state it was in before CA Automation Point executed the state script.) When CA Automation Point is not monitoring the console, the pause script places the console in the mode in which the operator normally uses it. CA provides a sample pause script called `pause.scr` on the CA Automation Point distribution media.

When the screen is full, the messages on a JES3 console wrap around, overwriting the older messages. A dashed line separates the older messages from the newer ones. CA Automation Point follows the dashed line when processing JES3 messages.

Message Processing for JES3 Consoles

This is how CA Automation Point processes JES3 console messages:

- CA Automation Point recognizes the first line of a message by the time stamp. If a message line has no time stamp, that the line is assumed to be a continuation of a previous line. CA Automation Point concatenates the lines and processes the message through rules.
- If a message is longer than two lines, CA Automation Point processes the first line through rules. The display characteristics of continued message lines (such as color, prefix, and whether the message should be displayed or suppressed) are set to match the display attributes of the first line of the message.

CA Automation Point recognizes both JES3 and z/OS messages on a JES3 console. The following points indicate how CA Automation Point determines whether a message is a z/OS or JES3 message:

- A z/OS message has the string R between columns 10 and 19. When CA Automation Point recognizes a z/OS message, it parses the message and places the information in these environmental variables:

- &HOSTTIME
- &JOBNAME
- &REPLYID
- &MSG

Important! CA Automation Point recognizes the "*" and "@" action message indicators for z/OS messages on a JES3 console.

- CA Automation Point treats any other message as a JES3 message. It parses the message and places the information in these environmental variables:

- &HOSTTIME
- &MSG

In this case, the &JOBNAME environmental variable always contains the JES3 value.

VSE Consoles

This section discusses CA Automation Point support for VSE consoles.

ASI Facility

When configuring CA Automation Point to support VSE consoles, you can set up your hardware to use VSE's Automatic System Initialization (ASI) Facility. When you use this facility, you can configure hardware for CA Automation Point automatically by reinitializing VSE. The ASI Facility lets you redefine the system console on the supervisor parameters command so that the console device address is automatically assigned to CA Automation Point during IPL. For example, the following command sets 0F3 as the SYSLOG console address:

```
0F3, $$A$SUP1, P, LOG
```

Note: The command must be the first statement in the ASI IPL procedure.

To be controlled by CA Automation Point, a VSE console must meet the following requirements:

- The console must not be in message redisplay mode. If it is, CA Automation Point does not see any new messages and rejects most VSE commands that would otherwise be issued.
- The console must operate in non-delete mode.

Scripts

The first time CA Automation Point recognizes this session as coming from a VSE console, it executes the VSE state script for this session (if you have defined one).

If CA Automation Point either detects a change in any of the parameters displayed on line 24 (ACT_MSG, PAUSE, SCROLL, MODE) or finds the text SUSPEND on line 24, it executes the VSE state script again.

If you have defined a pause script for this session and you select the Pause option from the Cmdarea menu, CA Automation Point executes the pause script. When CA Automation Point is not monitoring the console, the pause script places the console in the mode in which the operator normally uses it.

Message Handling for VSE Consoles

CA Automation Point analyzes the message prefix to determine whether a line on the screen is a command or a message. If CA Automation Point finds a reply ID and a partition name (such as Fn, AR, SP, or BG), it assumes that the line is a message. Otherwise, CA Automation Point treats the line as a command.

Important! CA Automation Point must have control of the console to efficiently process the messages that appear on it. Enter commands for a session from a CA Automation Point command area.

Note: Mainframe software that changes the behavior of the VSE console may disable message processing. If you are running such software, you may need to use REXX to control the console.

WARNING! Do not issue K commands from CA Automation Point that directly affect the emulation window (such as K E,n). Other K commands (such as K Q and K E,D which affect the queue) are acceptable.

Message Parsing

CA Automation Point parses messages on VSE consoles as follows:

- Treats the text beginning in column 10 as the message ID and places the text in the &MSG environmental variable
- Places the three-digit reply ID in the &REPLYID environmental variable
- Places the partition number and the reply ID in the &JOBID variable

Command Echo Parsing

CA Automation Point parses command echoes as follows:

- Treats the text beginning in column 2 as the message ID and places that text in the &MSG environmental variable
- Sets the &JOBID and &REPLYID environmental variables to null

Message Lines

When columns 2 and 3 of a message line are blank, CA Automation Point assumes that the message is continued from a previous line and does not pass the continued message text to its rules for processing.

The display characteristics for continued message lines (such as color, prefix, and whether the message should be displayed or suppressed) are set to match the display attributes of the first line of the message.

When the last message line is full, CA Automation Point issues a K command to delete messages that it has already seen from the screen.

When CA Automation Point finishes processing a highlighted message, it immediately issues a K E,*n* command to clear the message from the screen.

Action Message Handling for VSE Consoles

CA Automation Point also supports action message handling for highlighted messages on VSE consoles. Action message handling works the same way for VSE consoles as it does for SYSPLEX, MCS, and RCS consoles.

z/VM Consoles

This section discusses how CA Automation Point handles z/VM consoles.

Message Handling for z/VM Consoles

CA Automation Point treats each line on the z/VM console as a message, with each line containing no more than 132 characters. Messages that must be split for display purposes are not spliced back together for CA Automation Point processing.

To determine whether a session is running on a z/VM console, CA Automation Point looks for one of the following status indicators in the last line of the screen:

- RUNNING
- HOLDING
- MORE
- VM READ
- CP READ
- NOT ACCEPTED

State scripts are not commonly used with z/VM sessions. If the status is NOT ACCEPTED, CA Automation Point executes a z/VM state script for the session, if you have defined one.

To define a VM state script

1. Create the VM.SCR file in the %AP_SITE%\MyFiles directory that contains the actions to take for the NOT ACCEPTED condition. (A sample vm.scr file is not provided because it is rarely used.)
2. In Configuration Manager, navigate to the session definition panel for the z/VM session (Expert Interface, Automation, Session Definition Sets, Sessions, Automation Point Session Definition).
3. Select Customize Scripts.
The Customize Scripts dialog appears.
4. Assign this new state script (VM.SCR) to the z/VM state.

When CA Automation Point encounters the RUNNING status indicator, it is in a valid automation state. If the z/VM logon screen at your site displays the indicator, CA Automation Point processes the screen data through rules. In such a case, INIT or state scripts cannot log on to z/VM, although you could use CA Automation Point rules to execute a logon script.

Before it processes each message from the z/VM screen, CA Automation Point checks for the following conditions:

- If the screen contains either the "HOLDING" or the "MORE..." status indicator, CA Automation Point automatically issues a PA2 keystroke to clear the screen.
- If the console is not in z/VM state or any other defined state, CA Automation Point starts the INIT state script, which tries to reinitialize the session.

CA Automation Point processes all messages on the screen. When it reaches the last line of the screen, it looks for the HOLDING or MORE status indicators. If the screen contains either indicator, it rechecks the status of the session.

Message Parsing for z/VM Consoles

As it processes each line on the screen, CA Automation Point analyzes that line to determine where the message ID is and sets environmental variables. CA Automation Point parses messages as follows:

- Secondary Console Image Facility (SCIF) messages

If the message has a colon (:) in column 9 or 10 and non-blank data in columns 1 through 8, CA Automation Point treats the message as a SCIF message. It also does the following:

- Places the data in columns 1 through 8 in the &JOBID environmental variable
- Increments the "beginning of message text" to the first non-blank character past column 10

In the following sample message, VTAM2 is the value that goes into the &JOBID variable and TIME IS 09:07:25 EDT TUESDAY 10/12/90 is the message text.

```
VTAM2      : TIME IS 09:07:25 EDT TUESDAY 10/12/90
```

- CP messages

If the message has an attribute byte in column 1, CA Automation Point treats the message as a CP message (issued by the MSG or WNG commands). It then searches the message for the strings MSG FROM xxxxxx or WNG FROM xxxxxxxx. If either string is present, it places the value of xxxxxxxx in the &JOBID variable. (If neither string is present, this variable is set to null.) For CP messages, CA Automation Point increments the "beginning of message text" to the first non-blank character after xxxxxxxx.

In the following sample message, "Intervention-required" is the message ID, and VM3812B is the value that CA Automation Point places in the &JOBID environmental variable.

```
MSG FROM VM3812B : Intervention-required on 3812:
check paper path
```

- Time-stamped messages

CA Automation Point looks for a time-stamped message to contain a colon in columns 3 and 6 and two-digit numbers beginning at columns 1, 4, and 7. If the message contains these characters, it places the value of the time stamp in the &HOSTTIME environmental variable and treats the next non-blank character string as the message ID. The parsing logic described applies to *all* z/VM message types.

The following is a sample time-stamped message. Because the message contains a two-digit number starting in column 1, CA Automation Point places the time stamp in the &HOSTTIME variable and treats HCPCFC003E as the message ID.

```
09:02:19 HCPCFC003E Invalid option - HILITE
```

- Blank messages

CA Automation Point ignores messages that contain only blanks or include only a JOBID or HOSTTIME value. For example, CA Automation Point ignores these messages:

```
10:34:11  
MSG FROM VM3812B:
```

- All other messages

Some messages (such as command echoes, most CP command replies, and resource status messages) do not fit into any of the previously described categories. When parsing the messages, CA Automation Point uses the first word as the message ID.

In the following examples, SEND and LOGON/JOB are the message IDs:

```
SEND VTAM2 Q T  
LOGON/JOB INITIATION - INVALID PASSWORD
```

z/OS Guest Consoles

If you intend to use CA Automation Point to manage a z/OS session that runs as a guest under z/VM, you may need to configure the z/VM BREAK key to ensure that the z/VM console session operates smoothly. The BREAK key allows you to toggle between the z/VM console and the z/OS console.

You cannot send a key operation intended for z/OS using the key configured as the z/VM BREAK key because z/VM intercepts the key first. CA Automation Point requires that the BREAK key be set to Clear, which neither CA Automation Point nor the operator uses.

Note: The default z/VM BREAK key is PA1, which CA Automation Point does not use; however, a z/OS operator can use the PA1 key to retrieve and reissue the previous z/OS command when the console is under manual control. IBM suggests setting the z/VM BREAK key to a less commonly used key, such as PA2; however, the CA Automation Point MCS console scripts use the PA2 key to reset the console if an error state occurs.

Determining the Current z/VM BREAK Key Setting

To determine the current setting of the VM BREAK key, toggle to the z/VM console and enter the following command:

```
QUERY TERMINAL
```

The current BREAK key (BRKKEY) setting displays in the command response.

Changing the BREAK Key Setting to Clear

If the BREAK key is not set to Clear, you can set it temporarily or permanently.

To set the key temporarily, issue the following CP command:

```
TERMINAL BRKKEY CLEAR
```

The temporary setting is useful for verifying that your terminal type supports the modified BREAK key.

Note: If an error message displays, contact your z/VM systems programmer for assistance.

To set the key permanently, edit the PROFILE EXEC file, replacing the *keyname* value in this CP command with CLEAR:

```
TERMinAl BRKKEY keyname
```

(To use the z/VM XEDIT editor, issue the X PROFILE EXEC command. Be sure to file (save) the modified PROFILE EXEC file.)

Test the new BRKKEY setting by logging off the VM user ID and logging on again. The z/VM logoff causes the guest z/OS operating system to shut down, so you may want to schedule the test.

MCS Session Automation and the OSA-ICC DHD Option

When the Deferred Host Disconnect (DHD) option for the OSA-ICC controller is enabled, the OSA-ICC controller defers reporting the MCS console disconnect status to z/OS for a period specified by the DHD option.

If CA Automation Point's MCS session reconnects before the deferred disconnect time expires, then the OSA-ICC controller automatically simulates the 3270 CLEAR key, triggering z/OS to reformat the console screen and to continue sending console messages. In this case, CA Automation Point's MCS session automation automatically resumes upon reestablishing a connection.

If the deferred disconnect time expires before CA Automation Point's MCS session reconnects, the OSA-ICC controller signals z/OS that the MCS console is disconnected and the MCS console status is set to OFFLINE. In this case, once CA Automation Point's MCS session reconnects, the operator must manually vary the MCS console back to the ONLINE status before MCS session automation can be restarted.

Note: For more information on the DHD option, see the IBM Redbook *OSA-Express Integrated Console Controller Implementation Guide*.

Chapter 4: Asynchronous Host Sessions

This chapter describes how you set up your asynchronous sessions and manage asynchronous communication.

Asynchronous Sessions

Data streams that originate from any host or device that communicates asynchronously using the RS-232 protocol and asynchronous terminal type must be defined as *asynchronous* sessions.

If CA Automation Point supports the asynchronous console for which you are defining a session—that is, if the console is one of the valid console types that you can specify in the Console Type field in the Session Definition dialog—you can define the asynchronous session as an *automated* session (meaning that CA Automation Point automates the session directly through rules). *If CA Automation Point does not support your console type, you cannot define an automated session.*

Note: CA Automation Point can control *non-automated* asynchronous sessions indirectly through REXX programs.

Configuring CA Automation Point for asynchronous sessions is similar to configuring it for 3270 sessions. For more information about configuring your asynchronous sessions, see the Configuration Manager HTML help.

Hardware Connections

Before connecting your hardware, check the console connections. If they do not resemble telephone wire, assume that the Data Set Ready and Clear To Send signals are present.

When you have determined which signals are present and have set any necessary communication settings, connect the host to your workstation and plug the connector from the host into the appropriate port on your workstation.

Using Serial COM Ports

To monitor asynchronous hosts or perform alphanumeric paging, you must use serial COM ports. Historically, computers were configured with two COM ports on the back of the machine. Some newer computers no longer provide such COM ports. If your computer has no COM ports, or if two ports are insufficient for your needs, you can use a network-attached serial port expansion device to increase the number of COM ports.

A network-attached serial port expansion device is connected to the corporate network. CA Automation Point (which also must be connected to the corporate network) communicates to the serial port expander through TCP/IP socket connections. Serial devices are connected to the serial port expander. CA Automation Point communicates to the serial devices through the serial port expander.

To operate with CA Automation Point, one must configure the serial port expander to act like a native COM port. This typically involves installing driver software from that vendor. The driver simulates a COM port, but then sends all of the data that it receives to the serial port expander over the TCP/IP network.

CA Automation Point is configured and operated as if it were using a native COM port on the back of the computer. You must follow the installation and configuration documentation from the vendor of the serial port expander to setup the expansion device and the driver software associated with that device. Contact CA Support for information on a serial port expansion device that has been fully verified for use with CA Automation Point.

How You Switch Asynchronous Signals with a Null Modem

To achieve acceptable asynchronous signal conditions, you may need to install a device known as a *null modem*. Most asynchronous devices communicate using a wiring protocol called RS-232. One side of a connection is called the data terminal and the other side is called the data set. As long as your host device is wired as a data set and the COM ports are wired as data terminals, there should be no communication problems. However, if the host you are connecting to is wired as a data terminal, mismatching signals will result. Consequently, you need to insert a null modem. Use the following components to establish this type of connection:

- 9-pin or 25-pin null modem (a null modem can also be purchased with a built-in gender changer)
- Standard cable
- 25-pin to 9-pin converter (you may not need this)
- Gender changer (if the connector of the standard cable is incorrect)

Note: When RS-232 signals are mixed up, one-way communication results.

How You Test Asynchronous Communications

Start CA Automation Point and jump to the session that you have just defined to verify that CA Automation Point can communicate with the host. If everything is working properly, you should see $Cn=R$ or $Dn=R$ (with n equal to a number from 1 to 8) in the status line of the session window.

Note: For more information about the status line, see the section [The Status Line](#) (see page 68) in this chapter.

Asynchronous Communication Problems

If CA Automation Point cannot communicate with the host, check the following items and use the status line to help you solve the problem:

- CTS signal
 - If you expect a CTS signal and you see either a minus sign (-) or an X as the third character in the display, verify that both sides of the cable are plugged securely into their respective devices. If they are, you may have a cable problem or you may need to insert a null modem into the line.
Note: For more information, see the chapter "[Establishing Host Connections](#) (see page 41)."
 - If you do not expect a CTS signal and see an X as the third character, change the Clear To Send setting in the communication settings for this session to NO. You should then see a minus sign (-) where the X used to be, indicating an acceptable state.
- DSR signal
 - If you expect a DSR signal and the fourth character in the display is either an F or a question mark (?), verify that both sides of the cable are plugged securely into their respective devices. If they are, check the Clear to Send indicator:
 - If the indicator is a question mark (?), the cable is probably faulty.
 - If the indicator is a minus sign (-), try installing a null modem. You may see the CTS indicator change to an equal sign (=), indicating that the CTS signal was present but required the null modem.
 - If you do not expect a DSR signal and see a question mark (?) as the fourth character in the display, change the Data Set Ready setting in the communications settings for this session to NO. You should then see an F where the question mark used to be, indicating an acceptable state.
- Cable connection

If you changed the values of both the Clear To Send and Data Set Ready settings to NO and you still cannot communicate with the host, try installing a null modem. If the null modem allows you to establish communication with the host, you may see the CTS and/or DSR indicators appear.

If none of the previous steps corrects the communication problem, discuss it with the communication specialist at your data center.

How You Verify Message Processing

After you establish communication with the asynchronous host, look at the Merged Messages window to verify that CA Automation Point is processing the console messages. If the window contains no messages, verify that you have specified a valid console type for your session.

Asynchronous Telnet Sessions

You can use CA Automation Point to automate an asynchronous Telnet client session using any of the VT series terminal emulators currently supported by CA Automation Point. Follow the steps in the next sections to set up and configure automated Telnet client sessions on the Automation Point Desktop.

How You Set Up Asynchronous Telnet Sessions

To set up an asynchronous telnet session, select TELNETCLIENT as the communication device in the RS232 settings for the session. Other settings for the telnet appear in the same dialog.

How You Test the Session

You should test the new asynchronous session before putting it into production.

To test the asynchronous session, start CA Automation Point and view to the AP Message Recall window.

If TCP/IP is not active on the system, the following message displays:

```
AXC0983E TCPIP not available
```

The following message indicates that the connection has been established to the host:

```
AXC1353I TELNETCLIENT INITIATED - Result: success Session: sessname Host: Telnet
```

If you do not see this message, verify that the Telnet host name specified in the communication settings is correct. If you are not successfully connected to the Telnet host, the status line continuously displays X-SYSTEM until a successful connection is established.

Asynchronous SSH Sessions

You can use CA Automation Point to establish up to 175 automated asynchronous session connections using version 2 of the SSH (Secure Shell) protocol (SSH-2). The SSH-2 client protocol implemented by CA Automation Point uses TCP/IP to establish a secure terminal connection to the remote host running a compatible SSH-2 server. You can define this terminal connection using any of the VT series terminal emulators currently supported by CA Automation Point.

Although the SSH-2 protocol defines a suite of applications used to send data securely across a TCP/IP network, CA Automation Point currently allows you only to establish a secure terminal session to the SSH-2 server. This terminal session functionality was designed to closely match the capabilities of the existing CA Automation Point Telnet interface. If you are upgrading from a previous release of CA Automation Point and have Telnet sessions already defined, you should be able to transition these sessions to our SSH-2 client implementation with minimal effort. As part of this design, CA Automation Point currently only supports the user authentication methods that require the user to specify login credentials directly (which mirrors the user authentication used for Telnet sessions). The supported SSH-2 user authentication methods (as defined by both the SSH-2 protocol and additions to the SSH-2 specification) are: "password" and "keyboard-interactive." Both of these user authentication mechanisms require the user to specify a valid login name and optional password to confirm their identity before a terminal session is established.

Security Features of the SSH-2 Protocol Implementation

One key advantage of using the SSH-2 protocol for asynchronous terminal connections is the combination of encryption and data integrity mechanisms defined by this protocol. The SSH-2 protocol defines a preliminary negotiation phase, during which the client and server agree upon both the encryption and data integrity algorithms to use for the current connection. After this negotiation phase is complete, all future communications between client and server are encrypted using the negotiated encryption algorithm. In addition, each data segment sent across the TCP/IP network is verified by the recipient to ensure that the contents of the data segment have not changed since the initial creation of the data segment by the original sender. This verification is performed using the data integrity algorithm selected during the initial negotiation phase.

The following encryption protocols are available for selection during the SSH-2 protocol negotiation phase: Blowfish, AES-256, AES-192, AES-128, and Triple DES. The following data integrity algorithms are available for selection during the SSH-2 protocol negotiation phase: HMAC-SHA1, HMAC-MD5, HMAC-SHA1-96, and HMAC-MD5-96. The selection of encryption and data integrity algorithms to use is negotiated between CA Automation Point and the SSH-2 server based upon the preference order of the algorithms listed previously. For example, if the SSH-2 server supports the Blowfish protocol, this protocol will always be selected because it is listed first in the SSH-2 client's (CA Automation Point's) encryption algorithm list. These algorithm lists are not configurable within CA Automation Point. If you wish to connect CA Automation Point to an SSH-2 server using a specific series of protocols, you should define the SSH-2 server to only advertise those protocols you wish to use.

In addition to providing encryption and data integrity mechanisms, CA Automation Point also employs host verification as outlined by the SSH-2 protocol. Each SSH-2 server is responsible for maintaining a public host key that uniquely identifies itself to SSH-2 clients. This public host key is securely sent to each SSH-2 client during the initial protocol negotiation phase. CA Automation Point stores the host key reported during the first connection request for a particular session and uses this stored key to validate the transmitted host key during future connection attempts. If the public host key transmitted by the SSH-2 server does not match the previously stored host key associated with this session, a warning message is displayed, and the user is given a choice whether or not to continue with the current connection attempt. This host verification helps to mitigate potential "man-in-the-middle" (MITM) cryptographic attacks by ensuring the identity of the SSH-2 server before sending any sensitive data.

Set Up Asynchronous SSH Sessions

You set up an asynchronous SSH session from Configuration Manager.

To set up an asynchronous SSH session

1. From the Session Definition Sets dialog, right-click the Sessions node and select Add Session to create a new session.

The Automation Point Session Definition dialog displays.

2. In the Console Type field, select ASYNCH, then click Customize Session Settings.

The Asynchronous Session Settings dialog displays.

3. Select SSHCLIENT as the communication device.
4. Select other settings as desired, then click OK to save session settings.

For more information on the options available for an asynchronous SSH session, see the Configuration Manager HTML help.

Test the Session

To make sure the new asynchronous host SSH session is established, start CA Automation Point and select the AP Message Recall window.

If the connection to the SSH host is successful, the following message displays in the AP Message Recall window:

```
AXC1361I SSHCLIENT INITIATED - Result: success Session: sessname Host: hostname
```

If you do not see this message, verify that the SSH host name specified in the communication settings is correct. If you are not successfully connected to the SSH host, the status line continuously displays X-SYSTEM until a successful connection is established.

If TCP/IP is *not* active on the system, the following message displays:

```
AXC0983E TCPIP not available
```

Send Key Operations to Asynchronous Sessions

When CA Automation Point is monitoring an asynchronous terminal session, you can send control key operations to that session using CA Automation Point rules, scripts, or REXX procedures. For example, pressing Ctrl+K from any asynchronous session sends a vertical tab.

For a list of the key operations available for asynchronous sessions and a description of how to send those operations to a session, see the appendix "[Customizing Special CA Automation Point Files](#). (see page 457)"

You must halt automation before entering keystrokes unless you specify View with Keyboard Input in the Automated Terminal Session Window field for the session. However, we recommend that you leave this setting at View Only; otherwise, the characters that you type could be confused with those that CA Automation Point generates while performing its automation tasks.

Manage Asynchronous Communication

CA Automation Point can communicate with many asynchronous hosts or devices conforming to the RS-232 wiring protocol. Communication is possible using a serial communications port accessible to your CA Automation Point workstation.

The Status Line

In an asynchronous console session, the bottom line of the terminal screen displays information about the status of message processing for that session. The status line has three components.

The following illustration shows you what the status line might look like for a VT320 console session:

```
C1=R          (01,020)      ?
 1            2            3
```

Callout	Explanation
1	<p>Communication status display</p> <p>The characters C1 indicate that this session is using the COM1 serial port on the workstation. The equal sign (=) indicates that the port is sending a CTS signal, and the character R indicates that the remote host is ready to communicate with CA Automation Point.</p>
2	<p>Current row and column position of the cursor for VT52, VT100, or VT320 sessions. In the example, the cursor is located at row 1, column 20.</p>
3	<p>The message parsing status display</p> <p>The question mark (?) indicates that this session has not received any messages to process.</p>

The Asynchronous Communication Status Display

The asynchronous communication status display consists of the characters in the left corner of the status line. These characters indicate the state of the communication device for the current session. Interpreting these characters can help you resolve most types of asynchronous connectivity problems.

The asynchronous communication status display includes four characters:

- A letter indicating the type of communications device the session is using. The letter is one of the following:
 - C for COM port
 - M for Memory session
 - T for Telnet
 - Z for SSH

- A number between one and nine, which is the number of the port or channel in use.
- A character indicating the status of the Clear to Send (CTS) line for this asynchronous connection. You may see any of these indicators:

Character	Meaning
=	The session is detecting a CTS signal. This is an acceptable state for the indicator.
-	No CTS signal is present, but CA Automation Point expects none because you set Clear to Send to No in the communication settings for this session. This is an acceptable state for the indicator.
X	No CTS signal is present, but CA Automation Point expects one because you set Clear to Send to Yes in the communication settings for this session. This is an unacceptable state for the indicator; it means that CA Automation Point cannot communicate with the host associated with this session.

Note: When X displays, take one of these actions:

- Determine why no CTS signal is present by checking cable and communications port connections.
 - Change the value of the Clear to Send setting to No if no CTS signal is expected.
- A character indicating the state of the Data Set Ready (DSR) signal. You may see any of these indicators:

Character	Meaning
R	The session is detecting a DSR signal. This is an acceptable state for the indicator.
F	No DSR signal is present, but CA Automation Point does not expect one because you set Data Set Ready to No in the communication settings for this session. In effect, this setting forces the DSR line into an acceptable state.
?	No DSR signal is present, but CA Automation Point expects one because you set Data Set Ready to Yes in the communication settings for the current session. This is an unacceptable state for the indicator; it means that CA Automation Point cannot communicate with the host associated with this session.

Note: When ? displays, take one of these actions:

- Determine why no DSR signal is present by checking cable and communications port connections.

For more information, see the chapter "[Establishing Host Session](#) (see page 41)s."

- Change the Data Set Ready communications setting to No if no DSR signal is expected.
- A plus sign (+) indicating that a Carrier Detect (CD) signal is present. If CA Automation Point is not receiving a CD signal, this position is blank. This indicator is for your information and does not affect the ability that CA Automation Point has to communicate with the host for this session.

Some cables, adapters, and host connections provide the CD signal for compatibility purposes; however, no CD signal is required if you are using a direct connection.

Example

For example, suppose that D3-R+ appears in the display. The indicators tell you that CA Automation Point is communicating with the host device for this session under these conditions:

- You are using the third accessible COM port.
- No CTS signal is present, but none is expected.
- A DSR signal is present and in an acceptable state.
- A CD signal is present and in an acceptable state.

The Message Parsing Status Display

For all types of asynchronous consoles, the right corner of the status line displays characters reporting the state of message parsing for the current session. Understanding the message parsing status display can help you solve problems when CA Automation Point does not generate the messages you expect.

If you have an ASYNCH console, the message parsing status display contains one character. (These console names designate how CA Automation Point processes the data on the screen. For more information, see the chapter "[Establishing Host Sessions](#) (see page 41).")

The characters you may see are as follows:

Character	Meaning
?	The session has not received any characters since CA Automation Point was started.

Character	Meaning
I	The session is receiving characters, but CA Automation Point is ignoring them because they do not fit the type of message that was requested. A session running on an ASYNCH console does not display the I character because all characters received by this type of console become part of a message.
H	CA Automation Point is trying to match the incoming characters as the message header for the message types requested. If the characters do not eventually complete the header, the H changes to an I and CA Automation Point resumes searching for the header.
F	CA Automation Point matched the appropriate header and is now filling the message text. Filling continues until the session receives the appropriate termination characters.
R	The session received the appropriate message termination characters and has released the message to the CA Automation Point rules processor.
O	The message being filled, which contained more than 512 characters, was released to the CA Automation Point rules processor because of an overflow condition. CA Automation Point ignores the extra characters (characters 513, 514, and so on) until it receives the beginning of the next message.
T	CA Automation Point was filling a message but did not receive any characters within the corresponding timeout period. The timeout period is ten seconds for ASYNCH consoles. If the timeout period expires, the session forwards the message text that it has already received to the CA Automation Point rules processor.

As message traffic flows between the host and CA Automation Point, the message processing state indicator changes. Normally, the character is I until the session receives a message header. The character changes to H as CA Automation Point checks the header, and then changes to F when CA Automation Point begins filling the message text. When the session receives termination characters, the indicator changes to R and CA Automation Point releases the message.

Normally, the O and T indicators do not appear. If you see either indicator, verify that the corresponding session window is receiving messages intact. Communication problems with the host may cause message text to become fragmented.

Communication Problems

You can use the information on the status line of your asynchronous console to resolve most common problems associated with asynchronous connections, including missing and mismatched signals. Because CA Automation Point does not try to communicate with the host until all required signals are present (or forced), you must achieve acceptable signal conditions before resolving other asynchronous configuration issues (such as baud rate and parity).

Asynchronous Console Management

CA Automation Point performs message processing on asynchronous consoles by monitoring the incoming character stream and extracting messages that conform to message criteria for the current console type. After CA Automation Point extracts a message, that message goes to the rules processor, just as any other console message would. At that time, rules can take whatever actions are specified, including sending one or more replies back to the session that generated the message.

Asynchronous Console Types

CA Automation Point supports message processing for the asynchronous console types shown in the following table.

Note: These terms are not industry-standard console names. Rather, they refer to the Console Type setting for the session, which designates how CA Automation Point should process the data on the screen.

Asynchronous Console Type	Description
ASYNCH	Generates messages matching the ASYNCH message criteria
TANDEM	Generates messages matching the EMS format
TANDEMALL	Generates EMS messages, but passes text not matching the EMS format for processing through rules as an ASYNCH message
VAX	Generates messages that match the OpenVMS OPCOM message criteria
VAXALL	Generates OpenVMS OPCOM messages, but passes text not matching the OPCOM format for processing through rules as an ASYNCH message
DTX	When attached to the DataFrame environmental monitoring product, this console type parses the special message format used by DataFrame and creates standard CA Automation Point messages

Message Criteria for Asynchronous Consoles

When processing an incoming character stream on asynchronous consoles, CA Automation Point looks for the following:

- A header to indicate that a message has started
- The text of the message
- A terminating condition or trailer to indicate that the message has ended

ASYNCH refers to the CA Automation Point asynchronous console driver. The console types in the next few sections are specific to various asynchronous formats. Specify the ASYNCH driver if none of the specific drivers apply to your console.

To the CA Automation Point console driver, a message begins with any character and continues up to the first new line (carriage return or line feed) character. If the console sees no new line character within ten seconds of receiving the last character, it releases the message text received up to this point to the CA Automation Point rules processor.

Define the console type as ASYNCH in the following situations:

- When CA Automation Point is connected to an asynchronous system such as UNIX or Linux
- When CA Automation Point is connected directly to another workstation running CA Automation Point, such as a workstation dedicated to voice processing
- When CA Automation Point is monitoring a serial printer port on a host
- When the data stream does not include escape (ESC) sequences

If an ASYNCH console receives a OpenVMS OPCOM message, each of its lines generates an individual message.

For example, suppose that the console receives the following sample message text:

```

%OPCOM 01-JAN-1994 12:00:00:00 %
Message from user JOB_CONTROL
%JBC-E-SYMDEL, unexpected symbiont process termination

>TEST

```

CA Automation Point treats each line of message text on an ASYNCH console as a separate message and sends each message to the rules processor. Given the sample message text, CA Automation Point generates five messages; for example, the first message sent to the rules processor is:

```

%OPCOM 01-JAN-1994 12:00:00:00 %

```

In this case, the OPCOM header is a message because ASYNCH message criteria do not distinguish between headers and other parts of the incoming character stream.

Readlog Technique

Readlog is a CA Automation Point technique for acquiring external events on remote systems that have no specialized event monitor.

On UNIX systems use the following command:

```
tail -f
```

You can use a comparable command line tool on any asynchronous system to continually feed a system log or application log into CA Automation Point.

Chapter 5: Managing Sessions Using CA Automation Point Windows

This section contains the following topics:

[Using the Automation Point Desktop](#) (see page 75)

[Understanding Function Windows](#) (see page 78)

[Window Menu Options](#) (see page 79)

[Customizing Menus](#) (see page 82)

[Customizing Screen Fonts](#) (see page 94)

[Issuing Commands](#) (see page 98)

[Displaying Recalled Messages](#) (see page 101)

[Displaying Recent CA Automation Point Messages](#) (see page 103)

[Merging Messages from Managed Sessions](#) (see page 104)

[Displaying Log File Contents](#) (see page 105)

[Displaying Notification Server and Notification Manager Messages](#) (see page 106)

[Getting Information About CA Automation Point Messages](#) (see page 106)

[Displaying Graphs of System Information](#) (see page 106)

[Stopping CA Automation Point](#) (see page 107)

Using the Automation Point Desktop

When CA Automation Point starts, the first screen that displays is the Automation Point Desktop. The desktop displays *icons* that represent the terminal emulator and CA Automation Point function windows you defined in the session definition set.

Window icons on the Automation Point Desktop represent your CA Automation Point windows. If you want, you can minimize all these windows to icon size and enlarge a window for display only when you want to read its contents.

CA Automation Point updates displayed windows periodically, regardless of their size, to reflect activity in the session for each window (such as new messages being displayed or new commands being issued). To enhance performance, the background windows are updated less often than the window in focus.

Layouts

After you have moved or resized your session and function windows, you can save the layout and restore it later as needed. A layout contains the position, size, and font specification for session, function, and dialog windows. This feature is common to the Automation Point Desktop and Remote Viewer applications. Remote Viewer-specific features are described in the section [Remote Viewer Layouts](#) (see page 78).

Notes:

- You must first save a layout before you can load one.
- If an operation being performed as a result of loading a layout cannot be processed (for example, a window does not exist in a currently active session definition set), it is silently ignored.
- After a successful save or load operation, the AXC2001I or AXC2002I messages are displayed, respectively.
- We strongly recommend that you let the user interface stabilize before performing another layout operation, especially when using Remote Viewer, where establishing of a connection can take a long time, depending on network latency.
- Layout name length is limited to 259 characters.
- Layout names `"*New*"` and `"*Empty*"` are reserved and cannot be used.

Working with Layouts

To manage your layouts, select Layouts from the Action menu. The Layout dialog displays, listing currently saved layouts.

- To save a new layout, select `*New*`, then click Save. You will be prompted for a name for the layout.
- To save your current layout to an existing name, select an existing layout and click Save.
- To load previously saved layout, select the layout name from the list and click Load.
- To restore default window positions according to the Sizing setting in the session definition set, select `*Empty*`, then click Load.
- To rename a layout, select the layout name and click Rename. You are prompted to enter the new name.
- To delete a layout, select the layout name and click Delete.
- To exit the dialog, click Close.

Command-line Option /l

The optional command line option `/l layout_name` lets you specify which layout should be loaded upon application startup.

If you specify `/l *Empty*`, no layout is initially loaded:

- For Automation Point Desktop, window positioning is based on the session definition set.
- For Remote Viewer, only the Remote Viewer Msg Window loads.

If you do not specify the `/l` option,

- For Automation Point Desktop, a layout named Default is loaded, if it exists.
- For Remote Viewer, no layout is loaded.

Examples:

```
axc2p /l "My layout"
```

```
apview /l Shift2
```

Active Layout

The optional `SCREEN_SAVE` and `SCREEN_LOAD` key operations use the *active layout* from which to save to or load layout information. The active layout is set by the following actions:

- When you specify the `/l layout_name` command line option. If this command is omitted, CA Automation Point uses the default value specific to the application
- When you save or load a layout or when you rename the current active layout from the Layouts dialog

Note: When the active layout is set to `*Empty*`, `SCREEN_SAVE` and `SCREEN_LOAD` operations do not perform any actions.

Remote Viewer Layouts

You can save your currently active user interface layout and restore it later as needed.

For general information about layouts, see the section [Layouts](#) (see page 76).

In Remote Viewer, a layout also contains connection information: which session on which host belongs to a specific window. When you load a layout in Remote Viewer, the application tries to restore the connection to the Automation Point Desktop that was connected at the time the layout was saved. Loading an empty layout closes all currently open connections.

Remote Viewer layouts are separate from Automation Point Desktop layouts; if you use both applications on the same machine, they cannot share layouts.

Security Notes:

- For connections to remote Automation Point Desktop hosts, username and password authentication may be required by the remote session. You are prompted a username and password when preparing a layout by opening the connections.
- Saving a layout does not store any usernames or passwords. However to facilitate quick loading, logon sets are stored that keep information about which sessions used the same username and password to connect to a specific host. When loading a layout, you are prompted for username and password for only one session from a given logon set. This logon information is then reused for all sessions that had the same username and password when they were saved. If the supplied username or password is not valid, you are prompted to re-enter them. If you choose to cancel at this point, no further sessions from that logon set are loaded.

Understanding Function Windows

The CA Automation Point function windows let you monitor and control CA Automation Point processing. CA Automation Point provides the function windows listed in the following table:

Function Window	Purpose
Command	Lets you issue commands to sessions that CA Automation Point manages. Some CA Automation Point function windows also have command areas which function like the Command window.
Merged Msg	Displays the messages that CA Automation Point collects from the sessions it manages and upon which CA Automation Point rules act. This window contains a command area and areas displaying the most recently issued action and normal messages.

Function Window	Purpose
Action Msg Recall	Displays the last 500 lines of outstanding action message text—messages waiting for an operator's response—received from sessions that CA Automation Point monitors. An <i>action message</i> is a message that requires a response from CA Automation Point or, in this window, from an operator.
AP Msg	Displays 20 lines of the most recently issued CA Automation Point messages. This window contains a command area.
AP Msg Recall	Displays the last 500 lines of message text that CA Automation Point issued.
Normal Msg Recall	Displays the last 500 lines of non-highlighted message text that CA Automation Point received from the sessions it monitors.
AP Log	Displays the contents of the current message log file.
Host Log	Displays the contents of the current host message log file.
Plot	Displays information about system operations shown as bar or line graphs. Create these graphs using the CA Automation Point Plot feature.
AP Notification Messages	Displays the last 500 messages generated by the notification server, notification (NMFIND) request, or Answer Tree application.
CA OPS/MVS Messages	Displays WTO messages generated by CA OPS/MVS.

Each of the CA Automation Point function windows is optional, except for the AP Msg Recall window, which always displays. You can configure CA Automation Point to display only the function windows that you want to see.

Window Menu Options

The CA Automation Point function windows and terminal emulator windows have a common Window pull-down menu listing options for managing your windows. To display the menu, click Windows on the menu bar.

The following table describes the Window menu options:

Option	Purpose
Jump to window	Displays the CA Automation Point window you specify. Selecting this option displays the Jump to Window dialog, which contains a scrollable list of available windows.

Option	Purpose
Next window	Displays the next window without giving you a list of windows from which to choose.
Increase font	Enlarges the current window and the size of the text in the window.
Decrease font	Reduces the current window and the size of the text in the window.
Select font	Displays the Select Font dialog, from which you choose a new font and size for the current window. For more information, see Changing the Size of a Window (see page 81) in this chapter.
Close this window	Removes the current window from the CA Automation Point desktop without affecting automation.

Selecting a Window

To view a CA Automation Point function window or a terminal emulator window, select its icon from the CA Automation Point desktop using the mouse or keyboard. When you choose a window, CA Automation Point displays the window in normal size unless you changed the size previously. If you change the size of a window, CA Automation Point displays the window in its new size each time that you select it.

You can jump between windows on the CA Automation Point desktop, regardless of their current size using, any of these methods:

- Click the icon for the desired window. To enlarge the window, double-click the window's icon. If you click the window name instead, CA Automation Point selects the window without enlarging it.
- Press the key defined for Jump (Alt+J is the default) repeatedly, jumping to the next available window until the CA Automation Point desktop normalizes your desired window. You can also choose Next Window from the Window menu.

Closing a Window

Closing a window does not eliminate that window's function, or stop the automation occurring in the window; it only removes the session from the desktop. You can reopen the window using the Jump to Window dialog. To remove a window's function, delete the session from the session definition set.

To close a window from the CA Automation Point desktop, select the window that you want to close, and then choose Close This Window from the Window menu.

Changing the Size of a Window

When you change the font that a window uses to display text, the size of the window changes automatically to accommodate the larger or smaller text.

Note: All of the window text fonts are large enough to read easily.

CA Automation Point gives you several options for choosing fonts:

- Select the next larger font in one of these ways:
 - Press the Ctrl+right arrow keys.
 - Choose Increase Font from the Window menu.
- Select the next smaller font in one of these ways:
 - Press the Ctrl+left arrow keys.
 - Choose Decrease Font from the Window menu.
- You can choose a *specific* font by choosing Select Font from the Window menu. Select one of the fonts listed in the Choose Font dialog.

Note: You can use the AXCUTIL2 utilities program to select other default screen fonts.

Displaying Multiple Windows

To display multiple windows in readable size

1. Choose the windows you want to display.
2. Select one of those windows.
3. Press the Ctrl+left arrow keys repeatedly until the window is the size you want.
4. Repeat Steps 1 and 2 for the next window you want to display.
5. Move the selected windows to the positions where you want them on your screen.

Note: You can select multiple windows to display together, but displaying too many windows at one time clutters the CA Automation Point desktop and makes it more difficult to use.

Moving Displayed Windows

When a window is opened from an icon, it displays in the upper-left part of the screen. If you move the window to a new location, you should save your desktop to ensure that the window displays in its new location each time you open it. The window displays in this new location until you move it again.

Scrolling Information in Windows

When a CA Automation Point function window displays too many recalled messages to fit on the screen, the scroll bar on the right side of the window indicates the current position in the file. You can scroll through the information using the mouse or the keyboard.

To scroll information using the keyboard, press these keys:

Key	Operation
Home	Go to the top of the file.
End	Go to the end of the file.
Page Up	Scroll backward one screen.
Page Down	Scroll forward one screen.
Up arrow	Move up one line.
Down arrow	Move down one line.
Left arrow	Scroll one column to the left.
Right arrow	Scroll one column to the right.
Tab	Toggle between the list management and command areas of the display. (Remote sessions only)
Enter	Return to the left margin on the screen.

Customizing Menus

CA Automation Point includes a set of predefined menus for all CA Automation Point function and session windows. By default, the Configuration Manger Session Definitions Sets dialog automatically selects the menu designed for the session of its type. Menu options simplify manual key entry by providing the console operator with point-and-click access to common CA Automation Point desktop operations.

Besides providing built-in menus for built-in operations, CA Automation Point also allows you to customize menus, extending the available menu options. You can add menu options that invoke site-specific commands or operations with the same point and click convenience from the CA Automation Point desktop. Using menu options to issue predefined keystroke strings also minimizes operator key entry and usage errors.

Understanding the Menu System

The CA Automation Point menu system controls the menus and menu options that are available for each CA Automation Point window on the desktop. Statements in the following two files define a menu system:

- **axc2p.mnu** is a file distributed by CA Automation Point and installed into the *installdir*\Distrib directory. This file defines the default menus for CA Automation Point desktop windows.

Note: axc2p.mnu is a fixed file. Do not modify it.

- **user.mnu** is a file created by the user and located in the %AP_SITE%\myfiles directory. This file defines user-customized menus.

Settings in the active Session Definition Set associate a menu with each session window or function window displayed on the CA Automation Point desktop.

You should never modify the axc2p.mnu file to customize menus. Instead, create a file named user.mnu in the %AP_SITE%\myfiles directory to specify all user menu customizations. User.mnu must be a plain ASCII text file. When defining a new menu, use Configuration Manager to override the default menu setting for a session definition or function window definition with a customized menu setting.

When you start CA Automation Point, it searches your site directory first for customized menu files before searching the files distributed with CA Automation Point. The user.mnu file (if it exists) is appended to the axc2p.mnu file before CA Automation Point creates the menu system. The CA Automation Point desktop displays windows and their associated menus according to settings in the active Session Definition Set.

Selecting Menus for the Function Windows

The menu displayed for each function window appearing on the CA Automation Point desktop is configurable from Configuration Manager through Session Definition Sets dialogs. Before a user-customized menu can appear as a selectable menu, it must be previously defined by MENU and SUBMENU statements in the user.mnu file as directed in sections titled, Adding New Menu Options to Existing Submenus, Defining a New Menu, and Adding New Submenus to Existing Menus.

If you do not want to customize the menu, CA Automation Point automatically selects from its default menu system a predefined menu for the function window based on its type. For example, if Terminal is set to 6530, then CA Automation Point selects the menu named SESSION_6530 for the default menu.

If you want to select a user-customized menu for a function window, follow these steps to configure the menu for a function window.

1. From Configuration Manager, click on Session Definition Sets to open the Session Definition Sets dialog. Expand the Session Definition Sets tree and double click on the function window you wish to configure.
2. From the CA Automation Point Function Window Definition dialog, click on the Menu edit box and select a menu from a list of defined menus. Click OK to save and close the dialog.
3. Click on Close to exit the Session Definition Sets dialog.
4. CA Automation Point must be recycled before a new menu displays on a desktop window.

Example:

When you create a new function window of type PLOT, CA Automation Point does not provide a predefined menu. Add the following MENU and SUBMENU statements to user.mnu to include 'plot' in the list of selectable menus that can be configured for a function window by the Session Definition Sets dialog.

```
MENU = plot,  
      SUBMENU = window,  
      SUBMENU = plot_control  
  
SUBMENU = plot_control,  
          NAME = (control),  
          ITEM = ('start plot', XCCMD('REXX "plot start"')),  
          ITEM = ('stop plot', XCCMD('REXX "plot stop"'))
```

Selecting Menus for Session Windows

The menu displayed for each session window appearing on the CA Automation Point desktop is configurable from Configuration Manager through Session Definition Sets dialogs. Before a user-customized menu can appear as a selectable menu, it must be previously defined by MENU and SUBMENU statements in the user.mnu file as directed in sections titled, Adding New Menu Options to Existing Submenus, Defining a New Menu, and Adding New Submenus to Existing Menus.

If you want to select a predefined CA Automation Point menu or specify a user-customized menu for a session window, follow these steps to configure a menu for a session window:

1. From Configuration Manager, click on Session Definition Sets to open the Session Definition Sets dialog. Expand the Session Definition Sets tree and double click on the session window you wish to specify a menu for.
2. From the Automation Point Session Definition dialog, click on the Menu edit box and select a menu from a list of defined menus. Click OK to save and close the dialog.
3. Click on Close to exit the Session Definition Sets dialog.
4. You must recycle CA Automation Point before a new menu displays on a desktop window.

If you do not want to customize the menu, CA Automation Point automatically selects from its default menu system a predefined menu designed for the terminal emulation specified by the Terminal setting. For example, if Terminal is set to 6530, CA Automation Point selects the menu named SESSION_6530 for the default menu.

Menu Control Statements

You can define the menu bar, submenus, and menu options through menu control statements in the user.mnu file. The statement's syntax and description refer to menus, submenus, and items. On the CA Automation Point desktop, they are defined as follows:

- From the CA Automation Point desktop, if you click on a window to select, the menu associated with the window displays on the menu bar at the top of the desktop window. A **menu** is the complete set of submenus and their associated menu options selectable from the menu bar for a given function or session window.
- A unique **menu name** must be assigned to each menu. Each session window or function window definition has a menu defined for it. The name of the menu used for a session or function window is specified in Configuration Manager's, Session Definition Sets dialog for the selected session or function window definition under Window Attributes, menu edit box.
- Each item on the menu bar is a **submenu**. If you click on a submenu, a drop down list of menu options will appear.
- Each menu option is a submenu **item**.

For example, if you select the Merged Msg function window from the CA Automation Point desktop, the menu bar displayed consists of submenus titled, Action, Window, Edit, Cmdarea, and Help. Select the Edit submenu and a drop-down list displays with two items titled, Copy and Paste.

MENU Statement

The MENU keyword assigns a name to the set of submenus designated to appear on the menu bar.

```
MENU=MenuName ,  
SUBMENU= SubmenuName [ ,  
SUBMENU= SubmenuName] [ ,  
...  
SUBMENU= SubmenuName]
```

MenuName

Identifies the unique name assigned to the menu.

SubmenuName

Identifies the internal name of the submenu to be included in the menu bar. *SubmenuName* refers to the corresponding SUBMENU statement(s), specified in the menu definitions file, that define each menu option appearing in the submenu.

Note: Place commas after each line in the statement except the last one. See the following example of the MENU statement.

Example:

The following MENU statement defines the menu for the Merged Msg function window.

```
MENU      = mergedmsg,  
SUBMENU  = action,  
SUBMENU  = window,  
SUBMENU  = edit,  
SUBMENU  = cmdbox,  
SUBMENU  = help
```

SUBMENU Statement

The SUBMENU keyword defines the items to appear in the drop-down list of menu options. Each item can be defined to perform either one of the following tasks:

- Keyboard operation: Standard terminal operations and CA Automation Point desktop operations that can be initiated from the keyboard. For more information, see the appendix "[Customizing Special CA Automation Point Files](#). (see page 457)"
- AXC command processor: CA Automation Point built-in command processors that perform automation-related tasks such as, issuing z/OS commands to an automated session or initiating a REXX program. For more information, see ADDRESS AXC Commands in the *Command and Keyword Reference Guide*.

The item also specifies the descriptive title for the menu option as well as an optional hot key that may also be used to invoke the menu option. Select the menu option to invoke the keystroke operation or AXC command string specified.

```
SUBMENU= SubmenuName, [NAME=(SubmenuTitle[, HelpKey] ) ,]
ITEM=('ItemTitle', KEY(KeyOperation) | XCCMD(AXCCommandString)
[ , ItemComment | TYPE(REM | PM) ] ) [,
ITEM=('ItemTitle', KEY(KeyOperation) | XCCMD(AXCCommandString)
[ , ItemComment | TYPE(REM | PM) ] ) ] [,
...
ITEM=('ItemTitle', KEY(KeyOperation) | XCCMD(AXCCommandString)
[ , ItemComment | TYPE(REM | PM) ] ) ]
```

SubmenuName

Internal name assigned to the submenu.

SubmenuTitle

(Optional) External submenu name or title that is displayed on the menu bar. A *SubmenuTitle* must be specified to define a submenu not previously defined either by CA Automation Point in the axc2p.mnu file or by the user in the user.mnu file.

(Optional) Submenu hot key: The *SubmenuTitle* text can also be used to specify a hot key. To assign a hot key to a submenu, insert a tilde character (~) before any letter in the *SubmenuTitle* text. The hot key must be unique within the menu.

HelpKey

(Optional) This option applies only to versions of CA Automation Point running in a non-Windows environment. CA Automation Point disregards this option when running in a Windows environment.

ItemTitle

Descriptive text for the menu option that displays in the drop-down list of menu options. The text must not exceed 31 characters.

(Optional) Menu option hot key: The *ItemTitle* text can also be used to specify a hot key. A hot key executes a menu option if you press it while viewing the menu containing that option. To assign a hot key to a menu option, insert a tilde character (~) before any letter in the *ItemTitle* text. The hot key must be unique within the submenu.

KEY(*KeyOperation*)

Invokes the key operation specified by *KeyOperation* keyword and the CA Automation Point keyboard parameter file. See "[Customizing Special CA Automation Point Files](#) (see page 457)" for a list of valid key operation keywords and their descriptions.

XCCMD('*AXCCommandString* ')

Issues the AXC command specified by *AXCCommandString*.

Note: &FOCUS_WIN is a special variable that can be referenced in the XCCMD statement that contains the name of the window from which you selected the menu option.

ItemComment

(Optional) Additional text that displays to the right of the *ItemTitle* in the drop-down list of menu options. This text may be used to specify the hot key for the menu option if it exists.

TYPE

(Optional) The TYPE keyword is used to indicate where the menu option is to be displayed. If the TYPE keyword is not specified, then the menu option is displayed in both the CA Automation Point desktop and Remote Viewer windows.

PM

Displays this menu option only in menus that appear on the CA Automation Point desktop.

REM

Displays this menu option only in menus that appear on Remote Viewer windows.

Notes:

- Place commas after each line in the statement except the last one. See the examples of SUBMENU statements.
- To enable the specified hot keys from the CA Automation Point desktop, press the Alt key to display hot key submenu indicators for the selected window. The hot key for the submenu will be underscored in the menu bar. Press any hot key and the associated drop down list of menu options displays. Hot keys for any of the menu options will also be underscored in the drop-down list. Press hot key designated for a menu option to execute.

Example 1:

The following set of SUBMENU statements specifies each submenu for the Merged Msg window.

```

SUBMENU = action, NAME = ( ~Action, help_window ),
  ITEM = ( '~Open remote connection', KEY(rem_conbox), TYPE(REM)),
  ITEM = ( '~Refresh automation rules', XCCMD('REXX "loadrules.rex"') ),
  ITEM = ( 'Save desktop ~layout', KEY(screen_save), TYPE(PM)
),
  ITEM = ( 'New ~host log', KEY(new_hostlog) ),
  ITEM = ( 'New AP ~msg log', KEY(new_xclog) ),
  ITEM = ( '~Dump screen to SCREEN.OUT', KEY(screen_dump) ),
  ITEM = ( '~Shutdown Automation Point', KEY(kill_xc) )

SUBMENU = window, NAME = ( ~Window, help_window ),
  ITEM = ( '~Jump to window', KEY(jump) ),
  ITEM = ( '~Next window', KEY(win_next), Alt+J),
  ITEM = ( '~Increase font', KEY(font_larger), Ctrl+right ),
  ITEM = ( '~Decrease font', KEY(font_smaller), Ctrl+left ),
  ITEM = ( '~Select font', KEY(font_dialog) ),
  ITEM = ( '~Close this window', KEY(quit) )

SUBMENU = edit, NAME = ( Edit, help_3270 ),
  ITEM = ( 'Copy', KEY(copy) ),
  ITEM = ( 'Paste', KEY(paste) )

SUBMENU = cmdbox, NAME = ( ~Cmdarea, help_3270 ),
  ITEM = ( '~Command recall', KEY(up), Up ),
  ITEM = ( 'C~lear command line', KEY(esc), Esc ),
  ITEM = ( '~Next session', KEY(next), Alt+N ),
  ITEM = ( '~Enter command to session', KEY(cmd_host) ),
  ITEM = ( 'Sta~rt REXX program or script', KEY(execute) ),
  ITEM = ( 'St~op REXX program', KEY(cancel_rexx) ),
  ITEM = ( '~Pause/restart automation', KEY(pause) )

SUBMENU = help, NAME = ( ~Help, help_help ),
  ITEM = ( 'A~P Help', KEY(xc_msg_help) ),
  ITEM = ( '~About Automation Point', KEY(about) )

```

Example 2:

The following SUBMENU statement specifies a menu option that enters a command to a session. This example uses the slash (/) character as an alternate delimiter in the AXC command string because the command string contains an embedded single quote.

```

SUBMENU = AS4aid, NAME = ( ~Aid, help_3270 ),
  ITEM = ( 'Display status', XCCMD('SESSCMD /dspstat ALL/ Session(as400rul)') )

```

Example 3:

The following SUBMENU statement specifies a menu option that will start a REXX program. This example uses the double quote (") character as an alternate delimiter in the AXC command because the command string contains an embedded single quote.

```
SUBMENU = action_opsmg, NAME = ( ~Action, help_window ),
ITEM = ( '~Refresh automation rules', XCCMD('REXX "loadrules.rex"') )
```

Example 4:

The following SUBMENU statement specifies hot keys for the submenu, 'Window', and its menu options.

```
SUBMENU = window, NAME = ( ~Window, help_window ),
ITEM = ( '~Jump to window', KEY(jump) ),
ITEM = ( '~Next window', KEY(win_next), Alt+J),
ITEM = ( '~Increase font', KEY(font_larger), Ctrl+right ),
ITEM = ( '~Decrease font', KEY(font_smaller), Ctrl+left ),
ITEM = ( '~Select font', KEY(font_dialog) ),
ITEM = ( '~Close this window', KEY(quit) )
```

Hot key specified for the 'Window' submenu is W. Hot key specified for the 'Next window' function is N. Press the Alt key to enable the hot keys. Then press W to display the menu options for the 'Window' submenu. Press N to select the 'Next window' menu option.

Adding New Menu Options to Existing Submenus

When you select a submenu from the menu bar in a CA Automation Point function or session window, a drop down list of menu options appear. If you wish to add a menu option to an existing submenu, you'll first need to reference existing names for menus and submenus before adding customized menu options. Follow these steps to customize.

1. Identify the *MenuName* specified for the function or session window you wish to customize. To determine the name of the menu used for a desktop window, go to Configuration Manager's, Session Definition Sets dialog and open the selected session or function window definition. See Window Attributes, menu edit box.

Note: If the menu setting is set to default, the actual name of the menu will follow these naming conventions:

- Default menus for function windows will have the same name as the function window type setting. For example, the Merged Msg window has a function window type of MERGEDMSG. Therefore, the default menu name for the Merged Msg window is MERGEDMSG.

- The default menu name for 3270 session windows with default terminal emulation specified is 'session_3278'.
 - The default menu name for asynchronous session windows with default terminal emulation specified is 'session_asynch'.
 - The default menu name for session windows with a non-default terminal emulation specified is 'session_' followed by the terminal emulation specified for the session.
 - The default menu name for a session window with terminal emulation of 3278 models 2-5 is 'session_3278'. Similarly, the menu name for a session window with terminal emulation of 3279 models 2-5 is 'session_3279'. For example, if the terminal setting for a 3270 session is 3279_4, then the name of the default menu associated with this session is 'session_3279'.
2. Identify the *SubmenuName* for the submenu to which you wish to add a menu option. Search axc2p.mnu and user.mnu files for all MENU statements whose name is *MenuName* as determined in step 1. This should return submenu names for each item on the menu bar. For each submenu name on the MENU statements, search axc2p.mnu and user.mnu files for the SUBMENU statement that defines the external submenu name or submenu title for the submenu you wish to customize.
 3. Add a SUBMENU statement to the user.mnu file. Specify the SUBMENU statement with your customized ITEM settings using the *SubmenuName* and *SubmenuTitle* identified in step 2. Edit user.mnu file to include the new SUBMENU statement and save.

Note: Submenus may be defined in more than one menu. If you add a menu option to an existing submenu, the menu option will appear in every desktop window that uses a menu defined with that submenu name. If this is not the desired result, you may wish to create a new submenu with the customized menu options and include that submenu on menus of your choice. For more information, see the section [Adding New Submenus to Existing Menus](#) (see page 92).

Example: The following SUBMENU statement included in the user.mnu file will add three more menu options to the Cmdarea submenu in the Merged Msg window.

```
SUBMENU = cmdbox,
ITEM = ( '~Fast background updates', XCCMD('SESSCNTL BWUPDATE(1)'),
ITEM = ( 'Normal ~background updates', XCCMD('SESSCNTL BWUPDATE(5)'),
ITEM = ( '~Slow background updates', XCCMD('SESSCNTL BWUPDATE(9)')
```

Adding New Submenus to Existing Menus

When you select a session or function window on the CA Automation Point desktop, the menu bar for the selected window displays at the top of the desktop window. If you wish to add a submenu to the menu bar of an existing menu, you'll first need to reference the name of the menu specified for the desktop window before creating a new submenu for it. The new submenu will appear only on those windows that use that menu. Follow these steps to customize the menu:

1. Identify the *MenuName* specified for the function or session window you wish to customize. To determine the name of the menu used for a desktop window, go to Configuration Manager's, Session Definition Sets dialog and open the selected session or function window definition. See Window Attributes, menu edit box.

Note: If the menu setting is set to default, the actual name of the menu will follow these naming conventions:

- Default menus for function windows will have the same name as the function window type setting. For example, the Merged Msg window has a function window type of MERGEDMSG. Therefore, the default menu name for the Merged Msg window is MERGEDMSG.
- The default menu name for 3270 session windows with default terminal emulation specified is 'session_3278'.
- The default menu name for asynchronous session windows with default terminal emulation specified is 'session_asynch'.
- The default menu name for session windows with a non-default terminal emulation specified is 'session_' followed by the terminal emulation specified for the session.
- The default menu name for a session window with terminal emulation of 3278 models 2-5 is 'session_3278'. Similarly, the menu name for a session window with terminal emulation of 3279 models 2-5 is 'session_3279'. For example, if the terminal setting for a 3270 session is 3279_4, then the name of the default menu associated with this session is 'session_3279'.

2. Add a SUBMENU statement to the user.mnu file. Specify the SUBMENU statement with your customized ITEM settings using a new *SubmenuName*. Edit user.mnu file to include the new SUBMENU statement and save.
3. Add the following MENU statement to user.mnu file using *MenuName* determined in step 1 and *SubmenuName* determined in step 2.

```
MENU = MenuName , SUBMENU = SubmenuName
```

4. Edit user.mnu file to include the new MENU statement and save.

Example:

Add the following MENU and SUBMENU statements to the user.mnu file to add a new submenu titled 'Refresh' to an existing menu defined for the Merged Msg window by CA Automation Point in the axc2p.mnu file. The default menu for the Merged Msg window is mergedmsg.

```
MENU = mergedmsg,
  SUBMENU = b_refresh
SUBMENU = b_refresh, NAME=(~Refresh),
  ITEM = ( '~Fast background updates', XCCMD('SESSCNTL BWUPDATE(1)'),
  ITEM = ( 'Normal ~background updates', XCCMD('SESSCNTL BWUPDATE(5)'),
  ITEM = ( '~Slow background updates', XCCMD('SESSCNTL BWUPDATE(9)')
```

Defining a New Menu

Follow these steps to create a new menu.

1. Add a SUBMENU statement for each new submenu you wish to include in the menu bar. Specify the SUBMENU statement with your customized ITEM settings using a new *SubmenuName*. Edit user.mnu file to include the new SUBMENU statements and save.
2. Identify *SubmenuName* for each existing submenu you wish to include in the menu bar. It is highly recommended that you include the existing submenu named, window, to your new menu. To determine the name of a predefined submenu given its submenu title, search axc2p.mnu and user.mnu files for all SUBMENU statements with the desired *SubmenuTitle* specified. The first operand in the SUBMENU statement is the submenu name.
3. Add a new MENU statement with a new menu name and specify all the submenu names you want to include in the menu bar. Edit user.mnu file to include the new MENU statements and save.
4. Assign a new menu to the function or session window. Use Configuration Manager's, Session Definition Sets dialog to open the selected function window or session definition dialog. Select the desired menu from the menu edit box and exit the Session Definition Sets dialog to save.
5. CA Automation Point must be recycled before the new menu displays on the desktop window.

Example:

When you create a new function window of type PLOT, you may need to create a new menu for it. The following MENU and SUBMENU statements added to the user.mnu file define a new menu with two submenus named, plot_control and window. Submenu plot_control is user defined and submenu window is defined in the axc2p.mnu file. A new plot type function window is configured to use a menu named 'plot'. The existing submenu named, window, provides common CA Automation Point desktop navigational operations.

```
MENU = plot,  
      SUBMENU = window,  
      SUBMENU = plot_control  
  
SUBMENU = plot_control, NAME = (~control),  
          ITEM = ('start plot', XCCMD( 'REXX "plot start"' ) ) ,  
          ITEM = ('stop plot', XCCMD( 'REXX "plot stop"' ) )
```

Customizing Screen Fonts

This section describes how you can customize the CA Automation Point screen fonts. It covers the following topics:

- Understanding screen fonts
- Changing the font list on the local machine
- Changing the font list for the remote workstation

Understanding Screen Fonts

If you choose, you can change the font of each window that you view with CA Automation Point. To do this, from the window, choose Window, and then select Increase font or Decrease font. You can also press Ctrl+right to make the font larger or Ctrl+left to make the font smaller.

When you open a window to its normal size, CA Automation Point uses the next-to-last screen font in your font list. When you maximize a window, CA Automation Point uses the last font in the list. Therefore, the last font should be large enough so that an 80-column, 25-row window fills the screen.

Note: Keep in mind that the fewer number of fonts in your font list, the fewer number of keystrokes or Next font menu selections you need to reduce or enlarge a window.

When you change the font size, CA Automation Point automatically resizes the window to be proportionate with the new font size.

Default Fonts

CA Automation Point lists its default fonts in the AUTOMATE.fnt file. This file is located in the *installdir*\Distrib directory. The default font list is optimized for monitors with a resolution of 640x480.

Note: If you have a monitor with a resolution other than 640x480, change the list to include fonts that are best suited to the resolution of your monitor.

Deciding Which Fonts to Include

You can experiment with the screen fonts to decide which ones to include in your font selection list.

For instructions on experimenting with different fonts for local and remote workstations, see the sections [Experimenting with Different Screen Fonts for Font Configuration](#) (see page 96) and [Experimenting with Different APVIEW Screen Fonts](#) (see page 98), respectively.

Changing the Font List on the Local Machine

You can change the font selection list on the machine that is running CA Automation Point locally. See the section [Changing the Font List for the Remote Workstation](#) (see page 97) in this chapter for information on changing the fonts on a workstation running CA Automation Point remotely.

If CA Automation Point and the Remote Viewer are running on the same workstation, a common font list is referenced. Changes to the font list using the font configuration program will affect the available fonts for the Remote Viewer the next time it is started.

Font Configuration Program

To change the font selection list, use the CA Automation Point font configuration program.

Note: The font configuration program and CA Automation Point cannot run at the same time. Before you run the font configuration program, verify that CA Automation Point is not running.

To change the font list on the local machine, follow this procedure:

1. Start Configuration Manager.
2. Choose Expert Interface, Automation, Automation Point Desktop Settings.

3. Click the Customize Fonts to start the font configuration program. The Select Fonts dialog displays.

A list of the available fonts that are installed on your workstation is displayed. The selection list includes only non-proportional (monospaced) fonts large enough to be readable, yet small enough to allow 80 columns of text to display. The fonts listed appear in order, from the smallest to the largest.

4. Highlight each font you want to use.

Keep in mind that the fewer number of fonts in your font list, the fewer number of keystrokes or Next font menu selections you need to reduce or enlarge a window.

5. Click OK.

6. Choose Action, Shutdown AP Utilities to stop the program and save your font selections.

CA Automation Point stores your font selections in the AUTOMATE.fnt file, which is used by both CA Automation Point and the Remote Viewer running on the workstation.

7. Restart CA Automation Point to put your updated font list into effect.

Experimenting with Different Screen Fonts for Font Configuration

To experiment with different screen fonts to include in your font selection list

1. Start the font configuration program as described in the section Font Configuration Program.
2. In the Select Fonts dialog, select *all* the fonts that the font configuration program offers.
3. Restart CA Automation Point.

4. Switch to any window (other than a Plot window) and select fonts one at a time to see how they appear.

If a font is too small to read or if it produces a window size too similar to that produced by another font, omit that font from your font list. Note the names and sizes of the fonts you want to use.

5. Start the font configuration program again and reselect all the fonts that you chose in Step 4.

Changing the Font List for the Remote Workstation

This section describes the procedure for changing the font selection list for a workstation that will run the CA Automation Point Remote Viewer.

If CA Automation Point and the Remote Viewer are running on the same workstation, a common font list is referenced. Changes to the font list using APVIEW /FONTS will affect available fonts for CA Automation Point the next time it is started.

APVIEW /FONTS Utility Program

To change the font selection list, use the CA Automation Point APVIEW/FONTS utility program

Note: The APVIEW /FONTS utilities program and the Remote Viewer (APVIEW.exe) cannot run at the same time. Before you run the APVIEW /FONTS program, verify that the Remote Viewer is not running.

To change the font list for the remote workstation, follow this procedure:

1. Issue this command from your workstation operating system prompt:

```
APVIEW /FONTS
```

The program displays a list of the available fonts that are installed on your workstation. The selection list includes only non-proportional (mono-spaced) fonts large enough to be readable, yet small enough to allow 80 columns of text to display. The fonts listed appear in order, from the smallest to the largest.

2. Highlight each font you want to use.

Note: Keep in mind that the fewer number of fonts in your font list, the fewer number of keystrokes or Next font menu selections you need to reduce or enlarge a window.

3. Click OK.
4. Choose Exit Remote Viewer from the APVIEW /FONTS system menu to stop the program and save your font selections.

CA Automation Point stores your font selections in the AUTOMATE.fnt file.

5. Restart the Remote Viewer to put your updated font list into effect.

Experimenting with Different APVIEW Screen Fonts

To experiment with different screen fonts to include in your font selection list for APVIEW

1. Follow the procedure in the section APVIEW /FONTS Utility Program to display a list of the available fonts that are installed on your workstation.
2. Select *all* the fonts that the APVIEW /FONTS program offers.
3. Restart CA Automation Point.
4. Switch to any window (other than a Plot window) and select fonts one at a time to see how they appear.

If a font is too small to read or if it produces a window size too similar to that produced by another font, omit that font from your font list. Note the names and sizes of the fonts you want to use.

5. Start the APVIEW /FONTS program again and reselect all the fonts that you chose in step 4.

Note: If your workstation does not already have suitable fonts installed, you may install third-party font packages. CA Automation Point will use only those fonts that are non-proportional (mono-spaced) and those that fall within a size range that would be viewable on the Automation Point Desktop. Third-party fonts must be of type Bitmap, which means the font file contains bitmap pictures of each character at a supported size.

If you plan to use third-party fonts with Remote Viewer, you must also install these fonts on all Remote Viewer client workstations. If you are unable to see these qualifying third-party fonts in the appropriate CA Automation Point font utility program, you can either rename or remove the automate.fnt file, then re-execute the font utility program.

Issuing Commands

This section contains information about issuing various types of commands.

Command Dialog

Use the Command Dialog as a quick way to enter console commands to a session.

To access the Command Dialog

1. Select the session window.
2. Choose Command Dialog from the menu, or press Alt+X.
3. Enter the command and press OK, or press Submit for sending multiple commands to the same session.

Command Area

Several CA Automation Point function windows display the *command area* in the lower part of the window. From the command area, you can:

- Issue console commands to sessions that CA Automation Point manages
- Start a REXX program or a CA Automation Point script

The Command Area State

The command area prompt indicates the current **state** of the command area and determines what types of commands you can enter. There are two possible states for the command area, so the prompt can be either of the following:

- AXCO100A: Enter command for session
- AXCO103A: Enter REXX program or script to execute in session

The current state depends on what type of command operation you chose last from the Cmdarea menu, or the last command that was issued.

Changing the State of the Command Area

To change the current state of the command area for *another* type of command operation, select another option from the Cmdarea menu, or do the following:

1. Press the up arrow and down arrow keys to scroll through previously entered commands.

As each command displays, the state of the command area (associated with each recalled command) changes.

2. When you have found a command and command area state that you want, press Esc to erase the command text (or REXX program or script file name) from the command area.
3. After you erase the command text, the current state of the command area (associated with the erased command) remains unchanged. You can then enter a similar command.

Issuing a Console Command

To send a console command to a specific session

1. Open a window that contains a command area.
2. Press the Next Session key (the default key is Alt+N) or click the Next session option from the Cmdarea menu until the command area displays the name of your target session.
3. Select the Cmdarea menu.
4. Choose Enter command to session.
5. Enter the console command in the command area.

Note: You can send only displayable characters. You cannot send key functions such as a PF key, Reset, or Clear.

Executing REXX Programs or Scripts

To start a REXX program or a script in a session managed by CA Automation Point

1. Open a window that contains a command area.
2. Press the Next Session key (the default key is Alt+N) or click the Next session option from the Cmdarea menu until the command area displays the name of your target session.
3. Display the Cmdarea menu.
4. Choose Start REXX program or script.
5. Enter the name of the REXX program or script in the command area.

Important! CA Automation Point uses the three-character file name extension to determine whether a REXX program or a script should execute. Follow this convention when naming your REXX programs and scripts:

- Give the extension .cmd or .rex to all REXX programs.
- Give the extension .scr to all scripts.

When you enter the REXX program or script file name in the command area, supplying the extension is optional. If you specify a file name *without* an extension, CA Automation Point searches first for a REXX program, and then for a script with that file name.

Stopping a REXX Program

To end the execution of a REXX program

1. Choose Stop REXX program from the Cmdarea menu.
A dialog displays that lists all of the queued REXX programs.
2. Highlight the REXX program that you want to terminate.
3. Select Stop to stop the REXX program or Cancel if you change your mind.

Recalling Previously Entered Commands

The command area allows you to recall commands that you had entered previously. The recall feature recalls both the text of the entry and the command area state.

To find a previous command, do either of the following:

- Choose Command recall from the Cmdarea menu.
- Move the cursor into the command area and press the up arrow and down arrow keys to scroll through previous commands.

When you have found the command that you want, you can do any of the following:

- Press Enter to issue the command again.
- Edit the command, and then press Enter to issue it.
- Press Esc to erase the command text (or REXX program or script file name) from the command area.

You can then enter another similar command.

Displaying Recalled Messages

This section describes how to view recalled messages.

Windows for Viewing Messages

CA Automation Point provides the following windows for viewing messages:

- The Action Message Recall window

This window displays the last 500 lines of highlighted (action) message text. These messages, highlighted by a CA Automation Point rule, by MCS, or by VSE, are the same messages that appear in the action messages area of the Merged Msg window. Even if a message has scrolled off the Merged Msg window, you can view it in the Action Msg Recall window.

Highlighted messages usually report system errors and other conditions requiring a response from either CA Automation Point or an operator. The Action Msg Recall window displays these messages in the order in which CA Automation Point processed them.

- The Normal Msg Recall window

This window displays the last 500 lines of non-highlighted (normal) message text. Normal messages also appear in the main message area of the Merged Msg window. Even if a message has scrolled off the Merged Msg window, you can view it in the Normal Msg Recall window.

- The AP Msg Recall window

This window displays, in chronological order, the last 500 lines of message text generated by the currently active CA Automation Point.

CA Automation Point periodically updates the text displayed in its message recall windows; these windows have the following characteristics:

- You can scroll the displayed message list forward, backward, left, or right as you view it.
- You can use a highlighted bar to indicate a specific line in the message list.
- You can delete the message text in the highlighted bar.

Marking a Place in the List

You can designate a line on a message recall window to mark a particular place in the message list. The place marker line highlights message text in reverse video. When you first display a message recall window, the first line on the screen is the place marker line.

To select another screen line as the place marker, either click it or move to it using the up arrow and down arrow keys. After you position the place marker, it remains on the line you selected.

Deleting a Recalled Message

If a message is not important, you can delete it by selecting the line displaying that message as the highlighted place marker line, and then pressing the Delete key. You can also choose Delete highlighted line from the menu.

Browsing Recalled Messages

When viewing recalled messages, keep this information in mind:

- When a console receives an action message, CA Automation Point rules process that message. If a rule suppresses or lowlights the action message, CA Automation Point does not display it in the action message area of the Merged Msg window or in the Action Msg Recall window.
- If a rule highlights a normal message, CA Automation Point displays that message in the Merged Msg and Action Msg Recall windows.
- When an outstanding action message is resolved and the message is still displayed in the terminal emulator window, CA Automation Point automatically removes that message from the action message area in the Merged Msg window and from the Action Msg Recall window.
- Deleting a message from the Action Msg Recall window does *not* delete it from the console. The message remains on the console until it scrolls off.

Managing a Recalled Message List

CA Automation Point message recall windows include a Listmgt menu that lets you choose from a series of list management tasks. See the Automation Point Desktop online help for descriptions of the Listmgt menu options.

Displaying Recent CA Automation Point Messages

To see the last 20 messages that CA Automation Point issued, display the AP Msg window. The window also has a command area from which you can issue commands to sessions that CA Automation Point manages.

If you do not want to issue commands to sessions, or if you want to see the last 500 lines of message text issued by CA Automation Point, display the AP Msg Recall window instead of the AP Msg window.

Merging Messages from Managed Sessions

The CA Automation Point terminal emulator sessions allow you to operate multiple processor and software console sessions from one workstation. The Merged Messages (Merged Msg) window lets you use a single window to view both action and normal (non-highlighted) messages from all of those sessions.

The Merged Msg window displays only the most recent action and normal messages. You can view more of these messages by displaying the Action Msg Recall and Normal Msg Recall windows as described in the section Displaying Recalled Messages.

The following is a sample Merged Msg window. In this sample, the action message area displays at the top of the window. Below the action message area is the normal message area, which fills most of the window. The command area displays at the bottom of the window.

```

Merged Msg
3020 16143 IEA994E ALL SY21.DUMP DATA SETS ARE FULL
3020 16144 IEA405E WTO BUFFER SHORTAGE - 001 FULL
3000 16145 IEA994E ALL SY21.DUMP DATA SETS ARE FULL
3000 16145 IEA404A SEVERE WTO BUFFER SHORTAGE - 1001 FULL
3000 16150 IEA406I WTO BUFFER SHORTAGE RELIEVED
3000 16150 IEA122I UNRECOVERABLE OR FORCED ERROR ON NODE L12006 - VARY
      SMCT
VAX 16:15 REQUEST 2321, FROM USER OPERATOR ON VAXA PLEASE MOUNT VOLUME
      07AP15 IN DEVICE $1$MUAL: (HSC001)
3020 16151 IEA404A SEVERE WTO BUFFER SHORTAGE - 1001 FULL
3000 16152 IEA122I UNRECOVERABLE OR FORCED ERROR ON NODE L12006 - VARY
      SMCT
3020 16152 IEA406I WTO BUFFER SHORTAGE RELIEVED
3000 16:15 IMC220W MVS IS UNABLE TO CONTINUE: WAIT STATE CODE 0A2
CHAPAL3 16:16 Failure condition raised by Session VOM
TANDEM 16:16 TACL TERMINAL I/O ERROR 066, $ATP0.#TERM23
TINGMsg 16:16 Application Popup_26_I: Application popup: Messenger Service
      : Message from CHAPAL3 to CHAPAL3 on 7/24/2000 4:16:09 PM
      NM CALLING - ITEM 0207 - System 3006 is down with wait state
      code 0a2
TANDEM 16:16 TACL TERMINAL I/O ERROR 066, $ATP0.#TERM23
AMC0100A: Enter command for session 3020
  
```

The Merged Msg window contains these areas:

Title line

The top line of the screen displays the name of the current window

Menu bar

When the Merged Msg window is selected, the CA Automation Point desktop menu bar displays its associated Window, Edit, Cmdarea, and Help options

Action message area

The top part of the screen contains the action message area, which displays the highlighted messages that CA Automation Point receives from terminal emulator sessions. You can specify the size of this area in the Action Msg Area field of your Automation Point Desktop Settings.

The action message area can display up to 19 of the most recent highlighted messages. To see the last 500 lines of highlighted message text, display the Action Msg Recall window.

Main message area

The center of the screen contains the main message area, which displays messages from the sessions managed by CA Automation Point. If CA Automation Point manages more than one session, it merges messages from those sessions. The main message area displays up to 20 of the most recent normal messages. To see the last 500 lines of normal message text, display the Normal Msg Recall window.

Command area

The bottom of the screen contains the command area, from which you can issue commands to the session indicated in the AXC0100A message. The AP Msg and Command windows also have a command area.

Note: The number of messages that can be displayed for the entire Merged Msg window can be configured through Configuration Manager. To do so, choose Expert Interface, Automation, Automation Point Desktop Settings.

Displaying Log File Contents

Two CA Automation Point function windows let you view the contents of log files:

- The AP Log window shows you the contents of the AP message log file, which stores CA Automation Point messages.
- The Host Log window displays the contents of the host message log file, which stores host messages.

Scrolling Through Log Contents

To scroll forward and backward through the message text in the windows, press Page Down and Page Up, arrow keys, or operate the windows' scroll bar with your mouse.

Both windows also offer a Logmgt menu, giving you more scrolling options. See the Automation Point Desktop online help for descriptions of the Logmgt menu options.

Displaying Notification Server and Notification Manager Messages

The AP Notification Messages window displays the last 500 messages generated by the notification server, notification (NMFIND) request, or Answer Tree. The AP Notification Messages window contains two menus that enable you to obtain detailed information about the notification server:

- Diagnostics
- Trace

Displaying Diagnostic Information

The Diagnostics menu enables you to obtain diagnostics information about the notification server. See the Automation Point Desktop online help for descriptions of the Diagnostics menu options.

Displaying Trace Information

The Trace menu enables you to obtain trace information about the notification server. See the Automation Point Desktop online help for descriptions of the Trace menu options.

Note: Trace commands turn tracing on if it is currently off, and turn tracing off if it is currently on. To display the current status of all the trace types, use the Trace status command.

Getting Information About CA Automation Point Messages

For information about the messages generated by the CA Automation Point product, see the *Message Reference Guide*. This guide contains such information as the message ID, message text, detailed information about the message, and the action to take, if any, in response to the message.

Displaying Graphs of System Information

The Plot window displays graphs that contain information about your system. The CA Automation Point Plot feature generates the graphs. For more information, see the chapter "[Using the Plot Feature](#) (see page 441)."

Stopping CA Automation Point

Before it ends, CA Automation Point stops automation in all automated sessions and closes all windows.

To stop CA Automation Point

1. Choose Shutdown Automation Point from the Action menu.
2. When the Shutdown Automation Point dialog displays, select Shutdown at the warning prompt to quit, or select Cancel to return to CA Automation Point.

Chapter 6: Viewing Remote Sessions

This chapter explains how to use the Remote Viewer and Web Message Viewer (Web MV).

Overview

CA Automation Point provides you with the following applications for configuring and viewing managed sessions from a remote workstation:

Remote Viewer

Extends the CA Automation Point real-time console emulation to a remote Windows workstation.

Web Message Viewer (Web MV)

Allows you to remotely access and view a recent history of messages received and processed by CA Automation Point using a Web browser.

The Web Message Viewer provides you with a line-by-line interpretation of the messages received by CA Automation Point, *not* exact console emulation. However, the Web Message Viewer allows you to change the visual attributes of the messages and view messages that have scrolled off the screen.

Although the Remote Viewer and Web Message Viewer are both remote viewing applications, they perform different functions, as described in the following sections.

Remote Viewer

The Remote Viewer provides you with an exact emulation of the CA Automation Point messages. It allows you to access and control sessions on workstations that are running CA Automation Point at remote sites. The Remote Viewer allows one or more users to simultaneously connect to sessions from any number of CA Automation Point workstations.

Notes:

- CA Automation Point provides an HTML report that shows you the status of all Remote Viewer connections to the CA Automation Point server. The Remote Connections Report gives you an overview of all Remote Viewer users who are currently connected to the CA Automation Point server. It also shows what sessions or function windows these users are connected to, and when they were first connected. The report is automatically generated and dynamically updated whenever a Remote Viewer user connects or disconnects to the CA Automation Point server.
- The Remote Connections Report is provided on the CA Automation Point server where Remote Manager is running. You can display the report using Configuration Manager by specifying Expert Interface, Automation, Remote Viewing, and clicking View Remote Connection Report. If you keep the report open in your browser for a long period of time, be sure to refresh the browser's display to get the most recent content of the report.

Remote Access

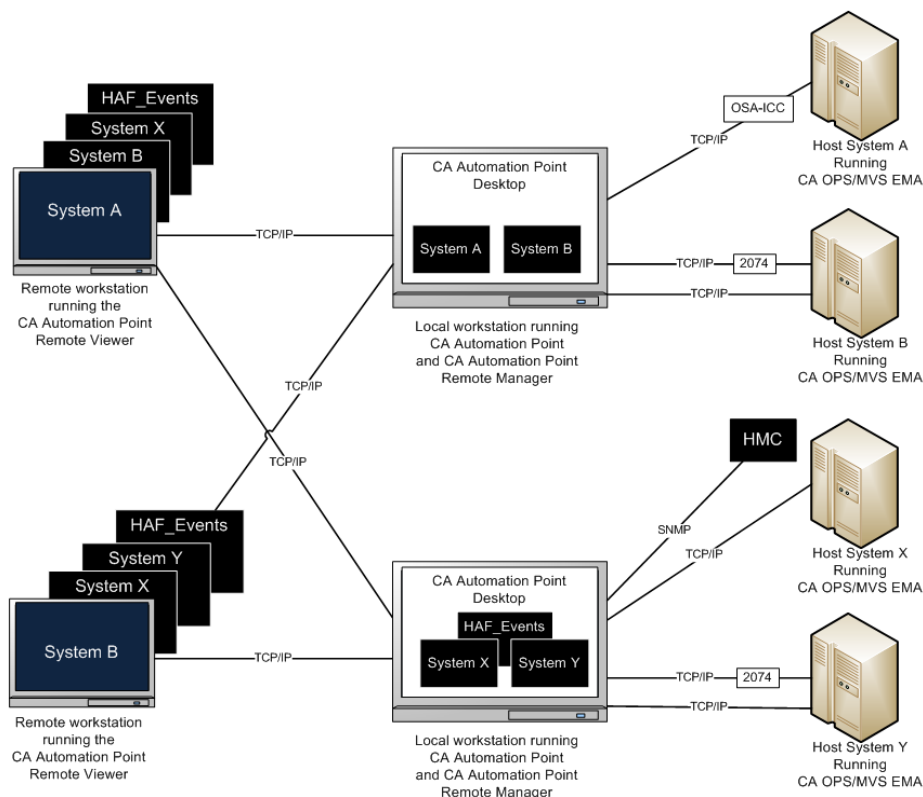
The Remote Viewer allows you to access and control sessions on workstations that are running CA Automation Point at remote sites. Many of the sessions that CA Automation Point manages are system consoles for the monitored platform (for example, master consoles for z/OS systems). Therefore, Remote Viewer is primarily a tool for those operators and systems programmers who need to access the system consoles, address exceptions and alerts, or perform IPL-related functions.

The Remote Viewer allows one or more users to simultaneously connect to multiple sessions from multiple CA Automation Point workstations. Through a TCP/IP connection, the Remote Viewer provides the capability to perform the following:

- Connect a single Remote Viewer to one or more sessions appearing on the host CA Automation Point desktop. Each CA Automation Point session displays on the Remote Viewer workstation as a separate window.
- Connect a single Remote Viewer to sessions across multiple CA Automation Point workstations. You can remotely connect to any number of CA Automation Point workstations using Remote Viewer.
- Multiple Remote Viewer sessions can connect to the same CA Automation Point workstation. These remote views can be opened by one or more Remote Viewers running on different remote hosts. They can also connect to the same session.

You can open up to 200 remote views to one CA Automation Point workstation from one or more Remote Viewers. Remote Viewer can open up to 200 remote views of sessions on one or more CA Automation Point workstations.

The following diagram illustrates the many capabilities of the Remote Viewer. In this illustration, two Remote Viewers are connected to two CA Automation Point hosts. Each remote workstation has a TCP/IP connection to the two CA Automation Point hosts. Notice that the sessions for System B, System X, and PCON B are being viewed by both remote workstations.



CA Automation Point is accessible from multiple Remote Viewers as long as there is a TCP/IP connection and CA AP Remote Manager is running. CA AP Remote Manager handles concurrent keyboard entries across multiple Remote Viewers and intercepts session screen displays on CA Automation Point to display them on all connected Remote Viewers.

Web Message Viewer

The Web Message Viewer provides you with a line-by-line interpretation of the messages received by CA Automation Point, *not* exact console emulation. Therefore, you can change the visual attributes of the messages and view messages that have scrolled off the screen. Web MV allows you to not only choose the messages that you want to view, but to break the messages down into columns of information based on your needs.

Remote Viewer

The ability to access and control system automation across multiple sites can be performed efficiently using the Remote Viewer application of CA Automation Point. The Remote Viewer provides access to sessions that are managed by CA Automation Point from a remote workstation.

Enabling CA Automation Point for Remote Viewing

CA Automation Point, through CA AP Remote Manager, allows other computers to view and interact with CA Automation Point remotely. On the workstation that is running CA Automation Point, CA AP Remote Manager manages the connections to the remote workstations that are running the Remote Viewer. For more information, see the *CA Automation Point Installation Guide* for instructions on installing the Remote Viewer.

To properly enable CA Automation Point for remote viewing, you first need to verify that TCP/IP is working by performing the following steps:

1. Try to ping the workstation from a command prompt by entering the following command:

```
echo %computername%
```

The name of your computer is returned.

2. Enter the following command:

```
ping name
```

where name is the name returned in the echo command. Ping returns an echo of "Reply from ip-address."

If there is no reply, you should install TCP/IP from the Windows installation process before proceeding. TCP/IP must be installed and configured on the computer for the Remote Access feature of CA Automation Point to work.

3. Press Ctrl+Break to exit ping.

Accessing the Remote Viewing Dialog

When the TCP/IP connection has been verified, you are ready to enable server-side support for the Remote Viewer.

To set up remote viewing

1. Start Configuration Manager.
2. Choose Expert Interface, Automation, Remote Viewing. The Remote Viewing dialog displays.
3. Click the Enable Remote Viewing check box.
4. If you intend to use Remote Viewer to remotely start CA Automation Point, specify a domain name, user account name, and password. The user Account is the account under which CA Automation Point runs when it is started from the Remote Viewer. The user account is used by both the Remote Viewer and the automatic startup feature (if enabled).
5. Optionally, enter the IP names or IP addresses for which you want to enable access to CA Automation Point.

If you specify IP names or IP addresses in the Trusted Remote Host Name list, CA AP Remote Manager server will accept connections from only those workstations that you specify in the list. If you do not specify anything in the Trusted Remote Host Names list, CA AP Remote Manager server will accept connections from any IP connection.

IP Names

IP names in the Trusted Remote Host Names list are compared to the resolved IP name of the incoming connection.

You can specify IP name masks to allow pattern matching. For example, an IP name mask of APLAB* matches any IP host name beginning with the characters APLAB. An IP name mask of *.mylan.com matches any incoming host name within the mylan domain.

IP Addresses

You can enter IPv4 or IPv6 addresses in the Trusted Remote Host Names list, keeping the following in mind:

- You can explicitly specify IPv4 addresses.
Explicitly specified addresses match only the exactly matching IP address.
For example, 172.24.51.6 allows access to connections only from IP address 172.24.51.6.
- You can use an IPv4 IP address containing one or more 0s to match a range of addresses.
For example, 172.24.51.0 matches 172.24.51.1 to 172.24.51.255. A mask value of 0.0.0.0 matches any address; therefore, access is not restricted by host name or IP address.

- You can explicitly specify an IPv6 address.

Explicitly specified addresses match only the exactly matching IPv6 address.

Note: You must enclose IPv6 addresses in square brackets ([]).

- For example, the mask value [2001:0db8:85a3:08d3:1:8a2e:370:7344] matches (and thereby allows access) to an incoming connection only from the IPv6 address 2001:0db8:85a3:08d3:1:8a2e:370:7344

You can explicitly specify IPv6 subnet mask to allow wildcard matching. IPv6 IP addresses containing a subnet mask are treated as wildcards matching any incoming connection within the subnet range.

For example, the IPv6 address [2001:0db8:85a3:08d3:1:8a2e:370:7344]/32, matches any address in the range 2001:0db8:0000:0000:0000:0000:0000:0000 to 2001:0db8:ffff:ffff:ffff:ffff:ffff:ffff.

6. By default, the CA Automation Point server and Remote Viewer communicate using the TCP/IP port number 5500. You may change it if it is already in use or you want it to be unknown. If you use a value other than 5500, you have to specify this value when you connect to CA Automation Point through the Remote Viewer.

7. Choose from the following levels of Remote User Login Security:

No User Security

Specifies that the user login (ID and password) is not verified. Session-level security is determined by the value set in the Permission Level field for your session.

Windows Security

Specifies that you want to use the Windows security system to enforce user access privileges. The user login is verified on the host machine or domain, if specified. Session-level security is determined by the value set in the Permission Level field for your session. Session-level security by user is optionally determined by mapping session permissions to file permissions.

Securing the Remote Viewer

The best strategy for preventing local access abuses is to limit physical access to the CA Automation Point workstation. You can run both the CA-AP Autostart Manager and CA AP Remote Manager services without accessing the desktop, requiring local users to use the Remote Viewer directly on the CA Automation Point workstation. To disable access to the desktop, open Configuration Manager and choose Expert Interface, Automation, Remote Viewing. On the Remote Viewing dialog, uncheck the Show AP Desktop checkbox.

Note: The Show AP Desktop check box is available only for Windows Server 2003 and earlier.

Security Capabilities

Before you use the Remote Viewer, review the security capabilities that CA Automation Point and the Windows operating system provide.

CA Automation Point provides the following security capabilities, which allow you to set up a secure environment for using the Remote Viewing Dialog:

- You can list the IP names and IP addresses that can access CA Automation Point using the Trusted Remote Host Names edit box. Doing so prevents all other workstations from accessing the CA Automation Point server.

You can choose the TCP/IP port number (the supplied default or one of your own) that the remote workstation uses when accessing CA Automation Point. Using a port number ensures that all TCP/IP communication to the CA Automation Point server from the Remote Viewer is done from within your sites firewall.

- To protect the transmission of data between CA Automation Point and the Remote Viewer workstation, encryption is provided for the logon (user ID and password), command, keyboard, and display packets. This function eliminates the sniffing of packets.
- You can set up a login ID and password for the Remote Viewer to use when connecting to CA Automation Point. The logon ID and password can be the same as those used for Windows (defined for your CA Automation Point server), or can be defined by Windows at the domain level. If the logon ID and password are defined at the domain level, Remote Viewers can connect to multiple CA Automation Point sessions with the same logon ID and password.
- You can configure the Single Sign-on feature which allows remote connection to an Automation Point server without the need to retype login credentials.
- You can specify session-level access permissions by using a combination of values set in your session definition set and User and Permission GUI facilities that are part of the Windows system.

Note: For information about session-level access permissions, see the section [Global and Local Session Settings](#) (see page 116).

- You can view any logon failures and connection attempts by illegal IP addresses using the Windows Security Event log on the CA Automation Point server.

Global and Local Session Settings

Use the Permission Level field in your global or local session definitions to specify access permissions for a session. You can apply this setting to all defined sessions or to a specific session.

The following levels of access permissions are available:

VIEW

All users can view the session, but they cannot issue commands.

EXEC

All users can view the session and issue commands through the command dialog.

FULL

All users are granted FULL access and can issue commands directly through the session.

NONE

No users are granted remote access.

FILE

The level of access is based on the user's permission level for a file that you specify.

Note: To set this level of access, in Configuration Manager, navigate to Expert Interface, Automation, Remote Viewing, Remote User Login Security, and make sure that Windows Security is selected.

To allow only one user at a time to control a session, you can enable Primary Control Mode. For more information, see [Primary Control Mode](#) (see page 121).

CA AP Remote Manager and Windows Security

When CA AP Remote Manager is configured to use Windows security, CA Automation Point prompts each Remote Viewer connection for a valid Windows user ID and password. However, Single Sign-on usually does not require a user ID and password. Valid user IDs and passwords are defined on the CA Automation Point server using the Windows User Manager and in the Active Directory database when the CA Automation Point server is part of a Windows domain.

CA AP Remote Manager can be configured to use Windows security with a specified Windows file. CA Automation Point maps the access permissions that are specified for the Windows file to correspond with the sessions' access permissions. User-level access permissions to the file are defined using the NTFS Windows File System. For a session defined with a Permission Level of FILE, the Remote Manager checks the specified Windows file access permissions before granting access to the session.

Security and the Windows Operating System

When Windows security is selected in the CA Automation Point Remote Viewing dialog, use Windows User Manager and NTFS Windows File system to define user IDs, passwords, groups, and access permissions.

Windows facilities serve the following functions for CA Automation Point Remote Viewing:

- Specify valid user IDs and passwords on the local machine so CA Automation Point can validate Remote Viewer logon before establishing a connection.
- Specify file-level access permissions by user ID or group so CA Automation Point can map them to correspond with its session-level access permission by user ID or by group.

Adding Users

You can add users and groups on the CA Automation Point workstation from Windows.

To add users on the CA Automation Point workstation

1. Open a Windows command prompt and execute the following command:

```
mmc.exe %windir%\system32\lusrmgr.msc
```
2. In the Local Users and Groups window, open the Users folder.
3. On the menu bar, click Action.
4. From the drop-down menu, choose New User. The New User dialog displays.
5. In the New User dialog, specify the appropriate information in the following fields:
 - User Name
 - Full Name
 - Description
 - Password
 - Confirm Password
6. Select or clear the check boxes (as desired) for the following fields:
 - User must change password at next logon
 - User cannot change password
 - Password never expires
 - Account is disabled

7. To create an additional user, click Create and then repeat steps 2 through 4. If you are not creating any additional users, skip this step and go to step 8.
8. To finish, click Create and then click Close.

The new users that you created appear in the Users folder in the console tree.

Adding Groups

You create new group IDs to associate Windows file permissions with a group rather than just a single user.

To add groups on the CA Automation Point workstation

1. Open a Windows command prompt and execute the following command:

```
mmc.exe %windir%\system32\lusrmgr.msc
```

2. In the Local Users and Groups window, open the Groups folder.
3. On the menu bar, click Action.
4. From the drop-down menu, choose New Group. The New Group dialog displays.

Note: To add one or more users to a group, click Add in the New Group dialog.

5. In the New Group dialog, specify the appropriate information in the following fields:

Group Name

Specifies the name for the new group.

Description

Describes the new group.

6. To create an additional group, click Create and then repeat steps 2 through 4. If you are not creating any additional groups, skip this step and go to step 7.
7. To finish, click Create and then Close. The new group(s) that you created will appear in the Groups folder in the console tree.

Defining Access Permissions

To define access permissions for a session for which you have specified FILE in the Permission Level field in the global or local session definition, you create a file on an NTFS file system. (The FAT file system does not support security.)

CA Automation Point maps the access permissions defined for this file to correspond with session-level access permissions.

We recommend that you create all security files outside of the Site directory. This prevents those files from being overwritten when you import or export other site configuration settings and preserves their security access permissions. When you import a saved Site configuration, all files in the Site directory are deleted, and new files are extracted to the Site directory. The import operation resets all permissions of all files and directories to be inherited from the parent folder.

Important! Before you assign access permissions to selected users or groups, see the following table. The table shows how CA Automation Point maps assigned Windows file permissions to correspond to sessions' access permissions.

Windows Permission	CA Automation Point Session Permission	Default Permission Level
Read	User can view sessions but not issue commands.	VIEW
Read & Execute	User can view sessions and issue commands through command dialog.	EXEC
Full Control	User can view sessions and type directly into console window.	FULL

If none of the permissions are checked, the user has no access to the session. The permission level then is NONE.

Note: If you are specifying permissions for groups, and the groups being added to the permission set are Domain level groups, ensure that you give the Domain group the "Logon locally" right on the CA Automation Point machine. This is a requirement for Domain level groups.

Single Sign-on

When Windows security is selected on the Remote Viewing configuration dialog, it is possible to enable the Single Sign-on feature. This feature allows a Windows user to open a Remote Viewer session without manually specifying login credentials. The Windows workstation can be part of a Windows domain and can use an Active Directory user accounts to take full advantage of this feature.

A requirement for this feature is a valid Service Principal Name (SPN) registration in the Active Directory database for the designated domain. The CA Automation Point administrator can select automatic registration by the Remote Manager or select manually registration by the domain administrator.

Automatic SPN registration

The default Logon user account for the CA-AP Remote Manager service is the Local System account. The Local System account by definition has all the required *privileges for performing* automatic SPN registration.

If a different Logon user account is specified for the CA-AP Remote Manager service using the Windows Control Panel, ensure that it is granted the required privileges. In this case, the Logon user account that is specified for the CA-AP Remote Manager service must have the *Write ServicePrincipalName* privilege that is assigned by a domain administrator.

Enable automatic SPN registration by selecting the Automatic radio button under the Service Principal Name group box. With automatic registration, the Remote Manager service registers an SPN in the Active Directory database in the following form:

apview/<hostname>

Manual SPN registration:

Under normal circumstances, use automatic SPN registration. If automatic SPN registration is not **possible** or a different SPN format is required, use the manual SPN registration process.

Enable manual SPN registration by selecting the Manual radio button under the Service Principal Name group box. Enter the SPN, manually registered by the domain administrator, into the Service Principal Name edit box. This registration can be done using a utility such as setspn.

Note: If the specified SPN is missing from the Active Directory database, the feature can fail under certain circumstances.

Command-Level Security for the Merged Msg, AP Msg, and Command Windows

The command area that displays on the Merged Msg window, the AP Msg window, and the Command window can be used to issue commands to any automated session. By default, remote access to these windows is unrestricted (FULL permission). If the permission level of any of these windows is either FULL or EXEC, the remote user has the ability to type into the command area of that window. If the command area is in the session command state, the remote user will only be able to submit the session command if this user has at least EXEC permission for the session that is the target for the command. If the permission level for the window containing the command area is VIEW, the remote user will not be able to type into the command area, no matter what permission level is assigned to that user for the target session. This permission level of VIEW also prevents the remote user from using the command area on that window to execute REXX programs or scripts.

Be sure to secure access to these windows by specifying an appropriate permission level for each window containing a command area defined in the session definition set.

Primary Control Mode

When multiple remote viewer users interact simultaneously with a remote host over the same session, enable Primary Control Mode to manage contention for session access.

While Remote Manager operates in Primary Control Mode, only one remote user with primary control, is permitted to submit host commands at a time. All other remotely connected users with secondary control are prohibited from submitting host commands. At any time, remote users can dynamically request or relinquish primary control using session window menu options. Remote Manager processes those requests and transfers primary control according to policy specified by Primary Control Mode settings for each session.

Configuration

To enable Primary Control Mode for a session, checkmark the option, Enable Remote Viewer Primary Control Mode in the session definition's session settings.

When Primary Control Mode is enabled, only one remotely connected user can have primary access (EXEC or FULL) to the session. All other users have secondary access (VIEW or NONE). No user can gain a higher access level than entitled to by settings. Only users that have EXEC or FULL permission to a session can gain primary control.

The following configuration options are available:

Enable Primary Control Mode

Specifies if the Primary Control Mode feature is enabled.

Primary Control – Required Permission

The minimum privilege that is required for Primary Control can be set to EXEC or FULL.

Primary Control – Grant primary control to the first user

This option specifies whether an eligible remotely connecting user automatically gets Primary Control when there is no other user with Primary Control at the time. This option is forced enabled when the Secondary Control – Highest Permission option is set to NONE.

Secondary Control – Highest Permission

The Highest (effective) permission level a remotely connected user can have while in Secondary Control of a session. This permission is NONE or VIEW. If a user normally has a higher permission level, it is lowered to the one set in this option.

Secondary Control – Auto-request primary control

This option specifies if a user remotely connecting to a session where another user already has Primary Control, can automatically request a Primary Control transfer. This option is valid only when the Secondary Control – Highest Permission option is set to NONE.

Request Timeout – Timeout duration

Specifies the amount of time in seconds a user has available to respond to a Primary Control transfer request before deemed unresponsive, at which point the configured timeout action is taken. When set to zero, the timeout action takes place immediately.

Request Timeout – Timeout action

CA Automation Point server uses this action when a Primary Control transfer request times out. This permission can be KEEP or YIELD. The KEEP value means that the request is denied. The YIELD means that the request is accepted.

Remote Viewer Session Windows in Primary Control Mode

When Primary Control Mode is enabled for a given session, the control state for the session connection is displayed in the session window's title bar. For a detailed description of each status indicator, refer to Remote Viewer Help. From the AP Viewer Msg window, select Help, AP Viewer Help.

Remote users are expected to refer to the title bar to view their current session control status. Given their current session control state, a remote user can initiate actions to change it. From the session window's Action menu, a user can select options to yield primary control or request primary control. While the request is being processed, the remote user can view the AP Viewer Msg window for Primary Control Mode messages reporting results. For a detailed description of how primary control is transferred among remote users, refer to Remote Viewer Help. From the AP Viewer Msg window, select Help, AP Viewer Help.

Securing Remote Viewers to Interface with CA OPS/MVS

CA AP Remote Manager enforces Remote Viewer user logons and access permissions according to the security policy set for the CA Automation Point workstation. After the username is validated and access permissions are granted, requests to issue commands through the CA OPS/MVS Interface are sent and identified with the username issuing the request. After the request is received by CA OPS/MVS, it is subject to the security policy set on the local platform.

CA OPS/MVS OSFSECURITY Parameter

When the CA OPS/MVS OSFSECURITY parameter has a value of CHECKUSERID—The following rules apply:

- When you use Remote Viewer to access CA Automation Point sessions, all commands are associated with the Remote Viewer username. The remote Viewer username overrides the CA OPS/MVS APDEFAULTUSERID parameter, and must conform to the restrictions of your z/OS security package.
- When you use Remote Viewer to access CA Automation Point sessions, all commands are associated with the Remote Viewer username. The remote Viewer username overrides the CA OPS/MVS APDEFAULTUSERID parameter, and must conform to the restrictions of your z/OS security package.
- The Remote Viewer username(s) should be granted the proper security access on z/OS.

When the CA OPS/MVS OSFSECURITY parameter has a value of NOSECURITY—Commands sent from CA Automation Point to CA OPS/MVS execute with the security attributes of the OSF TSO servers.

For details about these parameters, see the CA OPS/MVS documentation.

Starting Remote Operations

To access CA Automation Point from a remote workstation.

1. Ensure that the CA Automation Point host machine has been enabled for remote viewing using Configuration Manager.
2. Establish connectivity through TCP/IP or an internet provider.
3. Install the Remote Viewer component from the CA Automation Point client.
4. From Windows, click Start on the taskbar, and then choose Programs, CA, CA Automation Point, Remote Viewer. The AP Viewer Msg window displays.
5. To connect to a CA Automation Point machine, choose Open Remote Connection from the Action menu. The Connect to Automation Point dialog displays.
6. In the Host field, enter the name or host address of the CA Automation Point host machine that you want to connect to.

You can use the optional `:port` operand to specify the port number to which you want to connect. If CA Automation Point has been enabled for remote viewing using a port other than the default, specify that port number.

7. In the Session field, specify the name of a CA Automation Point desktop window you want to view remotely. Guidelines for specifying different session types follow:

- For terminal emulator sessions:

Prefix the session name with Session. For example, to connect to a session named RCS2 in the session definition that is set, specify Session RCS2.

- For function windows

Enter the name or title of the function window as it appears on the CA Automation Point desktop. The default window names follow:

- AP Msg Recall
- Merged Msg
- Action Msg Recall
- Host Log
- AP Log
- Normal Msg Recall
- AP Msg
- Command
- AP Notification Messages

Notes:

- The Session field is case-sensitive.
 - If you do not specify a value for the Session field, by default, CA Automation Point opens AP Msg Recall session.
- 8. If Single Sign-on is configured and you want to use a different user account set the Use Explicit Credentials checkbox.
- 9. Click Connect.

If session-level security is configured, the Automation Point Login dialog displays.
- 10. Enter your user ID and password, if requested.
- 11. After you are connected to one CA Automation Point machine, you can connect to others from the AP Viewer Msg window by choosing Action, Open Remote Connection.

To start CA Automation Point remotely, click Startup. You can specify the optional parameter *session definition set name* in the Start field when you want to start CA Automation Point with a particular session definition set.

For example, to start CA Automation Point with the session definition set, SessDefSet, you would specify this name in the Start field of the Connect to Automation Point dialog.

Note: If you do not specify any session definition set in the Start field, the active session definition set is used to start CA Automation Point.

After you are connected to one CA Automation Point machine, the AP Message Recall window or the window specified in the Session field displays. You can view other session and function windows by choosing the Jump to the window option from the Window menu.

Starting the Remote Viewer Using a Command Line

The following two examples show ways of starting CA Automation Point remotely using a command line. These are alternatives to using the dialog discussed previously.

Examples

- To connect to the host named DataCenter, start CA Automation Point (if it is not already running) and view the Merged Msg window:

```
APVIEW /hDataCenter /s"Merged Msg" /x
```
- To connect to the IP address 172.24.123.456 on port 7000, start CA Automation Point (if it is not already running) and view the session named PCON:

```
APVIEW /h172.24.123.456:7000 /s"Session PCON" /x
```

In the preceding two scenarios, the /x switch without the session definition set name causes the active session definition set to be used.

Notes:

- The name of the function window or session must be in double quotes. Single quotes are not allowed.
- To connect to a session, you must prefix the name of the session with the word "session."

To remotely start CA Automation Point using a different session definition set, first verify that the following conditions are true:

- The new session definition set exists
- CA Automation Point is not running

If either of the conditions listed previously are not met, *you will not be able to access CA Automation Point from the remote workstation using a different session definition set.* Someone at the host workstation site must then either terminate CA Automation Point so that you can try again, or you can restart CA Automation Point using the different session definition set so that it is already running when you connect to the host workstation. For example:

```
APVIEW /hDataCenter:7000 /x"SessDefSet"
```

In the preceding example, the session definition set, SessDefSet is used to remotely start CA Automation Point.

Note: You must enclose the name of the session definition set in double quotes. Single quotes are not allowed.

Moving Between Sessions After Initiating the Remote Viewer

You can select a new window for the sessions that are managed by the same CA Automation Point host by choosing Jump to Window from the session Window menu.

In the Jump to Window dialog that displays, select the desired window from the list and click OK (or press Enter).

The selected window displays on your screen.

Issuing Commands to a Session Using the Command Dialog

You use the Command dialog to issue commands to a session. You can use this dialog with sessions that are automated, controlled, or not controlled. A key benefit of this dialog is that it allows sessions to be shared by more than one remote user.

To display the Command dialog use the hot key (Alt+X) or choose Command Dialog from the SessCntl menu. When selected from a Session window, the dialog prompts you for a character string to send to that session without the possibility of intermingled characters.

Note: The Remote Viewer permits multiple users on their own remote workstations to access the same CA Automation Point host session. When multiple users type directly into a shared window, they risk intermingled characters because CA Automation Point is managing only one physical console for each session. When remote viewing is enabled, CA AP Remote Manager provides a view of the same physical console to all Remote Viewers.

The command dialog has the following buttons:

OK

Submits the command and closes the dialog

Submit

Submits the command and keeps the dialog open

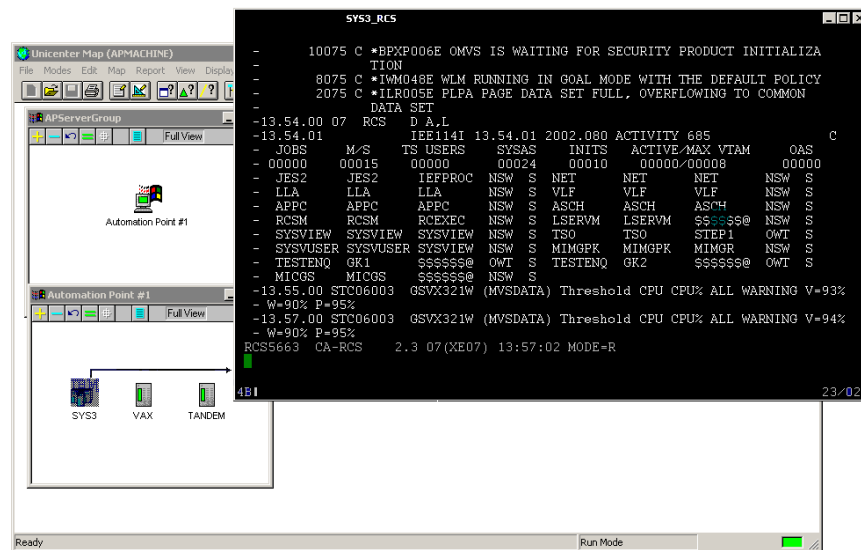
Cancel

Closes the dialog without submitting the command

Launching the Remote Viewer from the CA NSM WorldView Map

You can launch the Remote Viewer from a WorldView Map connection from the menu of the icon that represents the system you want to access. The simplest way to customize CA NSM WorldView for this type of access is to import CA Automation Point classes, and then create objects with icons on the map for your managed sessions. These icons will have a menu (accessed by right-clicking the mouse) that contains a "Go to console" item for entering the Remote Viewer.

The following windows illustrate this use:



Note: The Remote Viewer component must be installed on the machine containing the CA NSM WorldView Map. For instructions on configuring CA NSM Objects for CA Automation Point, see the chapter "[Interacting with External Event Systems](#). (see page 351)"

Web Message Viewer

This section discusses the Web Message Viewer.

What is Web MV?

Web Message Viewer (Web MV) provides a common, remotely accessible message window that allows you to view all of the messages received and processed through rules by CA Automation Point in as close to real time as possible. This includes not only the messages received from CA Automation Point-automated sessions, but messages generated by CA Automation Point also.

In addition to viewing CA Automation Point messages, Web MV allows you to access detailed information about each message, separated into logical columns and displayed according to your specification. Web MV also allows you to specify the amount of messages you want to store in a database, allowing you to scroll back as far as you need to view previous messages.

With Web MV, you can also:

- Set the visual attributes of messages (including color and font)
- Send commands to monitored sessions
- Issue CA Automation Point DOM requests for action messages

Areas of Web MV

Web MV can be broken into the following four major areas:

Web MV Command Processor

Receives messages from the CA Automation Point rules engine and passes them to the Web MV Web server. It only receives CA Automation Point messages that are eligible for Web viewing.

For more information on the eligibility requirements for Web MV messages, see [Messages Eligible for Web Viewing](#) (see page 131) in this chapter .

Web MV Web Server

Receives new messages from the Web MV Command Processor and receives requests from remote Web MV clients. All new messages received from the Web MV Command Processor are written to the Web MV database using the Web MV Database Interface.

Web MV Database Interface

Archives and recovers CA Automation Point messages that are selected for remote Web viewing. The Web MV Web Server uses this interface to store and retrieve CA Automation Point messages. This interface is also used to store individual user profiles for remote Web MV clients.

Web MV Graphical User Interface (the Web MV main window)

The primary graphical user interface (GUI) for viewing Web MV messages. It communicates directly with the Web MV Web Server.

How Does Web MV Work?

This section explains Web MV message transport, Web MV data source, and Web MV message processing.

Web MV Message Transport

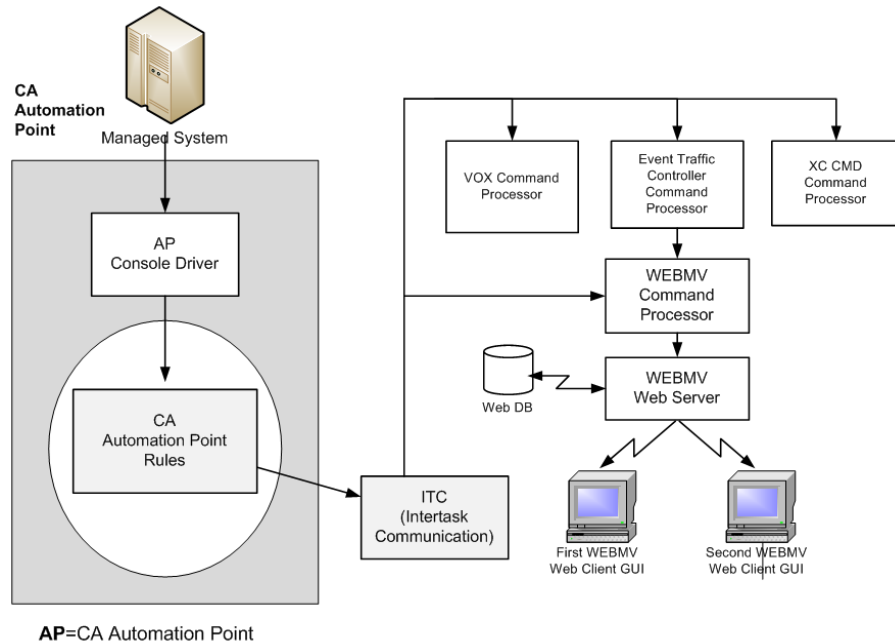
To communicate with the CA Automation Point server machine, Web MV uses an approach similar to that of the Remote Viewer. The steps in the communication process are:

1. From your workstation, request an HTML page from the Microsoft IIS Server that is currently running on the CA Automation Point server machine.
2. The HTML page allows you to download and launch the Web MV GUI.
3. The Web MV GUI establishes a connection with the CA Automation Point Web Message Server service that is running on the CA Automation Point server machine.
4. The CA Automation Point Web Message Server service receives and processes requests from your workstation and responds to you with the requested session information.

Processing Messages Forwarded from Rules

Web MV may resemble the Remote Viewer regarding message transport, but the data source for the Web MV is completely different. Unlike the Remote Viewer, Web MV receives messages on a line-by-line basis. The messages Web MV receives are those forwarded from within the CA Automation Point message parsing routines, which are tightly integrated with CA Automation Point rules processing.

On the CA Automation Point server, a separate command processor is provided to asynchronously process the messages forwarded from rules processing. (This approach closely mimics the way CA Automation Point forwards messages to a CA NSM Event Management Console.) The following diagram explains this concept:



Unlike the process CA Automation Point uses to forward messages to a CA NSM Event Management Console, Web MV does *not* use rules keywords to enable the propagation of messages. The Enable Web MV field in the local session settings of the session definition set determines whether the specified session forwards messages through the Web MV Command Processor.

Note: You can enable Web Viewing in the global session settings to apply it to all defined sessions, unless specifically overridden at the session level.

Messages Eligible for Web Viewing

For Web MV to receive a session's messages, that session must either be defined as automated, or managed by an external REXX program that sends messages through the CA Automation Point rules engine using the CA Automation Point MSG command processor.

Only messages that originate from Web MV-approved sessions (sessions for which Web MV has been enabled) are forwarded to the Web MV Command Processor. Once inside the Web MV Command Processor, these messages are sent to the Web MV Web Server, which temporarily stores them in the Web MV database. The Web MV Web Server then forwards these messages to all Web clients who are currently registered to receive messages from the specified session.

The AP Msg Recall, CA-OPS/MVS Messages, and AP Notification Messages windows are internal sessions. The AP Msg Recall window is *not* configurable from the session definition set and will *always* appear on the CA Automation Point desktop. The CA-OPS/MVS Messages and AP Notification Messages windows are configurable under function windows. These three sessions are Web MV-enabled through the Configuration Manager Web Message Viewer dialog. All other sessions *must* be web-enabled through their session definitions in the session definition set in order to be viewed from the Web MV graphical user interface.

Note: For information on how to enable the AP Msg Recall, CA-OPS/MVS Messages, and AP Notification Messages sessions using Configuration Manager, see the section [Enabling Web MV](#) (see page 133) in this chapter .

Important! Any time you pause a session from CA Automation Point, the word PAUSED displays on the top of the Web MV GUI screen at the end of the toolbar. While a session is paused, you will not be able to view its messages, because messages of paused sessions do not go through CA Automation Point Rules, and therefore are not forwarded to Web MV.

Timely Delivery of Messages

To allow you to receive CA Automation Point messages in as close to real time as possible, a message subscription convention exists between you and the CA Automation Point Web Message Server service. When you request messages of a given type (based on session name), all messages matching this request are forwarded to you by the CA Automation Point Web Message Server.

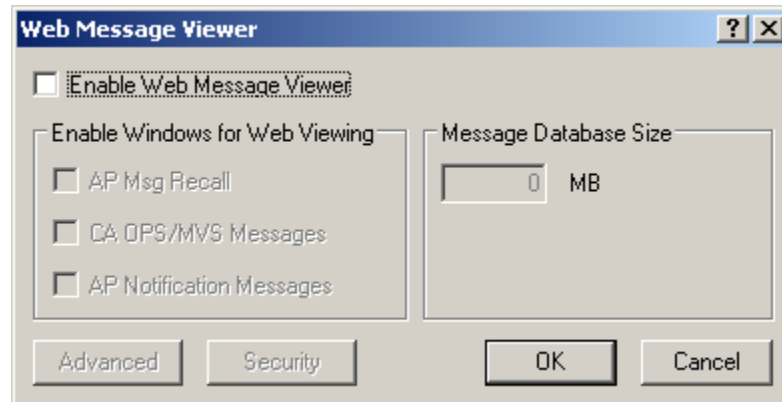
Installing Web MV

The installation process for Web MV is very simple, because you do not have to install any software remotely. As long as your remote client workstation has a Web browser, Web MV can be automatically downloaded from the server and run on the local machine. This also reduces your installation efforts for future versions, because the Web browser will download the new versions and the software will not have to be updated manually at each specific workstation.

For software requirements for Web MV, see the *Release Notes*.

Enabling Web MV

To access the Web Message Viewer dialog and configure your Web MV options, choose Expert Interface, Automation, and then Web Message Viewing from the Configuration Manager main window. The Web Message Viewer dialog displays:



Check the Enable Web Message Viewer check box to activate Web MV. After Web MV is activated, you can change the configurable parameters pertaining to Web MV.

Web MV Dialog

The areas of the Web Message Viewer dialog are as follows:

- The Messages Database Size area allows you to specify the amount of disk space that you want to allocate to store CA Automation Point messages. The higher this value, the more messages you can store in your database. You will only be able to view as many messages as can be stored in this database. For example, if you specify that you want to allocate 10 MB, you will be able to view 12,771 messages at one time. If try to view message 12,772, Web MV overlays message 1 with message 12,772. If you then decide to view message 12,773, Web MV will overlay message 2 with message 12,773, and so on. In this scenario, Web MV will keep overlaying messages, only allowing you to view 12,771 messages at a time.

You can specify a maximum of 16,000 MB. We recommend that you allocate a minimum of at least 10 MB.

Important!

- The database resides in the directory *installDir\STOREDAPMSG.S*. This directory is created on the hard drive on which CA Automation Point is installed.
- Web MV deletes the existing database before creating a new database with an updated message capacity.
- Web MV deletes the existing message database when Web Message Viewer is disabled, but saves the configuration settings for future use. These configuration settings are used when you next activate the Web Message Viewer.

- The Enable Windows for Web Viewing area of the dialog allows you to specify which CA Automation Point-managed sessions are eligible for Web Message viewing. Because these CA Automation Point sessions are not defined in the session definition set, these check boxes provide the same functionality as the Enable Web MV field for user-defined sessions. To view messages from these sessions, you must select the appropriate session name in the Web Message Viewer application. The following session names are assigned to these CA Automation Point-managed sessions: AXC (AP Msg Recall), OPS (CA OPS/MVS Messages), and VOX (AP Notification Messages).

- The Advanced button displays the Advanced dialog, which allows you to further configure Web MV.

For information on the Advanced dialog, see the Configuration Manager help.

- The Security button displays Remote Security dialog, which allows you to configure security for Web MV.

Note: For information on the Remote Security dialog, see the Configuration Manager help.

Web MV Security

Web MV uses the same security as Remote Viewer. However, Web MV does not use all of the permissions handled by Remote Viewer and does not support the Single Sign-on feature.

Note: To learn how to set up the security for Web MV, see the section [Securing the Remote Viewer](#) (see page 114) in this chapter. For Web MV-specific security issues, see the following sections.

Important! CA AP Remote Manager Service is required for Web MV security to work.

User ID and Configuration Protection

Each remote user of Web MV must sign on with a valid user ID. The user ID determines the level of access granted to the remote user. The CA Automation Point server machine also uses this user ID to create an individual user profile. Each user profile contains settings for all of the Web MV client-configurable options. These configuration settings are read by the CA Automation Point Web MV Web Server and passed to the Web Message Viewer graphical user interface upon client initialization.

For a detailed explanation of the login process, see the section [Connecting to Web MV](#) (see page 135) in this chapter.

Protection by Level of Access

As mentioned previously, remote users are granted access to Web MV in different levels. These levels of access affect Web MV security as explained in the following tables.

This table explains the levels of access permitted for Web MV sessions by Windows:

Windows Special Access	Web MV Session
Deny	Permission Level is NONE. That session will not be displayed on the session drop-down list on the user's remote workstation.
Read	Permission Level is VIEW. The user can view sessions but cannot issue commands or DOM action messages.
Write	Permission Level is FULL. The user can view messages, issue commands, or DOM action messages.
Execute	Permission Level is EXEC. The user can view messages, issue commands, or DOM action messages. Note: For Web MV, Write and Execute Permission are the same.

Connecting to Web MV

To connect to Web MV

1. To launch Web MV from a remote Web browser, type in the URL that includes the names of your default virtual directory and your CA Automation Point server machine, followed by /WebmvGUI.html.

For example, if you choose the default virtual directory name AP, and the name of the CA Automation Point server machine is appgh1, the following URL launches Web MV from a remote Web browser:

```
http://appgh1/AP/WebmvGUI.html
```

The home page of the Web MV GUI displays.

- The Web MV connection dialog is automatically launched to establish a connection to a CA Automation Point server. When a Java Runtime Environment is not installed on the computer, instructions for downloading a JRE are displayed. A link to the JRE download website is also displayed.

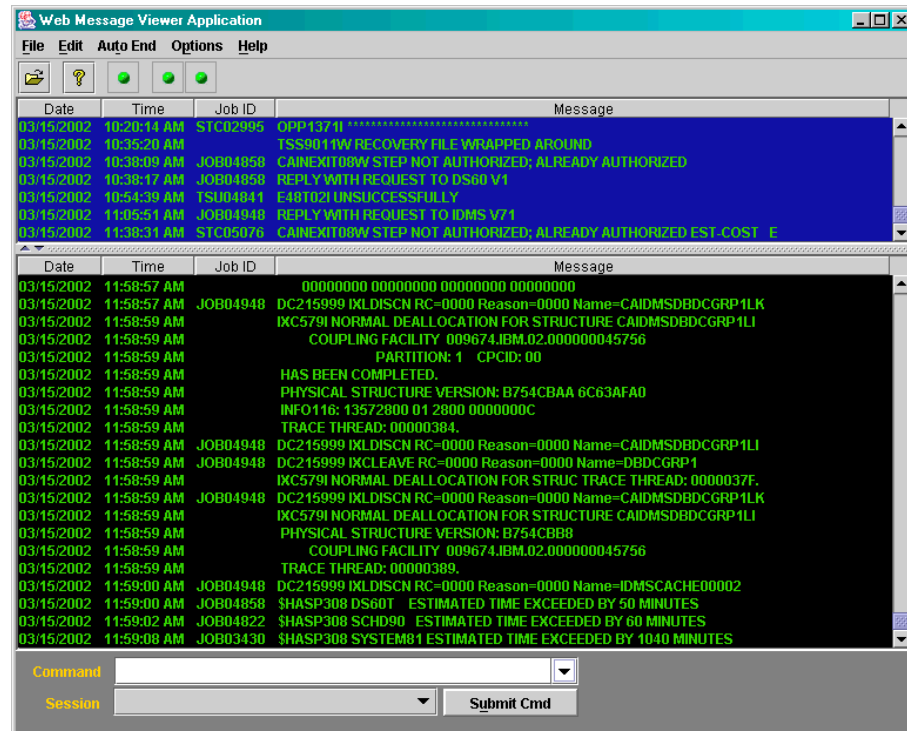
Important! You must have internet access to download the JRE.

Note: During the installation of the JRE, the Java Plug-in is set as the default Java runtime for your web browser. You can remove this default setting by using the Java Plug-in Control Panel.

After you have connected to a CA Automation Point server, the Web MV main window displays.

Web MV Main Window

The following is an example of the Web MV main window:



Areas of the Web MV Main Window

The Web MV main window consists of the following three areas:

Action Messages table

Displays action (highlighted) messages. You can delete messages from this table. If you do, the corresponding messages will not be deleted from the Normal Messages table, so you will still be able to view them from there. Deleting a message from this Action table sends a CA Automation Point DOM request for the selected message. The Action Messages table is located under the menu bar in the top portion of the Web MV main window.

Normal Messages table

Displays both action (highlighted) *and* normal (non-highlighted) messages. The Normal Messages table is directly underneath the Action Messages table.

Note: Web MV allows standard Java table controls and edit box handling; for example, you can alter the view size of the tables or independently resize the table columns.

Session and Command area

Allows you to either submit a session command to the current session or change the currently monitored session. The command area portion of the window is located directly beneath the Normal Messages table, at the very bottom of the window.

Permission to submit commands is granted to remote users on a session-by-session basis and is validated through the same CA Automation Point security as the Remote Viewer. If you submit a session command, it is sent to the Web MV Web Server service, which sends it to the CA Automation Point engine, where it waits in queue to be processed.

Note: You can submit a specified command to more than one session by selecting ALL from the session drop-down list. The ALL option allows you to issue a command to one or more of the sessions for which you have command execute permission. For more information about using the ALL option, see the section [The Built-in ALL Option](#) (see page 139) in this chapter.

For more information on CA Automation Point session level security, see the section [Securing the Remote Viewer](#) (see page 114) in this chapter.

Menu Bar

The following information describes the options you can access from the menu bar of the Web MV main window.

- The File menu at the top left of the Web MV main window contains three options: Connect, Disconnect, and Exit.
 - The Connect option displays the host connection dialog, where you specify a login ID and password.

Important! The server name cannot be changed because the host must be the CA Automation Point server from which this application was launched.
 - The Disconnect option closes the connection to the host and clears the Action and Normal Messages tables.
 - The Exit option exits the Web MV main window.
- The Edit menu contains two options:
 - The Copy option allows you to copy the text of the selected message(s) in the Action and Normal Messages tables to the system clipboard.
 - The DOM Action Msg option. This option lets you issue a DOM request for each selected message in the Action Messages Table.

Note: To use the DOM Action Msg option, you must have the authority to issue commands to the currently selected session and at least one Action message must be selected from the Action table.
- The Auto End menu allows you to display and set current table modes. If a table is in Auto End mode, it automatically scrolls to the last message received from the CA Automation Point server. By default, both the Action Messages and Normal Messages tables are placed in Auto End mode each time Web MV is launched or a new session is selected from the Session drop-down box in the Command area of the Web MV main window.
- The Options menu contains configuration options used to customize messages. From this menu, you can access the Visual Attributes dialogs. For descriptions of these dialogs and the visual attributes that you can configure with them, see the section [Configuring Visual Attributes](#) (see page 142) in this chapter.

Viewing Messages with Web MV

Filtering Messages

You can filter the messages you want to view based on their session names. Each Web MV client can either specify a session name for message filtering or select the built-in ALL session.

Note: All CA Automation Point messages originating from Web MV-enabled sessions are stored in a relational database for later retrieval. For detailed information about this database and how to select the volume of messages you want to store, see the section [Enabling Web MV](#) (see page 133) in this chapter.

Viewing Previous and Newer Messages

The Web MV Action Messages and Normal Messages tables each allow you to view up to 500 messages from the CA Automation Point server at any given time. However, you can scroll back or forward to view previous or newer messages in each table, using the scroll bar on the right side of the table:

To view previous messages, do one of the following:

- Drag the scroll box to the very top of the scroll bar
- Click the Up arrow

A previous message request is automatically generated and the previous 100 messages are added to the top of the table.

To view newer messages, do one of the following:

- Drag the box to the very bottom of the scroll bar
- Click the Down arrow

A newer message request is automatically generated and the next 100 messages are added to the bottom of the table.

Note: This is only necessary if the table is not in Auto End mode. If the table is in Auto End mode, it will automatically scroll to the latest message received from the CA Automation Point server.

The Built-in ALL Option

ALL is an option on the session drop down list in the Session and Command area of the Web MV main window. The ALL option allows you to:

- View messages
- Submit commands

Viewing Messages with ALL

When you select the built-in ALL option from the session drop-down list, the Action and Normal message tables display the messages received from all sessions that you have permission to view. This allows you to view messages from more than one session.

Submitting Commands with ALL

The ALL option also allows you to issue a single command to more than one session. To issue the specified command to more than one session, select ALL from the session drop-down list, type the session command in the command area, and then click Submit Cmd.

The Multiple Session Selection dialog displays, allowing you to choose which sessions will receive the session command that you have specified.

Notes:

- Every time you select the ALL option, and you want to issue a session command, the Multiple Session Selection dialog displays, allowing you to choose which sessions will receive the session command that you have specified.
- Depending on your Web MV security configuration, the Target sessions for command list may not be the same as the list in the session drop-down list in the Session and Command area of the Web MV main window. Consequently, you may be able to view certain sessions, but be unable to issue commands for them.

For more information about the fields in the Multiple Session Selection dialog, see the Web Message Viewer HTML Help.

Messages Columns

The Web MV main window displays messages in two tables, the Action Messages table and the Normal Messages table. Each table displays one line for each message that is received. Each line is then separated into columns of information for that message.

The information found in the columns for each message is displayed in the following table:

Column Name	Information Displayed
Session	Name of the CA Automation Point session from which the message was captured
System	Processor ID of the host system (if available), or host system name as defined in the CA Automation Point session definition (SYSNAME)

Column Name	Information Displayed
Date	Current system date on the CA Automation Point server
Time	Current system time on the CA Automation Point server
Host Time	Time at which a message was received from the host (if available)
Job ID*	Current JES job ID associated with the message
Job Name*	Job name of the address space that issued the message
EMS Date*	Date contained in first eight characters of a TANDEM EMS message
JES3 Name*	System name of a JES3GLOBAL or sysplex console
MON Name*	Source system name for a DataFrame message
MON Num*	Source system number for a DataFrame message
MON Prty*	Priority of the alert message issued by the DataFrame system
MON Type*	Type of alert message issued by the DataFrame system
Act Ind*	Character used to denote action messages on the host system (for action messages only)
Msg Num	ID of the message (assigned by CA Automation Point)
Seq Num	Sequence ID of the message (assigned by CA Automation Point) Note: This number is greater than zero for multi-line messages.
Message	Text of the message
Reply ID*	Reply number for a message that is a WTOR
Sess Type	Type of session as defined by the Console Type control on the Session Definition window
Act Msg	Specifies whether this message is currently an action message (in Y/N format)

* These columns may not contain data, depending upon the type of message.

For information on how to choose the columns you want to display, see the section [Specifying Table Columns for Your Messages](#) (see page 143) in this chapter.

Configuring Visual Attributes

From your workstation, you can configure the visual attributes of the CA Automation Point messages displayed in the Web MV main window. The attributes that you configure are then saved as part of the user profile.

For each subsequent user connection, Web MV displays the message tables as specified in the user profile. The attributes you can configure are:

- Message font
- Message table columns
- Message color
- Time format

Keep these tips in mind when configuring visual message attributes:

- Visual attribute configurations are saved in your user profile on a per-user, per-session basis
- You can configure visual attributes on a per-table basis, allowing action messages to be different in appearance from normal messages
- Visual attribute settings for a specific table applies to *all* messages contained in that table. You can determine the foreground color of a message by mapping the message color assigned by CA Automation Point rules to a user-selected color. You can perform this mapping through a configuration dialog on the Web MV main window. Color mapping is stored on a per-user, per-session basis. For information on how to choose the columns you want to display, see the section [Message Color Mapping](#) (see page 144) in this chapter.
- Time format is stored only on a per-user basis and is shared across all sessions.

The following sections provide detailed descriptions of the visual attributes that you can configure with Web MV.

Changing Message Font

Web MV allows you to change the font name, font style, and font size of your messages.

To change the font of your messages

1. From the Options menu at the top of the Web MV main window, choose Fonts.

The Message Font Selection dialog displays the currently selected fonts for both the Action Messages and Normal Messages tables. The selected fonts are displayed in *font name, font style, font size* format.

2. Click Change next to the name of the table whose font you want to change.

A table-specific font selection dialog displays. For example, if you click Change for the Action Messages table, the Action Message Font Selection dialog displays

3. Choose a font from the list of currently installed fonts, and select font attributes as desired.
4. Click OK.

By default, the font scheme selected from the Message Font Selection dialog applies only to the currently monitored CA Automation Point session. To apply the font scheme to all sessions available to you on the selected CA Automation Point server, select the Apply to all sessions check box at the bottom of the Message Font Selection dialog.

Specifying Table Columns for Your Messages

Web MV provides several columns of information for each message displayed in the Action Messages and Normal Messages tables. With Message Column Selection, you can select which columns you want to display, depending on the information you want to access.

Column selection applies to *both* tables. For example, if you choose to display the Session, Date, Msg Num, Message, and Sess Type columns for the Action Messages table, those same columns are displayed on the Normal Messages table.

To select the columns you want to display

1. From the Options menu, choose Table Columns
The Message Column Selection dialog displays,
2. Select the message columns you want to display, using the Add and Add All buttons to move column names from the Available Columns list to the Displayed Columns list.
3. Use the Move arrows to change the order in which the columns will appear in both the Action Messages and Normal Messages tables.
4. Click OK.

Specifying the Color of Your Messages

Web MV allows you to specify or change the colors of your messages. This feature may be helpful to you for message organization, viewing, and so on, particularly if you are viewing a large volume of messages.

To specify or change the background color of your messages

1. From the Options menu on the Web MV main window, select Colors.
2. Select Table Colors
The Table Color Selection dialog displays

3. Select the Table Background from the Color Properties drop-down list for the desired message type.
4. Click the corresponding Change button.
The specified Background Color Selection dialog displays.
5. Select a new color for this field, then click OK.

By default, the color scheme selected from the Table Color Selection dialog applies only to the currently monitored CA Automation Point session. To apply the color scheme to all sessions available to you on the selected CA Automation Point server, select the Apply to all sessions check box at the bottom of the Table Color Selection dialog.

For more information about the fields in the Table Color Selection dialog, see the Web Message Viewer HTML Help.

Message Color Mapping

Web MV allows a user to define a specific color to a message. To understand the concept of message color mapping, consider the following scenario:

Jack is responsible for maintaining the CA Automation Point server machine and writing CA Automation Point rules. Jack determines that all messages beginning with IEE136I will appear yellow on the CA Automation Point Merged Messages window. To enforce this, Jack adds the following line in the CA Automation Point rules file:

```
MSGID(IEE136I) COLOR(YELLOW)
```

Susie is a Web MV user viewing a session from the CA Automation Point machine for which Jack is responsible. Therefore, by default, the messages that begin with IEE136I also appear yellow to Susie (because Jack has defined them as yellow). However, Susie can choose to remap the color yellow to another color of her choice; for example, blue. Messages that begin with IEE136I will still appear yellow in the CA Automation Point Merged Message window, but Susie's viewing of Web MV will display them as blue.

To map message colors

1. From the Options menu on the Web MV main window, choose Colors.
2. Choose Msg Color Map.
The Message Color Mapping dialog displays
3. Map a specific color to each of the sixteen CA Automation Point color constants.

4. Click Change.

CA Automation Point remaps the currently selected CA Automation Point color to a "display color" of your choice.

After you make your display color selection(s) (for one or more of the sixteen possible message colors), all messages you receive from CA Automation Point for the selected session will use this new "color map" to determine the foreground color used for displaying these messages.

Note: Message color mapping does *not* change the color attribute originally assigned to each message by CA Automation Point Rules; it just allows you to display each message in the remapped color for your viewing in Web MV.

Reset All

Resets all sixteen-color mapping assignments back to their original default values.

By default, the color scheme selected from the Message Color Mapping dialog applies only to the currently monitored CA Automation Point session. To apply this color scheme to all sessions available to you on the selected CA Automation Point server, select the Apply to all sessions check box at the bottom of the Message Color Mapping dialog.

For more information about the fields in the Message Color Mapping dialog, see the Web Message Viewer HTML Help.

Changing Time Format

WebMV allows you to change between 12 hour and 24 hour time formats. The time format choice is stored on a per-user basis and is applied to all sessions.

Choosing the Right Remote Viewing Tool

When deciding which tool to use to view messages from monitored systems, you can make your selection by comparing the functionality provided by each of these two components. This section should help you make this decision by stating a functionality requirement and suggesting the appropriate viewing tool.

Requirement:

I require the ability to remotely manage the Automation Point Desktop application, including the ability to shut down and restart this application.

Answer:

Remote Viewer. This is the only remote viewing tool that allows you to initiate both a remote shutdown and a remote activation of the Automation Point Desktop application.

Requirement:

I require the ability to change the visual attributes of messages received from an automated session.

Answer:

Web MV. This tool allows you to set visual attributes such as message text font, background table color, message column order, message columns displayed, and size of message display windows, and choose a foreground message text color mapping assignment. Each of these visual attributes is saved on a per-user, per-session basis.

Requirement:

I require the ability to remotely view and manage non-automated sessions.

Answer:

Remote Viewer. Web MV will allow you to view only messages received from automated sessions. Remote Viewer allows you to view and interact with both session and function windows defined to the Automation Point Desktop application.

Requirement:

I require the ability to remotely submit commands to managed sessions.

Answer:

Both. If all of the managed sessions are automated, you can use either Remote Viewer or Web MV to submit commands back to the managed session. If one or more of these sessions are non-automated, you must use the Remote Viewer application to submit commands to these sessions.

Requirement:

I require the ability to remotely initiate the execution of REXX programs on the CA Automation Point server machine.

Answer:

Remote Viewer. By design, Web MV does not natively allow remote users to initiate the execution of REXX programs on the CA Automation Point server. This activity is viewed as an administrative function, better suited for the Remote Viewer application.

Requirement:

I require the ability to allow remote users to see messages received from managed (automated) systems without having to install any remote software directly.

Answer:

Web MV. Because the Web MV client application is designed to run inside the Java Virtual Machine contained within the supported web browser, no additional CA Automation Point software is required on the remote machine.

Requirement:

I require the ability to pause and restart automation, either for an individual managed session or for the entire Automation Point Desktop application.

Answer:

Remote Viewer. The Remote Viewer application allows the remote user to use the session menu system to execute commands, including the ability to pause and restart session automation. If one of the CA Automation Point function windows containing the command area is viewed, the remote user can then either pause or restart automation for all managed sessions. The ability to execute menu commands is controlled by the security assigned to the remote user for the specified window.

Requirement:

I require the ability to view multiple session or function windows from multiple CA Automation Point server machines on the same remote client machine simultaneously.

Answer:

Remote Viewer. Remote Viewer allows you to open connections to multiple CA Automation Point server machines and display any number of session or function windows from each remote server. You can then build your own "virtual" Automation Point Desktop from windows defined on multiple servers by arranging these windows in a tile pattern.

Requirement:

I require the ability to view a large number of messages received from automated sessions over a long period of time.

Answer:

Web MV. During configuration of the Web MV component, you can choose the size of the database used to store messages received from automated sessions. This defined database size is used to determine how many messages will be stored for later retrieval using the Web MV client application. The Web MV user can make use of the scroll bars located on both the Action message area and Normal message area of the Web MV client application to view these previously stored messages. When the Web MV database has reached the defined maximum size, the oldest messages are overwritten with newly received messages. This database implementation of a rolling message log allows you to scroll back through a large number of messages without exceeding the current hard disk capacity.

Chapter 7: Writing Rules

This chapter describes the CA Automation Point rules language. It covers the following topics:

- Understanding the rules language
- Using variables in rules
- Specifying text strings in rules
- Writing time rules
- Writing message rules
- Writing command rules
- Enabling a rules file

Understanding the Rules Language

The CA Automation Point rules language is a set of keywords and symbolic variables that you use to create rules. Through rules, CA Automation Point automates system tasks by managing messages, variables, and commands. Each rule specifies a message, group of messages, a command, a group of commands, or time interval to which CA Automation Point responds. When a message displays in a console session that CA Automation Point controls, it processes any rules related to that message and displays the resulting message (if any) in the Merged Msg window, and in either the Normal Message Recall or the Action Message Recall window. When a user issues a command through CA Automation Point, it processes any rules related to that command.

Types of Rules

CA Automation Point uses three types of rules:

Message rules

Tell CA Automation Point what action to take when a specified message displays on a console.

Time rules

Tell CA Automation Point what action to take when a specified time or time interval occurs.

Command rules

Tell CA Automation Point what action to take when a specified command is issued through CA Automation Point.

Sample actions for these rule types include:

- Displaying console messages in color on the Merged Msg window
- Sounding alarms when critical messages appear
- Suppressing messages from Normal, Action, and Merged Msg windows
- Executing batch files
- Issuing commands
- Suppressing commands from being issued to the host
- Issuing reworded commands
- Replying to WTOR messages
- Issuing a Notification Manager Find request
- Forwarding a message to the CA NSM Event Console
- Writing a message to a Program-to-Program Queue (PPQ)
- Executing REXX programs
- Executing CA Automation Point scripts

Ways to Use Rules

Rules can manage most simple automation tasks and can issue CA Automation Point command processors that automate complex tasks such as:

- Managing a voice response system
- Manipulating the variables that CA Automation Point uses to store system data needed for automation

When an operations task requires multiple actions (separated by required wait states), you can write REXX programs that your rules can invoke.

For more information about REXX programs, see the chapter "[Configuring and Writing REXX](#) (see page 181)."

Rules Keyword Summary

The following tables summarize rules keywords. See the *Command and Keyword Reference Guide* for keyword syntax.

Keywords for Defining a Rule Type

Use the following keywords to define rule type.

MSGID

Defines the beginning of a message rule (which activates when a message having the specified ID displays)

CMDIN

Defines the beginning of a command rule (which activates when the specified command is issued by the user through CA Automation Point)

TIME

Defines the start of a time rule (which activates at a certain time or after a specified time period has passed)

Keywords for Defining Automation Criteria

Use the following keywords to define automation criteria.

EVERY

Specifies how often you want an action to be taken

LIMIT

Specifies how many times a CA Automation Point rule can execute in a given minute

MATCHLIM

Limits the number of times that an action specified by a rule takes effect

SESSION

Restricts the processing of MSGID or CMDIN rules to a given session

WHEN

Defines additional conditions that must be true for a rule to activate

Keywords for Responding to System Events

Use the following keywords to respond to system events.

DOSCMD

Issues an operating system command or executes a command file

OSCMD

Issues an operating system command to a console

PPQWRITE

Writes an item to a PPQ

REPLY

Specifies the reply to a WTOR message

REXX

Invokes a REXX program

SCRIPT

Starts a CA Automation Point script

SESSCMD

Sends a keystroke string to a session

SET

Creates, deletes, modifies, or assigns a value to a status variable

XCCMD

Invokes a CA Automation Point command processor from a rule

Keywords for Controlling the Display

Use the following keywords to control the display.

COLOR

Specifies the color in which you want a message to appear

DISPLAY

Displays a previously suppressed message in the Normal, Action, or Merged Msg window

DOM

Deletes an action message from the Action Message Recall window and from the action message area of the Merged Msg window

HILIGHT

Displays a message in the Action Message Recall window and in the action message area of the Merged Msg window

LOWLIGHT

Displays a message in the Normal Message Recall window and in the main messages area of the Merged Msg window

PREFIX

Specifies the prefix of the messages that appear on CA Automation Point Message Recall and Merged Msg windows

REWORD

Alters the text of a message or a command

SUPPRESS

Prevents a message from being displayed in Automation Point Message Recall and Merged Msg windows; prevents a command from being issued to the host

WTO

Issues a message to the Merged Msg window and either the Action Message Recall window or the Normal Message Recall window

WTXC

Issues a write-to-operator message and sends it to the CA Automation Point Msg window

Keywords for Logging Messages

Use the following keywords to log messages.

LOG

Sends a message to the host message log file or the CA Automation Point message log file

NOLOG

Prevents a message from being sent to the host message log file

PRINT

Prints a message on the hardcopy log

NOPRINT

Prevents a message from being printed on the hardcopy log

Keywords for Notification

Use the following keywords for notification.

ALARM

Sounds an alarm when a specified message occurs

ALARMSAY

Issues a text-to-speech alarm when a specified condition occurs

NOALARM

Specifies that no alarm be sounded when a message displays

Keywords for CA NSM (Event Traffic Controller)

Use the following keywords for CA NSM (Event Traffic Controller).

NOUNIFWD

Tells CA Automation Point *not* to forward a message that is processed by rules to CA NSM

SNMPTRAP

Sends an SNMP trap to the specified host

UNICMD

Tells CA NSM Event Manager, which resides on the specified host, to execute the supplied command

UNIFWD

Tells CA Automation Point to forward a message that is processed by rules to all recorded CA NSM hosts

UNIWTO

Sends the supplied message to CA NSM Event Manager on the specified host

How CA Automation Point Processes Rules Keywords

When processing a rule, CA Automation Point always executes keywords in this order:

type_keyword [*conditional_list*][*output_list*][*action_list*]

type_keyword

Specifies the type of rule. Specify one type keyword per rule at the beginning of a rule. Use one of these keywords:

- MSGID
- CMDIN
- TIME

conditional_list

Specifies the restricting conditions for the rule. Specify each conditional keyword only once in a single rule. The rule condition keywords execute in the order in which they appear in the rule. You can use these keywords:

- EVERY
- LIMIT
- MATCHLIM
- SESSION
- WHEN

output_list

Sets output attributes in CA Automation Point. Specify each output keyword only once in a single rule. The rule output keywords execute in the order in which they appear in the rule. You can use these keywords:

ALARM	NOLOG
COLOR	NOPRINT
DISPLAY	PREFIX
HILIGHT	PRINT
LOG	REPLY
LOWLIGHT	REWORD
NOALARM	SUPPRESS

The rule output keywords set output attributes that can be changed by subsequent rules. When a rule contains one or more output keywords, the conditions they set remain in effect until CA Automation Point executes that rule and all other rules referencing the same message ID or time value.

action_list

Specifies tasks for CA Automation Point to execute. These actions occur or queue for execution immediately and cannot be undone by a subsequent rule. You can specify action keywords multiple times in a single rule. The action keywords execute in the order in which they appear in the rule. You can use the following keywords:

ALARMSAY	SCRIPT
DOM	SESSCMD
DOSCMD	SET
OSCMD	WTO
PPQWRITE	WTXC
REPLY	XCCMD
REXX	

How Rules Are Processed When More Than One Rule Applies to a Message or Command

CA Automation Point processes rule clauses in the following order

- MSGID rule clauses:--MSGID(), MSGID(*string*), MSGID(*)
- CMDIN rule clauses:--CMDIN(), CMDIN(*string*), CMDIN(*)

Creating a Rules File

Use any editor to create your own rules file or to customize the sample rules file (AXCRULES.rul).

When writing CA Automation Point rules, follow this syntax:

- Begin each rule with a MSGID, CMDIN, or TIME keyword; follow it with other appropriate keywords.
- Use positions 1 through 255 of each line.
- Do not split a keyword phrase across lines.
- Separate rules keywords within a line by using a comma, one or more blanks, or both.
- Write a comment line by placing an asterisk character (*) in column 1.

Using Variables in Rules

With some rules keywords, you can use special symbolic variables to store useful system information. CA Automation Point rules can contain two types of variables:

- Environmental variables
- Status variables

Environmental Variables

Environmental variables contain information about the system environment that exists when CA Automation Point processes a rule. For example, a variable can contain the current time or the job ID associated with a message.

Environmental Variable Values

CA Automation Point sets an environmental variable when a rule first references it. If later rules also refer to that variable, the value of that variable remains the same until CA Automation Point processes all rules for a system event. Some environmental variables can have null values. For example, if a message contains four words, the variable &WORD5 is null.

Valid Environmental Variables

Environmental variable names begin with an ampersand (&). The following table lists the valid environmental variables that can be used in CA Automation Point rules. Special notes are included for events that are extracted from CA NSM or the Windows event logs, and for messages generated by CA OPS/MVS.

&AP_ACTIVE_RULES

Specifies the name of the active rules file.

&AP_PARMFILE

Specifies the name of the active session definition set.

&CMD

Specifies the first line of the command text.

&DATE

Specifies the current system date for the workstation in the form *mm/dd/yy*.

&DAY

Specifies the current day of the week for the workstation in the form SUN, MON, TUE, WED, THU, FRI, or SAT.

&DOMID

Specifies the internal pointer to the current action message.

&EMSDATE

Specifies the date contained in the first eight characters of a Tandem EMS message.

&FOCUS_WIN

Specifies the desktop window that is the currently active window. This variable is only set in the menu. It is **unavailable** in rules, REXX, and all other areas of CA Automation Point.

&HAFHMC

Specifies the name of the HMC that generated this HAF message.

&HAFMSGNUM

Specifies a six-digit number that is incremented each time a message is received by HAF from an HMC. Each HMC has its own counter; that is, you will see message number 000001, message number 000002, and so on from each HMC. This value is useful because many HMC messages must be split up into multiple HAF messages so they can be presented properly (see the description of HAFMSGSEQ). This number allows you to relate the pieces of an HMC message to one another in the rules that you write.

&HAFMSGSEQ

A single HMC message often needs to be split into several HAF messages. HAFMSGSEQ is a six digit number that starts with a value of 000001 for each new HMC message. It is incremented for each new message that HAF creates when presenting the HMC message. As with HAFMSGNUM, you can use this to relate the pieces of an HMC message to one another in rules. HMC messages are split for one of two reasons; either the HMC message itself contains formatting characters that request that the HMC message be split into multiple HAF messages, or the HMC message contains more data than can be passed into rules in a single HAF message. Each HAF message in a multi-line HMC message has the same message ID (HAF*xmmnn*) and the same message number (&HAFMSGNUM), but &HAFMSGSEQ will be incremented by one for each HAF message that is generated from the HMC message.

&HAFMSGTYPE

A single HMC message often needs to be split into several HAF messages (see &HAFMSGSEQ). HAFMSGTYPE is set to LAST if this is the only HAF message or last HAF message for an HMC message. It is set to FIRST if this is the first HAF message for an HMC message that has been split into more than one HAF message. It is set to MIDDLE if either of the two preceding conditions is not met.

&HAFOBJECT

In HMC terminology, specifies the name of the object that this HMC message is describing. Typical HMC objects are CPCs, LPARs, profiles, and IOCDSs.

&HOSTTIME

Specifies the time when a message is received from the host or when a command is issued. It is in the form *hh:mm:ss*. If the &HOSTTIME value is not available, CA Automation Point uses the &TIME value.

CA NSM and Windows Events:

Specifies the time at which the event was generated.

&JES3NAME

Specifies the system name of a JES3 GLOBAL or sysplex console (parsed from the JES3 or sysplex message stream).

&JOBID

Specifies the current JES job ID associated with the message or command. This variable is available for session types MCS, RCS, and VM.

CA NSM Event:

Specifies the TCP/IP address of the device that sent an SNMP trap.

Windows Event:

Specifies the Windows event log ID.

&JOBNAME

Specifies the job name of the address space that issued the message or command. This variable is available only for JES3 sessions and certain types of VM messages.

CA NSM Event:

Specifies the value from the CA NSM Process field; this field normally contains the value "*ProcessID,ExecutableName*".

Windows Event:

Specifies the name of the source (application, service, driver, subsystem) that generated the event.

&JULDATE

Specifies the current date for the workstation in Julian format: *yyddd*, where *ddd* can be 001 to 366.

&LDATE

Specifies the current system date for the workstation in the form *mm/dd/yyyy*.

&LJULDATE

Specifies the current date for the workstation in Julian format: *yyyyddd*, where *ddd* can be 001 to 366.

&MONNAME

Specifies the source system name for a DataFrame message.

CA NSM and Windows Events:

The name of the host that generated the event.

CA OPS/MVSCA OPS/MVS:

Specifies the name of the CA OPS/MVS Multi-System Facility (MSF) that generated the message.

&MONNUM

Specifies the source system number for a DataFrame message.

CA NSM Event:

Specifies the name of the CA NSM Event Manager host from whose log the event was extracted.

Windows Event:

Specifies the name of the Windows host from whose logs the event was extracted.

CA OPS/MVS:

Specifies the system ID of the Event Notification Facility (ENF) that issued the WTO.

&MONPRTY

Specifies the priority of the alert message issued by the DataFrame system.

CA NSM Event:

Error, Warning, Info, Success, or Failure.

Windows Event:

Error (error event), Warning (warning event), Info (information event), Success (successful audit event), or Failure (failure audit event).

CA OPS/MVS:

If specified, the console name indicated on the CA-OPS/MVS ADDRESS WTO command.

&MONTYPE

Specifies the type of alert message issued by the DataFrame system.

CA NSM Event:

Twenty bytes from the CA NSM Facility field are combined with twenty bytes from the Category field, resulting in a value of "*Facility,Category*".

Windows Event:

Specifies the name of the Windows event log (System, Application, or Security) from which the event was extracted.

&MSG

Specifies the first line of the message text.

&MSGTYPE

Specifies the type of message, either NORMAL or ACTION.

&PREMSG

Valid in asynchronous sessions only, this keyword contains the text of the message prior to the position specified in the Begin Message fields in the session definition. (See the Configuration Manager HTML help for more on these fields.)

&REPLYID

Specifies the reply number for a message that is a WTOR (write-to-operator with reply). The value is null if the message is not a WTOR.

&SESSION

Specifies the name of the session from which the message or command was issued. If the message was issued by CA Automation Point itself, the session name will be AXC. If the message was issued by the Notification Server, the session name will be VOX. If the message was received through the CA-OPS/MVS Interface, the session name will be OPS.

&SYSNAME

Specifies the default system name specified for the session.

&TIME

Specifies the current time for the workstation in the form *hh:mm:ss*.

&WORD n

Specifies each word in the message or command text, where n is a number from 1 to 85. (&WORD0 is not a valid variable name.) For example, &WORD1 is the first word in the message text and &WORD50 is the fiftieth word in message text. Only the first 85 words of a message or command are available to CA Automation Point. A message or command containing ten words causes CA Automation Point to store a null value in variables &WORD11 through &WORD85.

The value stored in the &WORD n variable can be as long as the entire message or command. When parsing message or command text into words, CA Automation Point recognizes blank spaces or commas as delimiters between words. If the message or command contains no blanks or commas, the &WORD1 variable stores the whole message or command as a single word.

WARNING! Spaces in session names and/or file names will affect word count and position. You must account for this when you write rules against CA Automation Point messages or commands.

&USER

Valid for CMDIN rules only, this keyword contains the logon information in the form of *logon_name@logon_node* of the user who issued the command remotely. It is blank for local users using the CA Automation Point desktop.

&XC_VER

Specifies CA Automation Point version information.

Status Variables

Status variables are symbolic variables useful for building a body of operations data. These variables allow multiple rules or sets of rules to share the same data and let CA Automation Point communicate with REXX procedures. Any rules keyword that uses environmental variables can also use status variables. You can use the data to:

- Pass information between rules
- Pass information between CA Automation Point and REXX programs
- Control processing of a subsequent message
- Provide input for automation

Types of Status Variables

Status variables can be either volatile (residing in memory only) or non-volatile (disk-based). For more information about both types of status variables, see the chapter "[Understanding Global Variables](#) (see page 190)."

Creating Status Variables

To create a status variable and assign a name to it, use one of the following methods:

- Use the keyword `SET(&varname=value)` in a rule.

For example, specifying `SET(&STATUS=WAITING)` assigns a value of `WAITING` to the variable `&STATUS`.

- Issue the `SETVAR` command processor from within a REXX program.

Specifying Status Variables Names

When specifying a status variable name in CA Automation Point rules, note the following rules:

- The name cannot contain more than 32 characters.
- Precede a status variable name with an ampersand (&).
- The first character of the variable name (after the ampersand) and all remaining characters must be one of the following: A-Z 0-9 ! ? _
- The name should end with one of the following characters or a space: ' " & . ()
- Use only uppercase alphabetic characters.
- Do not use the name of a CA Automation Point environmental variable.

Assigning Status Variable Values

When assigning status variable values, note the following rules:

- Values must contain printable ASCII characters.
- The length of a value must be from one to 255 characters.

Nonvolatile Status Variables

Nonvolatile status variables are disk-based. CA Automation Point stores nonvolatile variables in a special disk file named `AXC.glv` (which CA Automation Point reads and loads at startup). A nonvolatile status variable is distinguished from a regular (volatile) status variable by a special prefix in the nonvolatile variable's name.

To create a nonvolatile status variable, place the prefix `AXCDISK_` in the variable name. For example, specifying `SET(&AXCDISK_STATUS=WAITING)` assigns a value of `WAITING` to the nonvolatile variable `&AXCDISK_STATUS`.

In addition to the built-in AXCDISK_ prefix, you can define other nonvolatile-variable prefixes.

To define your own nonvolatile variable prefixes

1. From Configuration Manager, go to Expert Interface, Automation, Non-Volatile Status Variables.

The Non-Volatile Status Variables dialog displays.

2. Enter the new prefix name in the Prefix field.
3. Click OK.

Using Dynamic Status Variable Names

In rules, you can create status variable names from other variables and from constants. Therefore, you can dynamically create a status variable name so that a single *logical* status variable can reference or set several related status variables.

A dynamic status variable name can have these components:

- Other status variables
- Environmental variables (in message rules only)
- Unquoted text

Note: Always enclose the components of the dynamic status variable in parentheses. At execution time, CA Automation Point resolves all variables to form the actual variable name.

Example:

Suppose that your site sometimes has problems locating output tape volumes that are not in the tape library, because they are still mounted on a drive, sitting on top of a drive, or sitting on a cart to be returned to the tape library. The tape drive where the tape volume was last mounted is a good place to begin looking for the tape. z/OS issues the following message each time you dismount a tape volume:

```
IEF234E D 480,TEST1,,DSIPS11H,TAPEINIT
```

The following message rule uses a dynamic status variable to store the device address, date, and time for each tape volume as you dismount it:

```
MSGID(IEF234E), WHEN(&WORD4 NE ''),  
  SET(&(DVOL_&WORD4)=&WORD3.&DATE.&TIME)
```

Using the example message IEF234E, the dynamic status variable name &(DVOL_&WORD4) becomes &DVOL_TEST1. Assuming that the date is 03/23/00 and the time is 10:55:46, CA Automation Point sets this status variable to the value 48003/23/0010:55:46.

When you dismount the next tape volume, the message rule executes again, and saves the new device address, date, and time in a status variable containing the name of the current tape volume. In this way, the same SET clause sets values to different status variables.

Updating Status Variables

You can update status variables through rules or REXX programs.

From rules, use the SET keyword in a clause with this format:

```
SET(&statvarname=value)
```

&statvarname

Specifies the name of the status variable that you are updating, and *value* is a literal value or the name of another variable.

From REXX programs, remember the following:

- Issue the SETVAR command processor, or
- Use REXX's ADDRESS statement to access the CA Automation Point GLV command environment. From there, your REXX programs can issue global variable-control commands.

These methods are useful for setting status variables in REXX programs running externally (outside of CA Automation Point). See the *Command and Keyword Reference Guide* for more information on these methods.

Referencing Portions of Variables

You can use *substrings* to reference portions of a status or environmental variable. To specify a substring, use this format:

```
varname(startcol:endcol)
```

Example 1:

If the variable &TIMESTAMP contains the value 00:12:31:23:59:59, then &TIMESTAMP(4:8) equals 12:31.

Example 2:

The following SET statement sets the value of a status variable by using substrings of environmental variables. Suppose that the variable &DATE contains the value 12/31/00 and the variable &TIME contains the value 23:59:59. The following &TIMESTAMP variable is set as follows:

The following SET statement:

```
SET(&TIMESTAMP=&DATE(7:8):&DATE(1:2):&DATE(4:5):&TIME)
```

Produces this variable setting:

```
&TIMESTAMP=00:12:31:23:59:59
```

Referencing Parts of Dynamic Status Variables

You can also use substrings to reference parts of a dynamic status variable.

Example:

If the variable &WORD4 has the value OPEN and the variable &FILE_OPEN has the value 480, then &(FILE_&WORD4)(1:1)=4.

This example uses environmental and dynamic status variables in a rule that sets the clock of a mainframe computer.

During IPL, z/OS issues this message:

```
IEA888A LOCAL DATE=2000.173,CLOCK=09.35.06 REPLY U, OR GMT/LOCAL TIME
```

To set the mainframe clock, you can write this rule:

```
MSGID(IEA888A LOCAL), WHEN(&(TIMESET_&SESSION) EQ YES),  
  OSCMD(R 00, U)
```

```
MSGID(IEA888A LOCAL), WHEN(&(TIMESET_&SESSION) NE YES),  
  SET(&(TIMESET_&SESSION)=YES),  
  OSCMD(R 00, CLOCK='&TIME(1:2).&TIME(4:5).&TIME(7:8),  
    DATE=&LJULDATE(1:4).&LJULDATE(5:7)')
```

If the value of the &(TIMESET_&SESSION) status variable is not YES, the rule sets the variable to the current date and time according to the values in the &JULDATE and &TIME environmental variables.

Specifying Text Strings in Rules

Many rules keywords have free-form text strings associated with them. The text strings can be one or more of the following:

- Character strings
- Status variables
- Environmental variables that have values to be substituted into the text string
- Key operations such as @"CLEAR" or key abbreviations such as @C (supported by the SESSCMD rules keyword)

Specifying Character Strings

You can specify character strings with or without quotes. For example, these keyword phrases are equivalent:

```
OSCMD(S DEALLOC)
OSCMD('S DEALLOC')
```

Both quoted and unquoted strings can contain the names of environmental variables. The current value of the variable replaces the name within the string. For example, if the current job is DSIJS11A, CA Automation Point makes this substitution in a SESSCMD statement: SESSCMD(CANCEL &JOBNAME)

Substitution:

```
SESSCMD(CANCEL DSIJS11A)
```

If a normal delimiter such as a blank, comma, right parenthesis, or quote does not follow a variable name, terminate the variable name with a period (.) to distinguish it from the characters that follow.

Example:

This example uses a period to delimit a variable name. &JESCHAR is a status variable containing the JES2 command character (such as \$).

The statement SESSCMD(&JESCHAR.SI4) becomes:

```
SESSCMD($SI4)
```

You can use concatenation in any CA Automation Point rules keyword supporting text strings, except the WHEN keyword.

Enclosing Character Strings in Quotes

You must enclose a character string in quotes in these cases:

- When leading or trailing blanks are significant, as shown in this example:

```
WHEN(&JOBNAME EQ 'CICS  ')
```

- When the ampersand character displays in a string and it may be mistaken for the beginning of a variable name, as in the following example. This example also shows how to concatenate quoted and unquoted strings by placing them side by side in a text string.

```
WTO(NOT AN ENVIRONMENTAL '&' VARIABLE)
```

- When the string itself contains a quote or when the string has unbalanced parentheses, as shown in this example:

```
WTO('TEXT WITH MANY PARENS))))')
```

- Within a quoted string, you specify the quote character by writing two adjacent single quotes, as shown in these examples:

```
WTO('TEXT WITH ''QUOTES'' IN IT')  
OSCMD('SEND ''IMS HAS ABENDED'',BRDCST')
```

Note: If you place two adjacent single quotes *outside* a quoted string, the two quotes evaluate to null, as shown in the following example:

The phrase `OSCMD(SEND "IMS HAS ABENDED",BRDCST)` produces:

```
SEND IMS HAS ABENDED,BRDCST
```

Writing Time Rules

A time rule begins with the TIME keyword. Time rules can also contain any of the following rules keywords:

ALARM	SCRIPT
ALARMSAY	SESSCMD
DOSCMD	SET
EVERY	SNMPTRAP
EXPORTMSG	UNICMD
LIMIT	UNIWTO
MATCHLIM	WHEN
OSCMD	WTO

PPQWRITE	WTXC
REXX	XCCMD

Evaluating Time Rules

Use the Evaluation Frequency field in the Rules Settings dialog in Configuration Manager to determine how frequently CA Automation Point evaluates time rules. You can use these environmental variables in time rules: &DATE, &DAY, &JULDATE, and &TIME.

To execute a rule multiple times during the day, add a clause containing the EVERY keyword to that rule. If a time rule does not contain an EVERY clause, it executes only at the time specified on the TIME keyword.

Example:

Suppose that a time rule begins with the clause TIME(08:00). Because CA Automation Point uses a 24-hour clock to process time rules, this rule executes only once a day, at 8:00 a.m.

If you want the rule to execute at both 8:00 a.m. and 8:00 p.m., rewrite it as follows:

```
TIME(08:00), EVERY(12 HOURS) ruletext
```

If you want the rule to execute every four hours, rewrite it as follows:

```
TIME(08:00), EVERY(4 HOURS) ruletext
```

The revised rule executes for the first time at 8:00 a.m., then at 12 noon, 4:00 p.m., 8:00 p.m., and midnight. After midnight, the rule does not execute again until 8:00 a.m. (the time specified with the TIME keyword).

If CA Automation Point starts operating before the time when a rule is scheduled to execute, the rule executes at the specified time or time interval. If CA Automation Point starts up *after* the time when a rule is scheduled to execute, that rule does not execute unless it contains an EVERY clause. In such a case, CA Automation Point processes the rule as follows:

- CA Automation Point determines when the rule would have executed last, then continues executing the rule at the interval specified on the EVERY clause.
- After midnight, CA Automation Point stops executing the rule until the time specified on the TIME clause.

Evaluating Time Rules During Time Changes

This section describes, in detail, how CA Automation Point processes time rules during time changes.

CA Automation Point does not use dates when it evaluates time rules. The product does, however, support the concept of a *day* when it determines at what time the rules should be reset.

Time rules work on a 24-hour clock from 00:00 to 23:59. A *day* begins at 00:00. To determine when it should execute time rules, CA Automation Point internally checks the clock according to the interval specified in the Evaluation Frequency field in the Rules Settings dialog of Configuration Manager. See the dialog HTML help for more information about this field. The default interval is 60 seconds.

When the Evaluation Frequency interval expires, CA Automation Point checks the clock to see which rules should execute based on the current time and the time prior to the interval. Normally, all rules in this range execute. However, if the current time has a value that is less than the prior time—which, if the interval's value is 60 seconds, would normally occur between the interval starting at 23:59 and ending at 00:00—CA Automation Point resets the rules, bypassing any rules that have a time that is later than the start of the interval (for example, 23:59:59). Nevertheless, all rules between 00:00 and the end of the interval execute (for example, if the interval that crosses midnight moves from 23:59:30 to 00:00:30, a rule for 00:00 or 00:00:20 executes).

The reset process involves moving the NEXT time rule to fire pointer to the top of the list; this stops the processing of the EVERY keyword for the day that is ending (for example, the clause TIME(15:00) EVERY(1 HOUR) executes once an hour beginning at 3 p.m. and ending at midnight when the reset is completed).

Daylight Savings Time

The CA Automation Point Enable Clock Change field in the Rules Settings dialog determines whether previously executed time rules reexecute after a *backward* time change.

In CA Automation Point, time rule logic becomes complex when a workstation's clock is changed during a normal Evaluation Frequency interval, such as the one-hour fall back change that occurs when switching the clock from daylight savings time to standard time, or the one-hour move ahead change that occurs when switching from standard time to daylight savings time. The move ahead change works in the same manner as if you had set the Evaluation Frequency interval to one hour—all rules in the one-hour range execute, with the same exception that was described previously when midnight is crossed. In this instance, only the rules from midnight to the end of the interval execute.

By default, CA Automation Point interprets a fall back change to actually be a move *forward* by 23 hours. When 02:00 is set back to 01:00 during an Evaluation Frequency interval, CA Automation Point views the time as moving forward 23 hours, across midnight. In this example, all rules from 02:00 to 23:59 are bypassed (not executed) and all rules from 00:00 to 01:00 are executed. As the time moves forward, all rules continue to execute normally, including those that executed the first time the clock advanced from 01:00 to 02:00. (Remember, because of the fall back change, the clock advances from 01:00 to 02:00 *twice* that day.) While CA Automation Point internally views this as a move forward by 23 hours, you will see that the rules between 00:00 and 02:00 executed twice (or reexecuted) on that day; because CA Automation Point does not associate a date with rules logic.

The primary purpose of enabling clock change is to allow you to determine how you want to handle fall back changes. Additionally, you can use this setting to address any backward time resets of up to two hours.

When the Enable Clock Change box is unchecked *and* the time is set back two hours or less, any previously executed rules within the time range from the backward reset time to the time the clock was prior to being set back *will not* reexecute. The rules resume normal execution when the time reaches the exact time at which the clock was set back. For example, if the clock is set back from 02:00 to 01:00, the rules do not resume executing until 02:00 is reached again. The rules behave this way even when the set back occurs across midnight, as long as the time difference is two hours or less. For example, if the clock is set back from 00:30 to 23:30 (which, when the Enable Clock Change box is unchecked, CA Automation Point interprets as a one-hour backward reset), no rules execute until the time reaches 00:30.

When the Enable Clock Change box is checked *or* the time is set back more than two hours, CA Automation Point assumes that the time moved *forward*. This logic works in the same manner described previously. If the move forward does not cross midnight, all rules in the range execute. For example, if the clock is set back from 02:00 to 23:00, CA Automation Point assumes that the clock moved forward 21 hours and that all rules from 02:00 to 23:00 executed. If the move forward crosses midnight, only the rules from midnight to the end of the interval execute. If the clock is changed from 06:00 back to 03:00, CA Automation Point assumes that the clock moved forward and crossed midnight. The rules from 06:00 to 23:59 are bypassed (not executed), while the rules from 00:00 to 03:00 are executed.

Pictorial Time Rule Examples

This section contains illustrations that will help you to better understand how CA Automation Point processes time rules.

Note: All clocks are 24-hour clocks. The times noted on the outside of the clocks indicate when time rules are scheduled to execute.

Scenario 1: The time is set back from 02:00 to 01:00.



If Enable Clock Change is checked:

- CA Automation Point assumes that the time moved ahead 23 hours—from 02:00 to 01:00 the next day—and that the move crossed midnight. Therefore, all rules reset at 24:00.
- All rules defined from 24:00 to 01:00 execute immediately. Therefore, the 00:15 rule executes immediately.
- Rules processing resumes normally at 01:00. Therefore,
 - The 01:30 rule executes at 01:30 the same day.
 - The 23:45 rule executes at 23:45 the same day.

If Enable Clock Change is not checked:

- Any rules defined within the shaded area do not execute. Therefore, the 01:30 rule does not execute.
- Rules processing resumes normally at 2:00 the same day. Therefore,
 - The 23:45 rule executes at 23:45 the same day.
 - The 01:15 rule executes at 00:15 the following day.

Scenario 2: The time is set ahead from 01:00 to 02:00.



Enabling Clock Change does not affect forward moves.

- All rules defined within the shaded area execute immediately. Therefore, the 01:30 rule executes immediately.
- Rules processing resumes normally at 2:00. Therefore, the 4:00 rule executes at 04:00 the same day.

Scenario 3: The time is set back from 00:30 to 23:30.



If Enable Clock Change is checked:

- CA Automation Point assumes that the time moved ahead 23 hours, and that the move did not cross midnight. Therefore, the rules do not reset at 24:00.
- All rules defined outside the shaded area execute immediately. Therefore, the 02:00 rule executes immediately.
- Rules processing resumes normally at 23:30. Therefore,
 - The 23:45 rule executes at 23:45 the same day.
 - The 00:15 rule executes at 00:15 the following day.

If Enable Clock Change is not checked:

- Any rules defined within the shaded area do not execute. Therefore,
 - The 23:45 rule does not execute.
 - The 00:15 rule does not execute.

Rules processing resumes normally at 00:30. Therefore, the 02:00 rule executes at 02:00 the same day.

Scenario 4: The time is set back from 06:00 to 03:00.



If Enable Clock Change is checked:

- CA Automation Point assumes that the time moved *ahead* from 06:00 to 03:00 the next day and that the move crossed midnight. Therefore, all rules reset at 24:00.
- All rules defined from 24:00 to 03:00 execute immediately. Therefore, the 02:00 rule executes immediately.
- Rules processing resumes normally at 03:00. Therefore,
 - The 05:00 rule executes at 05:00 the same day.
 - The 23:00 rule executes at 23:00 the same day.

If Enable Clock Change is not checked:

- Because the time moved back is more than two hours, the results are the same as when Enable Clock Change is checked.

Writing Message Rules

In message rules, include the MSGID keyword and some text identifying the message or group of messages that causes the rule to execute; the text, which causes the rule to execute, is usually the message number. For example, a rule beginning with MSGID(\$HASP395) executes when CA Automation Point finds a message that begins with \$HASP395.

When writing rules to process messages from the VAX or VAXALL type of asynchronous consoles, you can specify the MSGID keyword and part of the message text. (For VAX messages, *do not include* the OPCOM header as part of the message text that CA Automation Point looks for.) For example, suppose that you want to write a rule that executes when CA Automation Point sees this message:

```
%%%%%%%% OPCOM 18-JUL-94 16:26:07.89 %%%%%%%%%  
Message from user DECNET...(message_text)
```

In your rule, specify Message as the text that CA Automation Point should look for. Your rule might look like this:

```
MSGID(Message), WHEN(WORD4 EQ DECNET), OSCMD(CANCEL)
```

The rule executes when CA Automation Point finds a message whose first word is Message and whose fourth word is DECNET. When it sees such a message, CA Automation Point sends the write-to-operator message CANCEL to the console.

CA Automation Point also processes its own messages through the CA Automation Point rules processor, allowing you to write rules to respond to those messages. To restrict a rule to only these internal product messages, use the SESSION keyword and specify the session name of AXC.

Writing Command Rules

A command rule begins with the CMDIN keyword followed by characters that identify a specific command or group of commands that cause the rule to execute. For example, a rule beginning with CMDIN(DRL) executes when CA Automation Point determines that a command starting with DRL is issued through CA Automation Point.

Command rules can also contain any of these rules keywords:

ALARM	REXX
ALARMSAY	SCRIPT
DOM	SESSCMD
DOSCMD	SESSION

EVERY	SET
LIMIT	SUPPRESS
MATCHLIM	WHEN
OSCMD	WTO
PPQWRITE	WTXC
REWORD	XCCMD

Command rules can be used when you issue a command from the red command area or from the Command dialog using the Automation Point Desktop, the Remote Viewer, or the Web Viewer.

Command rules do not execute if you enter the command directly in the session window. CA Automation Point checks the command rules, as well as the time rules, regardless of whether the session is configured as an automated session.

Note: You define a session to be automated by checking the Automate Session checkbox in the session definition.

In many instances, command rules and message rules are implemented similarly. However, two keywords, SUPPRESS and REWORD, have slightly different meanings in command rules than they do in message rules. See the descriptions of these keywords in the *Command and Keyword Reference Guide*.

Command rules let you suppress a command or allow operators to enter a pseudo-command from the command line and reword it as another command, without going into REXX mode. The command rule can also intercept the command and invoke a REXX routine or take other action as required.

Examples

- The following rule prevents an unauthorized remote operator from using an MVS STOP command to remotely stop CA OPS/MVS, and issues a write-to-operator message to the CA Automation Point message console.

```
CMDIN(STOP) WHEN(&WORD2 EQ 'OPSS' AND &USER NE '' AND &USER(1:9) NE 'SYSADMIN@')
SUPPRESS WTXC(&USER is not authorized to remotely stop CA-OPS/MVS)
```

- The following rule enables an operator to get a complete list of the outstanding replies using the DRL pseudo-command, which is converted to the D R, L, CN=(ALL) command.

```
CMDIN(DRL), REWORD(D R, L, CN=(ALL))
```

- The following rule rewords a complex MVS command so that the operator can issue the pseudo command 'SLIPOC1' instead of the complete MVS command.

```
CMDIN(SLIP0C1), REWORD(SLIP SET, ID=P0C1, COMP=0C1, A=SVCD, MATCHLIM=1,
JOBNAME=TESTPGR, END)
```

- When an operator issues the PAGE Bill command through CA Automation Point, this rule executes to intercept the command. It keeps (suppresses) the PAGE pseudo-command from being issued to the host, invokes the page.rex program to page Bill on demand, and issues a write-to-operator message to the CA Automation Point message console.

```
CMDIN(PAGE), SUPPRESS, REXX(page.rex &WORD2), WTXC(Requested paging &WORD2)
```

- The following rule enables an operator to get a complete list of the outstanding replies using the DRL pseudo-command, which is converted to the D R, L, CN=(ALL) command. rule tracks the date and time the START command is issued through CA Automation Point.

```
CMDIN(START), SET (&LAST_START = &TIME &DATE), WTXC(Most Recent START issued at &LAST_START)
```

Enabling a Rules File

This section describes how to activate a rules file—such as the sample rules file ACXRULES.rul provided with CA Automation Point—and how to replace the current rules file with another while CA Automation Point is running.

To activate the rules file that you want CA Automation Point to use at startup, specify its name in the Rules Settings dialog in the session definition set. The rules file that you specify takes effect each time you start CA Automation Point.

Editing a Rules File

To locate and edit a rules file using Configuration Manager

1. From Configuration Manager, choose Expert Interface, Automation, Rules, Edit Rule.
2. Select the rules file you want to edit.

Replacing the Current Rules File Dynamically

There are two ways to replace the current rules file while CA Automation Point is running:

- Issue the LOADRULES command processor.

CA Automation Point does not activate a new rules file immediately. Instead, CA Automation Point temporarily blocks processing of new incoming messages (that is, the messages that appear immediately after you issue the command to activate the new rules file).

CA Automation Point waits until all active rules in the current rules file have finished processing. It then disables the current rules file and activates the new rules file. CA Automation Point then releases the new incoming messages—those that were temporarily blocked—so that the newly activated rules can process them.

- Follow these instructions:
 1. From Configuration Manager, choose Expert Interface, Automation, Rules, Reload Rules.
 2. Select the rules file you wish to reload.

Important! Keep the following information in mind when activating a rules file:

- CA Automation Point activates the rules file that you specify in the Rules Settings dialog in Configuration Manager each time you start the product, *even if you enable a new rules file while it is running*.
- By default, CA Automation Point loads a new rules file even if one or more rules in the file contain syntax errors.
- If CA Automation Point cannot find the new rules file that you specify or if it does not exist, the current rules file remains in effect.

Queued REXX programs initiated by rules in the previous rules file may continue to generate messages even after you activate the new rules file.

Chapter 8: Configuring and Writing REXX

This section contains the following topics:

[Using REXX Programs for Automation](#) (see page 181)

[Configuring Open Object REXX](#) (see page 183)

[Invoking REXX Programs](#) (see page 183)

[Accessing Various Command Environments Through REXX](#) (see page 184)

[How to Issue Command Processors](#) (see page 186)

[Designing Portable REXX Programs](#) (see page 187)

[How Do CA Automation Point and REXX Communicate?](#) (see page 188)

[Understanding REXX Statement Processing](#) (see page 189)

[Understanding Global Variables](#) (see page 190)

[Understanding the Global Variable Environment](#) (see page 190)

[Addressing the Global Variable Environment](#) (see page 194)

[Testing REXX Programs](#) (see page 196)

[Binding a REXX Program to a CA Automation Point Session](#) (see page 198)

Using REXX Programs for Automation

To automate simple tasks such as managing messages, you can define short instructions using the CA Automation Point rules language. For more complex system operations tasks or tasks that require you to perform a series of procedures, you can use REXX programs for automation.

Note: REXX programs performing automated tasks should not contain the PULL statement, because this requires user interaction. You should invoke non-automated programs of this type from a Command Prompt. For more information, see the section [Invoking REXX Programs](#) (see page 183).

With REXX programs, you can control complex tasks such as:

- Communicating with networked workstations using PPQs
- Manipulating and evaluating data stored in status and other variables
- Managing interactions between incompatible operating systems such as z/OS and Linux.
- Manipulating mainframe status and configuration data stored as files on the workstation
- Interacting with a voice card in your workstation

For example, suppose that you want to respond to a host message by issuing a display command, comparing the display output with the original message text, and then responding to the original message. Managing such a complicated response is beyond the scope of rules, but you can do it through a REXX program that issues the CA Automation Point SESSCMD and GETSCRN commands.

Note: CA Automation Point cannot accept requests to run REXX programs unless you have checked the Enable AXCREXX check box on the REXX Settings dialog.

Supplied Applications

CA supplies several automation applications with CA Automation Point. The applications include:

- Example applications that illustrate the CA Automation Point advanced features
The example applications may not provide the precise automation solutions appropriate for your site, but they show how all the parts of a working application are coded. Use them as a starting point for building more complex applications.
- Complete solutions for common automation problems
After installing the applications and performing a few site-specific customization tasks, they are ready to run.
- Building blocks for common elements of larger applications
Building block applications are prerequisites for running other CA-provided applications.

Each application provided with CA Automation Point has a corresponding .txt file that contains information about the application, such as:

- A description of what the application does
- A list of application components (that is, file names and their descriptions)
- Installation and customization instructions
- Diagnostic hints and tips
- A logical overview of the application's internal operations
- Other relevant information, where necessary

You can find these sample files in the *installDir*\SAMPLE directory.

Configuring Open Object REXX

Note: You need to perform the instructions in this chapter only if you specifically installed Open Object REXX on your machine.

Configuration Manager enables you to use the CA Automation Point address environment from Open Object REXX programs.

Use the REXX Settings dialog, to configure Open Object REXX.

Note: CA Automation Point supports only Open Object REXX version 4.0 or above.

Invoking REXX Programs

You can invoke a REXX program in the following ways:

- From the CA Automation Point command area or from the workstation's operating system prompt

From the command prompt, specify the REXX program and any optional arguments as follows:

```
asorex rexxprog arguments
```

Note: If necessary, you can explicitly specify the directory where the REXX program resides. For information about CA Automation Point file directory structure, see the chapter "[Configuring CA Automation Point](#). (see page 19)"

With a REXX program, you can automate a long, command-intensive operator task, such as logically partitioning a mainframe processor and IPLing the partitions. Instead of manually issuing a series of CA Automation Point commands, system commands, or other types of commands, you can save time by manually invoking a REXX program to issue all of the necessary commands automatically.

- From a CA Automation Point rule

When a task is too complex to be automated exclusively with CA Automation Point rules, you can use rule-initiated REXX programs to perform automation. You may want to invoke REXX programs in this way when:

- Making the proper response to a message is too complicated
- Rules cannot provide all of the information needed
- The REXX program must execute at a particular time or after a specified time interval

- From another REXX program
The CA Automation Point REXX command processor allows you to invoke one or more REXX programs from within a single REXX program. All REXX programs queued by the original program can run concurrently while the original program continues to execute.
- By selecting REXX program(s) for CA Automation Point to run at startup
You can select these programs in Configuration Manager from the REXX Settings dialog.

Accessing Various Command Environments Through REXX

Besides a standard set of built-in functions, the REXX programming language allows access to other *command environments*. A command environment is a group of non-REXX commands that your REXX programs can issue through an ADDRESS statement. These command environments provide additional functions appropriate for each environment that eliminates the need to extend or modify the basic REXX language definition for each unique implementation.

Command Environments

The following table shows a partial list of command environments that may be available at your site:

Environment	Description
AXC (Default)	CA Automation Point command processors. The AXC environment is the default environment for REXX programs invoked from <i>within</i> CA Automation Point (using the AXCREXX program). For more information, see How to Issue Command Processors (see page 186) in this chapter.
PPQ	Program-to-program queue (PPQ) commands
TNG	The TNG ADDRESS environment provides a REXX-based programming interface to the objects contained in the CA NSM WorldView Repository
GLV	Global variable control commands
VOX	An environment that provides you with the ability to create an executable statement that performs a specific voice processing or voice-related action
OPS	An environment that allows you to communicate with CA OPS/MVS

Note: Product-specific commands may be necessary to activate and deactivate additional environments.

Using the REXX ADDRESS Statement

REXX programs can access other command environments through an ADDRESS statement. For example, this REXX statement accesses the CA Automation Point PPQ environment and allows you to issue PPQ commands:

```
ADDRESS PPQ "command [operands]"
```

When multiple command environments are active, the REXX ADDRESS statement directs commands to a specific environment. If an environment is not active, commands provided under that environment are not available. Attempting to address an invalid or inactive command environment generates a return code of -30, which is returned in the special REXX variable RC.

The CA Automation Point AXCREXX program activates the AXC environment and makes it the *default environment*. The AXC environment allows you to issue CA Automation Point command processors.

When the AXCREXX program executes, it generates these return codes:

Code	Description
0	AXCREXX started the specified REXX program successfully.
200	AXCREXX could not find the REXX program file that you specified.
201	You did not specify a REXX program file name.
202	AXCREXX could not register its CA Automation Point command processor with REXX.
203	AXCREXX could not detect the presence of CA Automation Point.
204	No more return queues are available from which AXCREXX can receive results of CA Automation Point REXX commands.

How to Issue Command Processors

You can issue CA Automation Point command processors from:

- REXX programs
- CA Automation Point rules
- A CA Automation Point menu

Command processors that you issue from REXX programs execute through REXX's external command interface. Use the XCCMD rules or menu keyword to issue command processors from CA Automation Point rules or menus.

The following sections describe each method for issuing CA Automation Point command processors.

Issuing Command Processors from REXX Programs

The CA Automation PointAXCREXX program activates the AXC environment and makes it the default environment. The AXC environment allows you to issue CA Automation Point command processors from your REXX programs.

All REXX programs invoked from *within* CA Automation Point—for example, from rules or from a CA Automation Point command line—run under AXCREXX (and, therefore, access the AXC environment by default):

- To issue a CA Automation Point command processor from a REXX program invoked within CA Automation Point, place the command processor and its associated operands on a separate statement line and enclose them in double quotation marks, as follows:

```
"command operands"
```

For example:

```
"SETVAR REXX_MESSAGE (REXX SENT THIS MSG)"
```

- To access the AXC environment and issue a CA Automation Point command processor in any other way (for example, from a visual REXX environment), use REXX's ADDRESS statement as shown:

```
ADDRESS AXC "command operands"
```

For example:

```
ADDRESS AXC "SETVAR REXX_MESSAGE (REXX SENT THIS MSG)"
```

Issuing Command Processors From Rules

To issue a CA Automation Point command processor from a rule, use equivalent rules keywords such as SESSCMD, REXX, SCRIPT, WTO, and WTXC. If an equivalent rules keyword is not available, use the XCCMD rule keyword in a rules clause; for example:

```
XCCMD(cmdtext)
```

Note: *cmdtext* is the text of any CA Automation Point command processor, its operands, and the operands' values.

Issuing Command Processors From Menus

You can issue CA Automation Point command processors by selecting a menu item from the menus. For more information about adding menu items to existing menus, see the chapter "[Managing Sessions Using CA Automation Point Windows](#). (see page 75)"

The following example shows sample menu control statements that you can add to your USER.MNU file to invoke the SCRIPT command processor:

```
SUBMENU=UserApps, NAME=(~UserApps, help_window),  
ITEM=('~My Script', XCCMD('SCRIPT MYSCRIPT.SCR SESSION(HELPPDESK)'))
```

Designing Portable REXX Programs

The REXX programming language, with its standard set of built-in functions, allows you to migrate your REXX programs between implementations. For example, you can migrate REXX programs between CA Automation Point and CA OPS/MVS.

When designing REXX programs for portability, keep the following points in mind:

- Command processors are generally portable between the workstation and CA OPS/MVS.

Note: Command processors under different platforms may have additional platform-specific parameters and unique return codes.

- Invoking host commands in a REXX program can cause that program to lose portability between environments.

REXX provides functions for determining the current host environment so that a REXX program designed for portability can trap environment-specific commands. For more information, see the section [How Do CA Automation Point and REXX Communicate?](#) (see page 188) in this chapter .

- Some characters are not available on both ASCII (workstation) and EBCDIC (host) platforms.

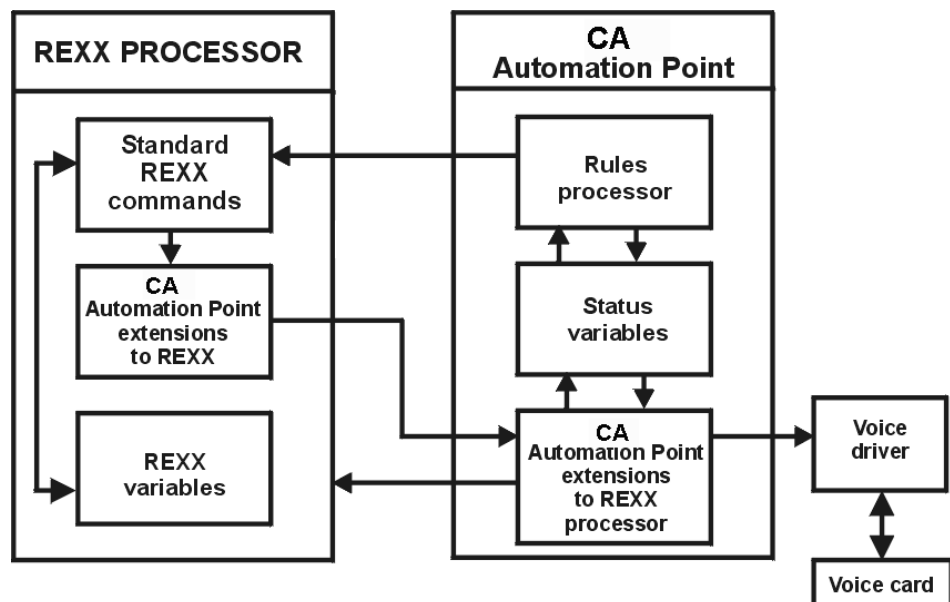
For example, the "`≠`" (not equal) operator works only on an EBCDIC platform; however, "`<>`" (also a not equal operator) is valid on both ASCII and EBCDIC platforms.

How Do CA Automation Point and REXX Communicate?

When CA Automation Point invokes a REXX program:

- The REXX program issues standard REXX commands and commands available in other environments.
- The commands interact with internal code in CA Automation Point; the code performs the low-level work required by each specific REXX function, and then returns control of the executing program to REXX.
- CA Automation Point passes information back to REXX in the form of REXX variables, or puts the information on the external data queue.
- The standard REXX commands in the REXX program process the new REXX variables.

The following illustration shows how information flows between CA Automation Point and REXX.



Understanding REXX Statement Processing

REXX processes statements as follows:

- REXX treats every word in a statement as a REXX variable. The default value for each variable is the variable name itself.
- REXX calls its own built-in functions if a statement requires them, and places the values returned from the functions into variables.
- REXX processes each statement arithmetically, removing operators such as *, /, +, and commas.
- If REXX does not recognize the resulting statement, it passes the statement to the current environment as a host command.
- REXX recognizes variables that the AXC ADDRESS environment must resolve when the variables appear within a string surrounded by quotation marks.

Specify variables for the AXC ADDRESS environment to resolve by prefixing them with an ampersand character (&). For example, the following statement places the value of the REXX variable MSG_CONTENTS into the CA Automation Point variable &MSG_TEXT:

```
"SETVAR MSG_TEXT (&MSG_CONTENTS)"
```

When specifying REXX variables as part of a statement, you can also use double quotation marks to offset the variables. For example, to substitute 'THIS IS A MESSAGE FROM REXX' for MSG_CONTENTS, use this syntax:

```
MSG_CONTENTS = 'THIS IS A MESSAGE FROM REXX'
"SETVAR REXX_MESSAGE ("MSG_CONTENTS ")"
```

When REXX encounters this statement, it substitutes only MSG_CONTENTS and preserves the parentheses.

Referencing a Status Variable From REXX

To reference a CA Automation Point status variable from a REXX program, you can either:

- Invoke the GETVAR command processor to copy the value of the CA Automation Point variable into a REXX variable.
- Access the CA Automation Point global variable (GLV) command environment using an ADDRESS GLV statement.

From the GLV environment, select the AXC group, the CA Automation Point built-in variable group for status variables. Each variable group has a unique group name, allowing GLV commands to act upon groups of variables. See the *Command and Keyword Reference Guide* for further details.

Preventing REXX From Parsing a Statement Incorrectly

To prevent REXX from parsing a statement incorrectly:

- Surround your requests for CA Automation Point command processors (and any other text that should not be parsed) with double quotation marks (").
- If text to be sent to CA Automation Point contains embedded double quotation marks, use single quotation marks (') to enclose that text.

For example, because logical operations that you want to send to the host session are always enclosed in double quotes, enclose the entire command string in single quotes, as shown in this example:

```
'SESSCMD /@"POWER_RESET"/ SESSION(sessname)'
```

Understanding Global Variables

This section discusses the CA Automation Point global variable environment. It covers the following topics:

- Understanding the global variable environment
- Addressing the global variable environment

Understanding the Global Variable Environment

The global variable environment allows your REXX programs to access and modify the values of all variables in the GLV variable pool. The GLV variable pool contains:

- CA Automation Point status variables:
 - Volatile (normal status variables)
 - Nonvolatile (disk based—stored in the AXC.glv file)

For more information about CA Automation Point status variables, see the chapter "[Writing Rules](#). (see page 149)"

- Other GLV variables:
 - Volatile
 - Nonvolatile (disk based—stored in .glv files)

REXX programs use GLV variables by placing their values into local REXX variables having the same names for processing. You can update GLV variables by assigning them the value of the local REXX variable, or you can update them directly.

What are Volatile Variables?

Volatile variables consist of regular CA Automation Point status variables and GLV variables. Through the global variable environment, multiple REXX programs can access volatile GLV variables if the programs are running concurrently.

Volatile GLV variables are transient, rather than permanent; and are deleted when the REXX program that created them terminates.

What are Nonvolatile Variables?

Nonvolatile variables are disk-based. The names of files containing nonvolatile variables have .glv extensions. The following facts apply to nonvolatile variables:

- The CA Automation Point nonvolatile status variables reside in the AXC.GLV file. Those status variables can be identified by a special prefix (such as the built-in AXCDISK_ prefix) or by a user-defined prefix.
- You can specify one or more prefix values. To add or delete user-defined prefixes, go to Configuration Manager, and choose Expert Interface, Automation, and open the Non-Volatile Status Variables dialog.
- CA Automation Point reads the AXC.glv file—and loads its contents into memory—at start-up.
- Nonvolatile GLV variables reside in .glv files, the names of which correspond to variable *group* names. (For more information about variable groups, see [What Is a Variable Group?](#) (see page 192).)
- You can specify one or more prefix values. To add or delete user-defined prefix values

Note: GLV creates backups of the .glv files in .glb files, the names of which correspond to variable group names.

Multiple REXX programs can access nonvolatile variables regardless of when the programs run. Also, nonvolatile variables are not lost even when CA Automation Point terminates or if you switch your workstation's power off.

What Is a Variable Group?

A variable *group* is a user-defined category of related variables. For example, you could group all VAX-related variables into a variable group named VAX_VARS or all variables relating to voice processing into a group named VOX_GRP.

A variable group contains both volatile and non-volatile (disk-based) variables. The .glv files containing non-volatile variables have the same names as their associated variable groups. For example, non-volatile variables in a group named REMGROUP are stored in a file named REMGROUP.glv.

The global variable environment regards variables that have the same name but belong to different groups as separate variables. Such variables can contain different values.

What Is the AXC Variable Group?

The variable group named AXC is the built-in group for CA Automation Point status variables. Non-volatile status variables are stored in the AXC.glv file.

Naming Conventions for Groups

Follow these guidelines when naming a variable group:

- The name can contain up to eight characters.
- These characters are valid:

A-Z 0-9 _ @ \$ #

Naming GLV Variables

Follow these guidelines when naming a GLV variable:

- The name cannot contain more than 32 characters.
- The name must be a valid REXX variable name.
- The following characters are valid: A-Z 0-9 . ! ? _ "
- GLV variable names cannot start with a period (.) or a digit (0 through 9).
- Use only uppercase alphabetic characters. Lowercase characters in a variable name are automatically converted to uppercase. (Lowercase characters do not generate errors.)
- Do not use the name of a CA Automation Point environmental variable in the AXC group.

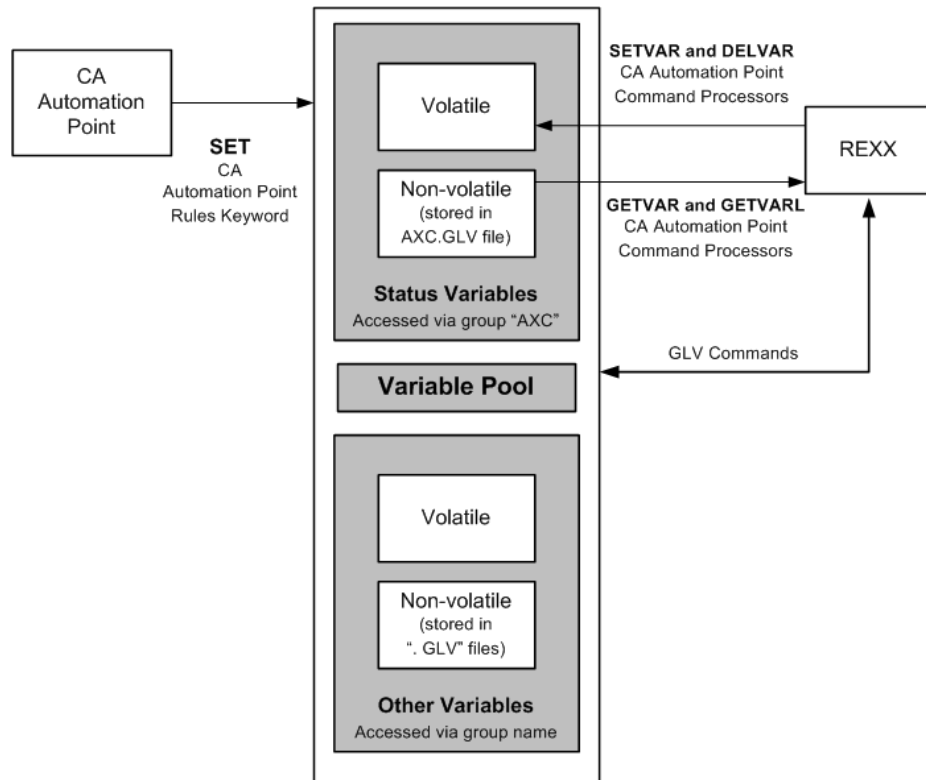
Assigning Values

Follow these guidelines when assigning a value to a GLV variable:

- Values must contain printable ASCII characters.
- The length of a value must be from 1 to 255 characters. When used in a REXX program, the length of a value can be up to 60,000 characters.

Using the Global Variable Environment

The following illustration shows the flexibility that CA Automation Point provides for accessing variables:



You can use the CA Automation Point SET rules keyword and CA Automation Point command processors to control CA Automation Point status variables. However, the global variable environment gives REXX programs global access to *all* variables in the variable pool.

See the *Command and Keyword Reference Guide* for more information on the SET rules keyword and CA Automation Point Command Processors.

Addressing the Global Variable Environment

This section describes how to issue GLV commands from REXX programs.

Setting the GLV Path

You can optionally create a separate directory on your hard disk drive specifically for storing .glv files. After you create this directory, assign the variable GLV_PATH to the directory's path. (From your desktop, click Settings, Control Panel, System, Advanced Tab, Environment Variables, and then Environment.) For example, to store .glv files on drive C in a directory named GLV_VARS, set the variable GLV_PATH to C:\GLV_VARS.

Note: GLV_PATH accepts more than one entry. If you do not create a separate directory for storing non-volatile variables, CA Automation Point stores the .glv files in the %AP_SITE%\Data\GLV directory.

Issuing GLV Commands

Issue a GLV command from within a REXX program by specifying an ADDRESS GLV statement, as shown:

```
ADDRESS GLV "glvcommand operand(s)"
```

GLV Command Summary

The following table lists the GLV commands that you can issue through the ADDRESS GLV statement in a REXX program. See the *Command and Keyword Reference Guide* for command syntax.

GET

Retrieves the current value of a status or GLV variable and assigns that value to a local REXX variable of the same name

GRPLIST

Places a list of all variable groups into the external data queue

GRPLISTV

Places a list of all variable groups directly into a REXX variable

LIST

Places the name and value of a variable into the external data queue

LISTV

Places the name and value of a variable directly into the REXX variable

PURGE

Deletes all volatile variables in a specified variable group

PUT

Assigns the value of a local REXX variable to a volatile status or GLV variable of the same name

PUTP

Assigns the value of a local REXX variable to a non-volatile status or GLV variable of the same name. (This command creates or updates a .glv file.)

SELECT

Selects a particular variable group for succeeding GLV commands to act upon. This command is valid globally as a stand-alone command or locally as an operand in most other GLV commands.

SET

Assigns values to one or more volatile variables

SETL

Assigns a literal value, which can contain blanks, to a single volatile variable

SETLP

Assigns a literal value, which can contain blanks, to a single non-volatile variable. (This command creates or updates a .glv file.)

SETP

Assigns values to one or more non-volatile variables. (This command creates or updates a .glv file.)

VER

Places the version number of GLV services on your workstation to the external data queue

VERV

Places the version number of GLV services on your workstation into a REXX variable

GLV Return Code Variable RCs

After a GLV command executes, it sets the special REXX return code variable RC. The RC variable contains a value of 0 (zero) if the command executed successfully; otherwise, the RC variable contains one of these values:

RC	Description
-86	Unrecognized command name
-84	Unrecognized command operand
-82	Command operand is missing

If the RC variable contains any other value, contact Technical Support at <http://ca.com/support>.

Testing REXX Programs

This section explains the following guidelines for testing REXX programs:

- Include a DEBUG parameter in your REXX program
- Run the REXX program outside of CA Automation Point
- View multiple windows during debugging
- Test message-driven REXX programs locally

Include a DEBUG Parameter in Your REXX Program

Before invoking REXX programs from CA Automation Point rules, be sure to test and debug them. A REXX program containing errors can go into a loop without calling or returning control to CA Automation Point.

Note: If you are not sure that a loop in a REXX program will end and you want to ensure that CA Automation Point can cancel the program if it "hangs," insert this statement into the suspect loop:

```
WAIT 0
```

If a REXX program does hang, you can always select the "Stop REXX program" option from the Cmdarea menu.

Example:

Most of the sample REXX programs in this guide contain a statement similar to the following, which starts the REXX trace facility (for REXX programs invoked from within rules, you can edit the rules file and temporarily add the DEBUG parameter):

```
PARSE UPPER ARG . 'DEBUG' +0 DEBUG +1  
IF DEBUG \= '' THEN TRACE R
```

The statement allows the REXX program to execute in "debugging" mode. Invoke the DEBUG option as follows:

```
programe arguments DEBUG
```

The trace output displays in the AXCREXX window.

Testing Procedure

To test a REXX program with the DEBUG parameter from within CA Automation Point

1. Open the Merged Messages window.
2. Choose Cmdarea.
3. Choose Start REXX program or script.
4. Enter the name of the REXX program (and any necessary arguments) with the DEBUG parameter.
5. Jump to the AXCREXX window to view the REXX trace messages.
6. Watch the REXX program execute.

If the program uses the CA Automation Point SESSCMD command processor, jump to the host-session window that the program is controlling. This allows you to watch the command processors as they are being issued.

Run the REXX Program Outside CA Automation Point

To debug a REXX program quickly, you can run it outside of CA Automation Point. You can edit the REXX program in a workstation window while CA Automation Point is active in another. Run your tests by invoking the program with AXCREXX, so that you can test the CA Automation Point command processors.

View Multiple Windows During Debugging

By default, the CA Automation Point desktop window is maximized so that it uses the entire screen. For debugging, however, it is often useful to view multiple windows. To do so, adjust the CA Automation Point desktop window size and move it to allow space for the other windows.

Test Message-driven REXX Programs Locally

Because of the heavy processing that CA Automation Point rules perform, it is sometimes difficult to manually invoke and test the REXX portions of some types of message-driven automation that use a combination of rules and REXX programs.

You can test a REXX program within a rule by using the following methods:

- When you create a test procedure on the host to send dummy messages, the messages activate the CA Automation Point rule, which in turn invokes the REXX program. However, such a scenario may not be possible on the particular host that you are automating.
- If you use the CA Automation Point MSG command processor to issue dummy messages locally from the same workstation, write a separate REXX program to invoke the CA Automation Point MSG command processor one or more times to produce a controlled message stream for testing.
- If you enter a message ID into the host session as a command, the host echoes the command, and then rejects it as invalid, causing CA Automation Point to process the unknown command as a message. The rule that you have written to intercept that message ID executes and the rule then executes your REXX program.

Binding a REXX Program to a CA Automation Point Session

You may need to *bind* a REXX program to a CA Automation Point session, temporarily dedicating the session to processing that REXX program without conflicts from other REXX programs. Binding a REXX program to a session is a good idea because multiple REXX programs could try to write to a session at the same time, causing the programs to interfere with each other.

Note: If the session is automated, you should also pause automation.

The CA Automation Point distribution media provides two REXX programs that you can call from within any other REXX program:

BIND.cmd

Gives the REXX program that calls it control of a CA Automation Point session

UNBIND.cmd

Releases the session for other processing

These programs can be found in the *installDir*\Distrib directory.

You can use BIND and UNBIND programs only if *all* of your REXX programs use them.

How Does Binding Work in CA Automation Point?

Using the BIND.cmd program to bind a REXX program to a session is a *user-implemented* coding standard. It does not bind a session in the same way that you would expect binding to work on the z/OS system.

To be effective at preventing REXX programs other than the current one from writing to a session, *all* of your REXX programs must implement the BIND.cmd and UNBIND.cmd programs.

If one of your REXX programs does not adhere to the standard, it can still write to a session. Also, CA Automation Point and your rules can always write to a bound session.

The BIND.cmd and UNBIND.cmd programs set and clear (respectively) a CA Automation Point status variable (&BIND_essname). When a REXX program that adheres to the standard tries to bind a session, it checks the status variable first. If the variable is clear, the program binds the session—preventing your other REXX programs from writing to that session.

Binding a Session

To call the BIND.cmd program, insert the following statement at the beginning of a REXX program that you intend to bind:

```
CALL BIND sessname waittime
```

sessname

Specifies the name of the session to be bound.

waittime

Specifies the number of seconds that you are willing to wait for the session to become available.

If the session does not become available within the time you specify:

- The session is not bound.
- BIND.cmd issues a return code of 20.
- The return code goes into the special REXX variable RESULT within the REXX program that issued the bind request.

Unbinding a Session

Each REXX program that calls the BIND.cmd program must also call UNBIND.cmd to unbind the session when the calling program terminates. Otherwise, no other REXX programs can run in that session.

To unbind a CA Automation Point session, insert the following statement at the end of the REXX program that issued the bind request:

```
CALL UNBIND sessname
```

This statement releases the session, allowing it to process other REXX programs.

When the BIND and UNBIND REXX programs execute, they generate one of the following return codes:

Code	Meaning
0	Successful execution
4	Syntax error
20	Time expired (BIND) Failure (UNBIND)

Chapter 9: Using Program-to-Program Queues

This section contains the following topics:

[Understanding PPQs](#) (see page 201)

[Design Guidelines](#) (see page 203)

[How to Issue PPQ Commands](#) (see page 204)

[Configuring PPQs](#) (see page 206)

Understanding PPQs

CA Automation Point provides program-to-program queue (PPQ) support. A PPQ is a general purpose, inter-process communication tool. PPQs provide an efficient and universal means for applications to communicate with each other by allowing them to write to—and read from—common message data queues.

A *queue* is a list data structure providing, by default, first-in-first-out (FIFO) access to items in the list. List items are requests waiting for processing, such as commands to execute, operations to perform, or messages to display.

The PPQ environment also supports last-in-first-out (LIFO) and random access. The queue *items* are requests waiting for processing, such as operations to perform or messages to display.

A PPQ is owned by the application that created it. The memory for queue-item storage resides solely on the workstation containing the owner application.

Types of PPQs

The two types of PPQs are:

- Network-shared PPQs are created on one CA Automation Point server and can be shared by applications running on remote CA Automation Point servers. These Network-shared PPQs are connected over a local area network (LAN). When developing automation scripts for your site, you need only reference the name of the network-shared PPQ to access it. The underlying PPQ facility automatically determines on which CA Automation Point server the referenced network-shared PPQ is physically located. Network-shared PPQs can also be shared by multiple applications running concurrently on the local server.
- Local PPQs can only be shared among applications running on the local CA Automation Point server.

General Limitations

The number of queues allowed varies from site to site. Two factors to consider when determining a maximum number of queues are:

- The size of a queue item

The maximum allowable size of any item on any queue is limited to 64 KB.

Note: If you are writing 128-byte items to a queue, you will not run out of memory as quickly as someone writing 64 KB items.

- The number of items written to a queue

A queue can contain up to 10,000 items. The default number of items is 100. You determine the maximum number of items allowed in a queue when you create the queue.

In addition, each queue that you create has a 164-byte overhead (approximately) and each queue item has a 256-byte overhead (approximately).

Memory Requirements

To get a general idea of the PPQ memory requirements for your site, use this formula for each queue you want to create:

$$(\text{Max_Num_Queue_Items} * (\text{Expected_Max_Item_Size} + 256)) + 164$$

Example:

Suppose that you want to create ten queues. For the sake of simplicity, assume that each queue can contain a maximum of 100 items (the default) and that the maximum item size is 1024 bytes (1 KB):

Maximum item size = 1024 bytes (1KB)

Item overhead = 256 bytes (approximately)

Maximum number of queue items = 100 (the default)

Queue "anchor" overhead = 164 bytes (approximately)

The amount of memory required for a single queue is approximately 128,164 bytes (125 KB). If you create ten queues, each having about the same requirements, the total amount of memory needed is approximately 1.25 MB.

Design Guidelines

The following section contains general design principles that you should read carefully *before* attempting to use PPQs:

- Deciding where to create queues

You should create a queue on the machine on which you plan to perform the majority of your processing. This will improve the performance of an application when it is reading and writing queue data by minimizing the amount of work that is executed over network connections.

You should also remember that PPQs do not back up queue data to a disk. When the PPQ server is terminated, all queues that were created locally on that machine are lost. If you need to save your data, you should extract it and save it to a disk.

- Using the product in a client-server architecture

One of the most common uses of PPQs is in a client-server architecture. For example, a client could request a service or function from a server by writing a message to a specific queue. (You should create the queue on the system on which the server runs.) The server would then read the message from the queue and process the request. In this scenario, the server does not have to send any data back to the client.

At times, a client may need to receive data from a server in response to a message that it placed on a queue. In this scenario, you would create a response queue on the system on which the client runs and include the queue name in the message. Including the name of the queue tells the server where it should send the message. The server would then process the request and write a response message to the client's queue. (If you want, you can distinguish the response by designating a client token.) If you want this queue to be temporary, you can delete it after the client processes the message.

- Using the product as a data transfer mechanism

Because PPQs are a data transfer mechanism, you should design your applications so that they allow only minimal quantities of data to accumulate within the queues. Data should be placed into a queue by one process and pulled out of the queue by another process as soon as possible. You should never use a queue as a data repository.

- Using the LIST command

For security reasons, the LIST command does not display every queue that exists in the PPQ complex. Users are only able to see the queues that they have read from, written to, or listed. To use a queue, you either need to be the creator of the queue or know the name of the queue.

- Verifying return codes
You should always check the return codes from PPQ calls because a code that may indicate an error in one situation can be acceptable in another situation. For more information, see information on PPQ return information in the *Command and Keyword Reference Guide*.
- Determining the size and number of items in a queue
You should try to keep the length of queue items at 100 to 200 bytes each. Because very long items use large amounts of memory, performance will be optimized if a queue contains a small number of very large items.
You should also try to keep the number of items within a single queue relatively small. (The default number of items is 100.) Generally, your applications run better if you have a few small queues rather than one large queue.

How to Issue PPQ Commands

You can issue PPQ commands from:

- REXX programs
- CA Automation Point rules
- CA OPS/MVS

The following sections describe each method for issuing PPQ commands.

Issuing PPQ Commands from REXX Programs

Issue a PPQ command from within a REXX program by specifying an ADDRESS PPQ statement, as shown:

```
ADDRESS PPQ 'ppqcommand operand(s)'
```

The following example creates a shared queue named MYQUEUE:

```
ADDRESS PPQ 'CREATE QUEUE(MYQUEUE) SHARE(YES)'
```

PPQ Command Summary

The following table provides a quick summary of the PPQ commands that you can issue through the ADDRESS PPQ statement in a REXX program. See the *Command and Keyword Reference Guide* for command syntax.

Use	Command	Description
PPQ setup	CREATE	Creates a new queue

Use	Command	Description
PPQ operations	LOCK	Prevents access to a queue by REXX programs other than the current one
	READ	Reads (accesses) one or more items from a specified queue
	UNLOCK	Restores access to a queue locked by a previously issued LOCK command
	WRITE	Writes (inserts) one or more items to a specified queue
Dismantling PPQs	DELETE	Deletes all elements in a specified queue and releases the memory allocated for the queue
	DISCONNECT	Breaks the connection with a specified remote queue and closes all sessions with the remote queue
Special PPQ commands	COUNT	Counts the number of items in a specified queue
	DEBUG	A diagnostic tool that controls debugging trace output. This command is useful for providing diagnostic information to Technical Support.
	LIST	Lists information about one or more queues residing on the local or remote machine, or both
	TRANSTATUS	A diagnostic tool that returns a table of transport-specific information. This command is useful for providing diagnostic information to Technical Support.
	VER	Provides the version number and configuration of CA Automation Point PPQs

Issuing PPQ Commands from Rules

Issue a PPQ command from a rule by specifying the PPQWRITE keyword, as shown:

```
PPQWRITE((item) QUEUE(queueName))
```

The following example sends information to a queue named MESSAGE:

```
PPQWRITE((HELLO THERE) QUEUE(MESSAGE))
```

For details, see the description of the PPQWRITE keyword in the *Command and Keyword Reference Guide*.

Issuing PPQ Commands from CA OPS/MVS

You can write to a PPQ on the CA Automation Point workstation using the CA Automation Point interface to CA OPS/MVS. For more information about the interface, see the chapter "[Using the CA OPS/MVS Interface](#) (see page 389)."

Configuring PPQs

If you installed Program-to-Program Queues (PPQs), you need to select the appropriate setting for PPQ transports from the list of available network transports on the Program-to-Program Queues dialog.

To configure PPQs

1. 1. Open Configuration Manager.
2. 2. Select Expert Interface, Infrastructure, Program-to-Program Queues.
The Program-to-Program Queues dialog displays.
3. 3. Click Enable use of PPQs.

PPQs use the system name as a unique identifier for this system in its network communications. By default, the PPQ system name is your computer name (shortened to eight characters). If your computer name is longer than eight characters, you should override the default and choose an eight-character name that is unique to all machines that will be running PPQs.

Using the TCP/IP Transport

To use the TCP/IP transport:

Follow these steps:

1. Enable the TCP/IP check box.
2. List the CA Automation Point servers that are permitted to share access to locally created PPQs over the network.

If you configured PPQs to use the TCP/IP transport, it defaults to TCP port number 2000. This port is acceptable when it is not being used. When the port is used change the default to another available port.

Notes:

- All PPQs within your complex must be configured to use the same TCP port.
- When entering IPv4 or IPv6 addresses in the TCP/IP Host Names list, follow these guidelines.
 - IPv4 addresses can be specified explicitly and the DNS server does not need to resolve these addresses.
 - IPv6 addresses can be specified explicitly. Enclose IPv6 addresses in square brackets ([]). The DNS server does not need to resolve these addresses.

PPQSTAT.bat Program

Run the PPQSTAT.bat program (located in the *installDir*\BIN directory) to display the status of CA Automation Point PPQs. You can use PPQSTAT.bat to verify successful installation or to monitor the status of PPQ network transports. PPQSTAT.bat displays the following:

- Queues on the local machine
- Queues from the remote workstations that are accessed by REXX programs on the local workstation
- The current state of all Configured Network Transports
- The current state of each TCP/IP connection listed under TCP/IP Host Names in the Program-to-Program Queue dialog

Chapter 10: Using Notification Services

This section contains the following topics:

- [Introducing the VOX Environment](#) (see page 209)
- [Paging Capabilities](#) (see page 215)
- [Configuring Notification Services](#) (see page 217)
- [Configuring the Dialogic Voice Card](#) (see page 218)
- [Configuring Call Progress Analysis \(CPA\) Parameters](#) (see page 220)
- [Configuring Channel Groups](#) (see page 223)
- [Configuring Voice Channels](#) (see page 224)
- [The Voice Word Library](#) (see page 225)
- [Configuring Answer Tree](#) (see page 226)
- [Configuring the Email Feature](#) (see page 231)
- [Configuring the Text-to-Speech Feature](#) (see page 232)
- [Two-Way Paging](#) (see page 235)
- [The VOX Command Environment](#) (see page 237)
- [Example VOX Environment Applications](#) (see page 244)

Introducing the VOX Environment

The VOX environment is multi-channel inbound and outbound voice processing software that transforms your workstation into a sophisticated voice server with paging capabilities. While a significant part of the VOX environment function is to provide notification services, paging and email services are also provided.

The VOX environment is easy to use. You can quickly create elaborate voice applications using the VOX environment voice processing (VOX) commands in your REXX programs. ADDRESS VOX statements in your REXX programs enable you to access the VOX environment command environment. For examples of REXX-based voice applications, see [Example VOX Environment Applications](#) (see page 244) in this chapter.

VOX Environment Capabilities

Possible application scenarios for the VOX environment include the following:

- **Paging**

A person can be notified of a particular event at any time of day or night by a numeric or alphanumeric page. A message can be sent to define the specifics of the event. If the pager supports two-way communication, this device can then be used to respond directly to the event that triggered the notification. If a one-way pager is used to receive the notification, the incoming telephone answer capabilities provided by the Notification Server can be used to remotely respond to the event. The combination of one-way paging, two-way paging, and telephone answer capabilities can provide a powerful tool for solving problems without having to go into the office. It also provides a way to inform an operator when certain events, such as backups, have been completed.

- **Help desk**

Through the Answer Tree application, the VOX environment waits for an incoming call, verifies a caller user ID and password, and presents the caller with a list of system incidents. When the caller selects an incident item, the VOX environment plays back a voice file that details the status of the incident. If appropriate, the VOX environment presents a menu of possible actions.

- **Problem escalation**

A specified event occurs, causing the VOX environment to play an announcement message over an intercom system one or more times, possibly increasing the volume and preamble each time. If an operator does not respond, the VOX environment dials the extension of each person on an internal notification list and plays the message. If the VOX environment receives no response, it begins to call each person on an on-call list, activating each beeper in the order listed. During each step, a specified time period elapses before the VOX environment repeats the announcement message or dials the next number.

- **Auto-attendant**

One or more voice channels are waiting to answer an incoming telephone call. On the specified ring, the VOX environment answers the call and presents the caller with a menu of several options. According to the menu selection, the VOX environment presents the caller with another menu or transfers the caller.

VOX Environment Features

This section describes the available features of the VOX environment:

- Voice channel support
- Call Progress Analysis customization
- Channel grouping
- Voice file support
- DTMF (Touch-tone) and MF Input
- Volume Control

Voice Channel Support

The VOX environment supports up to 36 analog voice channels in a single workstation. The number of channels that your REXX programs can realistically use at one time depends on the following factors:

- Peak load requirements
- Your workstation CPU type and speed
- The access time of your workstation's hard disk drive
- Whether the voice files reside in memory or remain stored on disk

The VOX environment can do the following:

- Answer and process incoming calls
- Place outgoing telephone calls, digital pager (beeper) calls, or alphanumeric pager calls using the pager applications
- Obtain and process DTMF (touch-tone) responses from a remote party
- Record a voice message from a remote party onto a disk file
- Send additional tone digits *after* establishing a connection

Call Progress Analysis Customization

The Call Progress Characterization (CPC) program accurately determines the cadence of both ringing and busy signals. You can insert the information that the CPC program provides into call progress analysis (CPA) variables using the CA Automation Point Configuration Manager, enabling you to customize the VOX environment call progress analysis for your environment.

For more information, see [Configuring Call Progress Analysis \(CPA\) Parameters](#) (see page 220) in this chapter.

Note: The VOX environment supports Dialogic voice cards. The CPC program is available from Dialogic.

Channel Grouping

The VOX environment enables you to logically group voice channels so that you can control the available channels more easily.

Example:

Suppose that you have a 12-line telephone system with the lines allocated as follows:

- Six channels attached to an incoming 800 number
- Four channels for internal help desk calls
- Two channels for customer support purposes

You could define three channel groups (incoming 800, help desk, and support). Then, you could issue commands pertaining to each group: answer any incoming 800 channel, secure the first available support channel to dial a digital beeper, and so on.

Note: For more information, see the section [Configuring Channel Groups](#) (see page 223) in this chapter.

Voice File Support

In addition to its own voice word library, the VOX environment supports multiple non-indexed voice files. A non-indexed voice file contains the voice data of a single prerecorded voice message and resides on a notification server workstation.

The VOX environment plays multiple non-indexed voice files without pauses or clicks between each file.

Note: For more information, see the section [The Voice Word Library](#) (see page 225) in this chapter.

DTMF (Touch-tone) and MF Input

The VOX environment accepts input from touch-tone and network devices. Touch-tone telephones generate Dual Tone Multi-Frequency (DTMF) signals. Telephone network devices such as Central Office (CO) and Private Branch Exchange (PBX) switches generate Multi-Frequency (MF) signals.

Note: Do not use the VOX environment with rotary-dial telephone systems. Your workstation's voice card cannot accurately detect a rotary system's audio-pulse and loop-pulse signals.

See the *Command and Keyword Reference Guide* for more information about DTMF and MF input.

Volume Control

You can control the volume at which the VOX environment plays your voice messages by issuing the SETVOLUME command, which is described in the *Command and Keyword Reference Guide*.

Understanding the VOX Environment's Components

The VOX environment consists of these primary components:

- The notification server
- The VOX command environment
- The voice word library

Notification Server

At the heart of the VOX environment, a notification server performs these tasks:

- Services all voice processing requests from the VOX client
- Manages the VOX environment interaction with the workstation voice card
- Manages the VOX environment interaction with the workstation modem to provide one-way paging capabilities
- Manages the VOX environment interaction with the Internet-accessible paging services to provide two-way paging capabilities
- Manages the VOX environment interaction with the selected mail protocol to provide email capabilities
- Returns result information to the VOX client

Note: Notification messages display in the AP Notification Messages function window on the CA Automation Point desktop provided that you first enable this function window in your session definition set.

VOX Command Environment

The VOX command environment enables your REXX programs to request notification services from one or more notification servers that are local or network-connected by PPQs. Resulting information is returned to the calling REXX program.

For information on specific VOX commands and VOX return information, see the *Command and Keyword Reference Guide*.

Voice Word Library (VOXLIB)

The voice word library contains an index file and a voice data file, giving the VOX environment a capability similar to text-to-speech.

The *index file* contains entries for each word in the library. Each entry consists of the following:

- The text of each word
- The starting position of—or pointer to—the associated (digitized) speech in the voice data file
- The length of the digitized speech segment (in bytes)

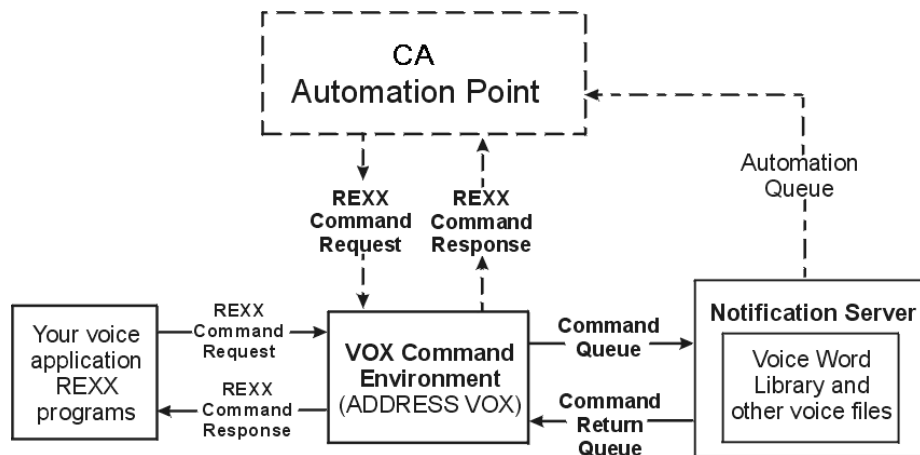
The *voice data file* contains the corresponding digitized speech data for each word in the index entry.

If a word does not exist in the voice word library, the VOX environment sounds out (spells) the letters of the unknown word, preventing the remote party from otherwise missing parts of a message.

Note: For more information on maintaining the voice word library, see the online help.

How the VOX Environment Components Work Together

The following illustration shows how the components of the VOX environment work together:



Paging Capabilities

CA Automation Point can issue page requests to any paging service that supports any of the following communication protocols:

- Telocator Alphanumeric Protocol (TAP)
- Simple Network Paging Protocol (SNPP)
- Wireless Communication Transfer Protocol (WCTP)

The TAP protocol (currently the most widely used protocol used by alphanumeric paging services in the United States) uses a modem to establish a direct connection to the paging service. This TAP connection, by nature of the protocol, can only be used to submit page requests. (Replies from two-way paging devices cannot be queried through this interface.) In contrast, the SNPP and WCTP protocols both use the Internet to connect to the paging service and provide mechanisms to receive solicited responses from two-way paging devices.

CA Automation Point provides several methods for initiating page requests to individuals. These methods include:

- **Issuing a page request through the VOX environment**—With the VOX PAGE command processor, you can issue a page request to any paging service supporting TAP protocol. With the VOX PAGE2WAY command processor, you can issue a page request and receive a reply to this request from any paging service supporting either the SNPP protocol or the WCTP protocol.
- **Issuing a page request with the TAPPAGE command**—CA Automation Point provides a REXX program, Tappage.rex, for issuing page requests directly through REXX without the use of the VOX environment. It conducts its own I/O directly with the modem that is dialing to the paging service. See Tappage.txt for details about using this program.

The advantages of issuing page requests directly through REXX include:

It does not require the notification server, which operates as a Windows service, to be active when voice or email services are not being used.

Although the TAPPAGE command supports the TAP protocol, the REXX program itself can be customized to support protocols other than TAP. Paging services in many countries outside the United States use other protocols.

- **Issuing a page request with the MAILPAGE command** —CA Automation Point provides a sample REXX program, Mailpage.rex, for issuing page requests by sending emails directly to the paging service. The MAILPAGE command issues the email messages through the VOX environment with the VOX SENDMAIL command processor. See Mailpage.txt for details about using this program, and for information about formatting email messages for the example paging services.

The advantages of issuing page requests through email messages include the following:

- A modem connection is not required.
- Call costs can be significantly reduced.

Batching Page Requests

Both the VOX PAGE command and the Tappage.rex program support the batching of page requests internally. *Batching* refers to the capability of sending multiple page requests that occur within a short period of time (15 seconds) with one call to the paging service, depending on the ability of the paging service to handle this. The advantages of batching page requests include the following:

- The average time to issue page requests is reduced because the process of establishing and dropping the call is performed only once for many requests
- Call costs can be significantly reduced

Both the VOX PAGE command and the Tappage.rex program require a modem. Depending on your level of concurrent paging activity, you may decide to go with more than one modem. Because each modem you use and each asynchronous host console you are monitoring requires a COM port, you may need to use a serial port expansion device to extend the number of ports.

Note: For more information about serial port expansion devices, see the chapter "[Asynchronous Host Sessions](#) (see page 61)."

TAPI Support

The notification server can use the Microsoft Telephony Application Programming Interface (TAPI) to drive modem initialization for page requests that require a modem connection, instead of using its own internal modem driver. You can configure TAPI using the Windows Control Panel. TAPI allows you to set the options for your modem without impacting any existing application code using a GUI provided by the Control Panel modem icon. TAPI also allows you to initialize each modem using the manufacturer's modem driver written specifically for TAPI, rather than the modem driver internal to the notification server, which was written for standard modems. Nonstandard modems should specify TAPI for modem initialization.

To configure the notification server to use TAPI

1. Install your modem on Windows:
 - a. Click Start on the taskbar
 - b. Choose Settings, Control Panel, and double-click the Phone and Modem Options icon.

A wizard guides you through the process of installing your modem.

2. Enable the communication port for TAPI:
 - a. From Configuration Manager, choose Expert Interface, Notification Services, Paging, Alphanumeric Paging.
The Alphanumeric Paging Options dialog displays.
 - b. Select a COMPort from the Available paging devices list by clicking on the COMPort name.
 - c. Check Use TAPI modem initialization
 - d. Click OK.

Note: Any COMPort that does not have a modem attached to it should have the Enable selected device for use by Notification Server box unchecked. This will prevent Notification Server service from attempting to access these COMPorts.

Configuring Notification Services

This section describes options for configuring the notification server and the VOX client components of CA Automation Point. The options for each component are found in different sections of the same dialog.

Notification Server Options

To configure notification server options

1. Open Configuration Manager.
2. Choose Expert Interface, Notification Services, Notification Services Startup Options. The Notification Services Startup Options dialog displays. The Notification Server area of the dialog controls the startup options for the notification server. (This area is disabled if the notification server has not been installed on this workstation.)
3. Check Enable Server.

You can now use the notification server on this workstation.

Note: For information about the other fields in this dialog, use the dialog help.

VOX Client Options

The VOX client must be installed on any machine that runs Notification Manager or for any REXX programs that issue ADDRESS VOX commands.

Note: The VOX client is automatically enabled if the notification server is enabled.

To configure VOX client options

1. Open Configuration Manager
2. Choose Expert Interface, Notification Services, Notification Services Startup Options.

The Notification Services Startup Options dialog displays.

The VOX Client area of the dialog controls the startup options for the VOX client. This group is disabled if the VOX client has not been installed on this workstation.

3. Check Enable Client.

You can now use the notification server on this workstation.

Note: For information about the other fields in this dialog, see the dialog help.

Installing the Dialogic Voice Card

Important! The voice card resides in a workstation that uses notification services. Perform the tasks in the section [Configuring the Dialogic Voice Card](#) in this chapter *only* if you plan to install the notification server component on this workstation.

You must install and configure the Dialogic voice card and ensure that it is working properly before you install and customize the notification services function. If you do not plan to use a Dialogic voice card with the CA Automation Point notification services, skip ahead to the section entitled [Configuring the Email Feature](#).

Note: The notification server also supports paging and email; a voice card is not needed for these functions.

Configuring the Dialogic Voice Card

This section describes how to configure one or more Dialogic voice cards. You should have physically inserted the appropriate card in your machine before proceeding with the following instructions.

Installing the Dialogic Software

To install the Dialogic software from your CA Automation Point DVD

1. Start the Dialogic Voice Card Driver installation program from the Configuration Manager main window.
2. Choose Expert Interface, Notification Services, Voice, Voice Card Driver.
3. Insert the CA Automation Point DVD #2.
4. Specify the root directory of your DVD drive. For example, if your DVD drive is the D: drive, you would specify the following:
D:\
5. Click OK.
The Welcome to Setup window displays.
6. Continue with the Dialogic installation program until the Select Components dialog is displayed.
7. On the Select Components dialog, select at least the Core Runtime Package for installation.
This option includes the device drivers that CA Automation Point needs to run its voice feature.
8. During the voice card driver installation, you may see a dialog called Found New Hardware Wizard. If this dialog displays, click Cancel to close the window.
9. During installation, you may see one of the following dialogs:
 - Security Alert – Driver Installation
 - Windows SecurityThese dialogs warn you that these voice card drivers have not been digitally signed. Click Yes or Install to continue with the installation process.
10. On the Setup Complete dialog, choose to reboot now by clicking Yes, then click Finish.
11. After you reboot the server, you must configure the voice card for automatic startup using the Dialogic Configuration Manager (DCM) application. You can launch this application by selecting Start, Programs, Dialogic System Release, Configuration Manager - DCM.
The Dialogic Configuration Manager dialog displays.
12. On the Dialogic Configuration Manager dialog, select Settings, System/Device Autostart, Start System.
This option automatically starts the voice card drivers when the server is rebooted.
13. Reboot the server before configuring the voice card for use with CA Automation Point.

Note: You may see several pop-up dialogs during this process. Click OK on these dialogs to continue the configuration process.

Troubleshooting the Dialogic Product System Service

If you have problems starting the Dialogic product System Service, you may need to reconfigure the device properties.

To reconfigure the device properties

1. From the Start menu, select Programs, Dialogic System Release, Configuration Manager - DCM.

The Dialogic Configuration Manager dialog appears.

2. Highlight the device you want to reconfigure.
3. From the Device menu, choose Configure Device.

The Dialogic Configuration Manager — Properties dialog displays.

4. Make the appropriate changes.
5. Click OK to save your changes.
6. From the System menu, choose Start System. (You can also do this by clicking the green arrow on the tool bar.)

Note: The parameter you change depends on the model of card you are using and the reason for failure. To see errors associated with the Dialogic product System Service, see the Event Viewer System Log.

Configuring Your Workstation for Voice Processing

If you install a voice card in your workstation, you can play back prerecorded voice files through CA Automation Point command processors invoked from REXX programs.

Configuring Call Progress Analysis (CPA) Parameters

The Dialogic voice cards provide built-in call progress analysis (CPA) parameters. With assistance from Dialogic or CA Technical Support, you can modify the CPA parameters to optimize the voice card performance and operation.

Note: Dialogic reserves the right to add, delete, or modify CPA parameters without notice.

You can modify CPA parameters permanently, using the Configuration Manager.

WARNING! When *permanently* modifying CPA parameters, be aware that:

- The VOX environment can operate unpredictably if you change parameter settings improperly. (If you need help, contact Technical Support at <http://ca.com/support>.)
- The Dialogic voice card requires that you restart the notification server workstation on which you make parameter changes. (Restarting is necessary for the changes to take effect.)

You should be extremely careful when making changes to CPA parameters because incorrect parameter combinations can have an adverse effect on the functionality of all CA Automation Point notification services. *We strongly recommend that you do not make changes to CPA parameters without first consulting with CA Technical Support.*

To configure CPA parameters from a Configuration Manager client, ensure that the Configuration Manager client is connected to a notification server. From the Configuration Manager main window, choose Expert Interface, Notification Services, Voice, Call Progress Analysis Parameters. The Call Progress Analysis Parameters –EngineDefault dialog displays automatically open to the Sets property sheet.

This dialog is discussed in the following sections.

Inserting a New Named Set of CPA Parameters

You manage named sets of CPA parameters from the Sets property sheet. Generally, you should not make any changes to the Engine Default set because you may need to restore this configuration if the changes leave the notification services inoperable.

To create a new named set of Call Progress Analysis parameters

1. Type the name of the set in the Name field.
2. Click Insert.

The Configuration Manager client creates a new named set of CPA parameters that contains the same settings as the set that was selected when you clicked Insert.

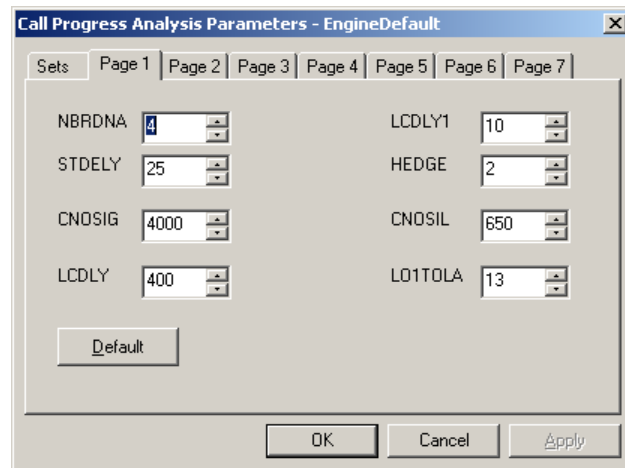
Important! The Apply button at the bottom of the property sheet becomes active when you click Insert. The Apply button commits changes to the notification server in the same way as the OK button; however, the Apply button does not close the property sheet like the OK button does. If you want to add multiple named sets of CPA parameters to the notification server, you *must* click Apply after each set you insert.

To add multiple named sets of CPA parameters

1. Type the name of the set in the Name field.
2. Click Insert.
3. Change any parameters that you want for the set. (For details, see the section [Configuring Call Progress Analysis](#) (see page 220) parameters that follows.)
4. Click Apply.
5. Repeat Steps 1 through 4 for each set you want to add.

Changing Individual CPA Parameters

After you have inserted a new named set of CPA parameters, you can choose another property sheet from the dialog. The following is an example of the Page 1 property sheet:



The CPA parameters are listed on the Call Progress Analysis Parameters pages. You can change the value for a parameter by either typing the new value over the old value, or by selecting a value using the up and down arrows. The field does not allow you to set a value outside the correct range for the parameter you are editing. Validate the new value by clicking Apply or OK.

The Apply button becomes active any time you make a change to a value, allowing you to commit your changes to the server immediately without closing the dialog. For example, you may want to use the Apply button if you are making incremental changes to a notification server followed by a test of the notification services.

For descriptions of each field on the Call Progress Analysis Parameters pages, click the dialog help button.

You can reset the default values for any page in the property sheet by clicking Default.

When you are finished making changes to the parameters, click OK to commit your changes to the notification server.

Note: The Cancel button will not commit changes to the parameters if you have not clicked Apply.

Configuring Channel Groups

Important! You can configure channel groups only on the workstation on which you plan to run the notification server.

To configure channel groups for the notification server, from the Configuration Manager main window, choose Expert Interface, Notification Services, Voice, Channel Groups.

The Channel Groups dialog displays. This dialog lets you to create and delete groups, insert and remove channels from those groups. The Channel Groups dialog always defaults to the group ALL when first displayed. The ALL group contains all channels that are available for the voice card you are using. You cannot modify the ALL group.

For descriptions of each field in the Channel Groups dialog, see the Configuration Manager help.

When a VOX channel group is defined as interruptible, the channels in the group can be released from a pending ANSWER GROUP when another program requests it using GETCHANNEL. When the other program releases the channel, the notification server automatically adds it back to the ANSWER GROUP.

Note: The interruptible state of a group cannot be changed if the group is the NMANSWER group or if the group has been defined to the AnswerTree application.

To create a new group of channels for the notification server

1. Type the name of the group in the Group Name field.
2. Click Create.
A new group is created that contains no channels. All channels that are available from the card are shown in the Available from Card list.
3. Use the Insert and Remove buttons to insert and remove channels from the selected group.
4. Click OK to commit your changes to the notification server or click Cancel to discard your changes.

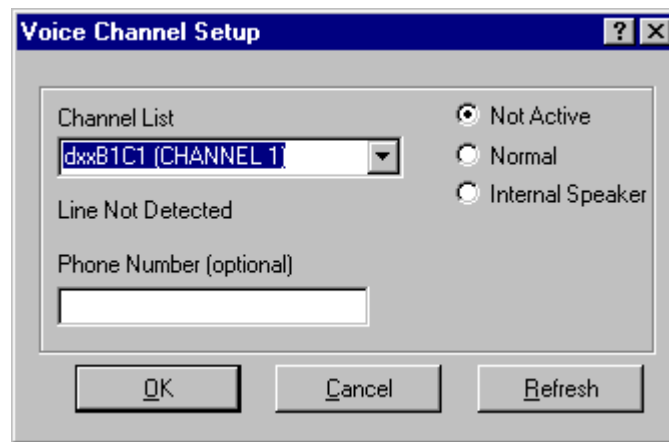
Configuring Voice Channels

Before using the notification server, you should run the Voice Channel Setup utility. This utility sets up the channels for use with the notification server, allowing faster access to the channels. It also lets you use special channel-related features such as an internal speaker.

To set up the channels for use with the notification server:

1. From the Configuration Manager main window, select Expert Interface.
2. Select Notification Services, Voice, Channels.

The Voice Channel Setup dialog displays:



Note: If this is your first time running the Voice Channel Setup utility, Configuration Manager initiates the channel detection process before displaying the Voice Channel Setup dialog. This channel detection process may take some time to complete, especially if the voice card contains a large number of channels.

If there is an active analog phone line connected to the channel, the channel's default value will be Normal. If an analog phone line is not detected, the default value will be Not Active. You can change the channel options to any of the following supported configurations:

Not Active

Indicates that the channel *is not* available for use in the notification server.

Normal

Indicates that the channel *is* available for use in the notification server.

Internal Speaker

Indicates the channel *is* available for use with an internal speaker (this speaker can be attached to the voice card—refer to the Dialogic documentation).

Use the Refresh button to detect physical changes to your voice card channels, such as adding or removing analog phone lines.

The Voice Word Library

A voice library of spoken words is contained in the file sets VOXF_A.vdi and VOXF_A.vds, which are located in the subdirectory in which you installed the VOX environment.

Note: The VOX environment ships with two sets of voice library spoken words, one by a female voice and one by a male voice. These files are listed as VOXF_A.* (female set) and VOXM_A.* (male set).

Within a REXX application, words from the word library can be played using the PLAY, CALLPLAY, PLAYGETDIGITS, or ANSWERPLAY ADDRESS VOX commands. This is achieved by coding FILETYPE(WORDLIB), with the FILE() and VAR() keywords (filled in appropriately) on the command. For details about these commands, see the *Command and Keyword Reference Guide*.

The .vdi file is an index (that is internally used by the VOX environment) to the .vds data file. Each word in the .vdi file refers to a particular voice stream or message segment in the .vds data file. If a particular word is not found in the .vdi file, the VOX environment sounds out the word (by explicitly spelling the word) and VOX message VOX5657I or VOX5658I is issued to the VOX environment Configuration Manager (and log), specifying the letter or word that could not be found.

Note: Within a REXX application, when referring to the word library, do not code the .vdi or .vds extension within the FILE() keyword.

The Voxmaint Application

The Voxmaint application is a GUI that allows you to perform the following actions on the words in a voice library maintenance voice word library:

View

View an existing voice word library.

Create

Create a new voice library rather than using the male and female libraries that were supplied with CA Automation Point.

Import

Add additional words to a voice word library.

Rename

Rename an existing word in a voice word library.

Delete

Delete a word from a voice word library.

Export

Export a *.vox word from a voice word library.

To access the Voxmaint application

1. Start Configuration Manager.
2. Select Expert Interface, Notification Services, Voice, Voice Library Maintenance, Open Voice Library.
3. When prompted, select the appropriate file.

The Voxmaint dialog displays. All the words in the selected voice word library are displayed in the list.
4. To perform a function, either use the buttons on the dialog or right-click a word and choose an option from the pull-down menu.

For more information about the specific functions of the Voxmaint application, see the online Help.

Configuring Answer Tree

Answer Tree is a feature of the CA Automation Point notification server. Answer Tree, which is written in REXX, allows you to implement a generic, configurable help desk utility.

This application provides the ability to wait for incoming calls on all channels in a group. Use the Answer Tree dialog to configure a tree-like structure of voice prompts to be played or REXX programs to be invoked when a call is received on one of the voice channels in the configured channel group.

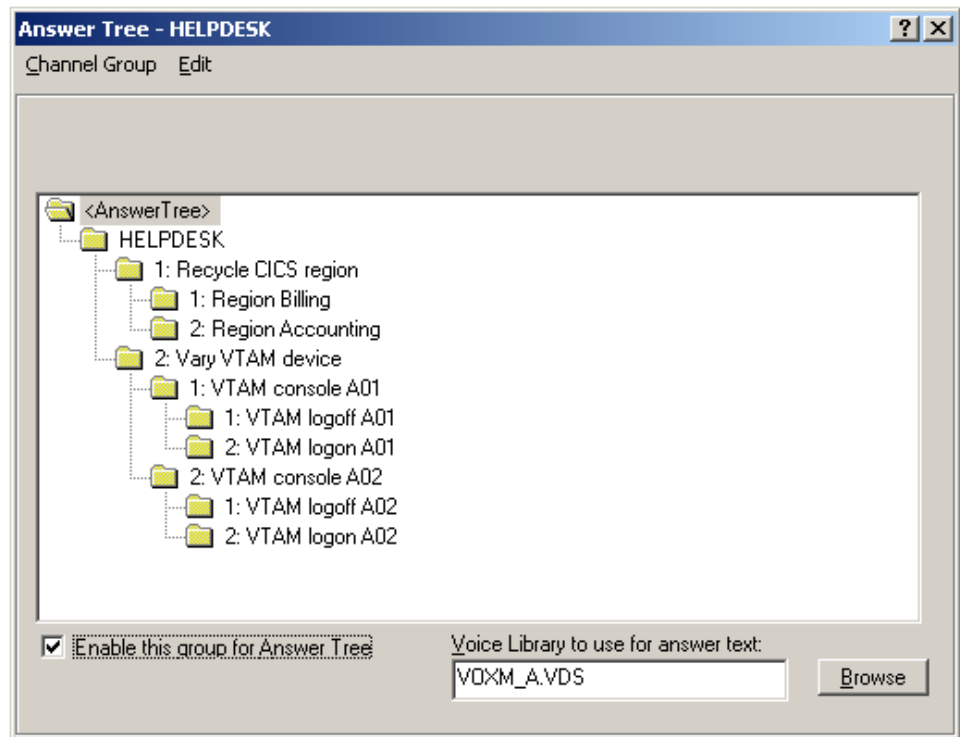
When an incoming call is detected on a monitored group of voice channels, the Answer Tree application reads the data configured by the dialog and guides the caller through the hierarchy of voice prompts and REXX programs, based on the caller's input of tone digits. In this manner, a series of prompts can be navigated and one or more actions taken based upon the user's input.

Any messages generated by Answer Tree are displayed in the CA Automation Point Notification Messages window. Because all messages appearing in the Notification Messages window are processed through rules, you can write rules to automatically handle any Answer Tree error conditions. For more information, see the *Message Reference Guide*.

Note: User-written REXX programs can be external files in the Myfiles\Rexx subdirectory of your current Site directory. See the file ATTemplt.rex in the \SAMPLE\ANSTREE directory for an example of how to write external REXX programs. You must supply the file extension (.rex or .cmd) when you specify the REXX program in the Answer Tree dialog. There is also a working sample named ATAskID.rex in the \SAMPLE\ANSTREE directory.

Sample Answer Tree Configuration

The following dialog is a sample HELPDESK application configured using the Answer Tree feature:



For a sample sequence of events using the HELPDESK group as configured previously, consider the following:

When a user calls into one of the channels in the HELPDESK group, the Answer Tree application:

1. Picks up the call
2. Plays a welcome message (this is a pre-recorded voice file that is supplied with Answer Tree)
3. Says "Press 1 to recycle a CICS region. Press 2 to vary a VTAM console online"
4. Waits for the user to press 1 or 2
 - If the user presses 1, the application advances down the tree structure under Recycle a CICS region. Next, the user will hear "Press 1 to recycle region Billing. Press 2 to recycle region accounts payable." Again, Answer Tree waits for the user to enter 1 or 2.

- If the user presses 2, he/she then hears "Recycle accounts payable," and the REXX program PAYABLE.rex is invoked. After the REXX program runs, the user hears "Goodbye," and the line hangs up. Answer Tree then resets itself to begin waiting on the line for another user to call in.

Accessing Answer Tree

To access the Answer Tree dialog, open Configuration Manager. On the Configuration Manager main window, choose Expert Interface, Notification Services, Voice, Answer Tree. Use the Edit menu or right-click in the main window to access controls for this dialog.

Note: When accessing the Answer Tree dialog for the first time, the Answer Tree option is disabled. To enable the Answer Tree option, you must first configure an Answer Tree group. After you configure an Answer Tree group, the Answer Tree application is enabled.

Configuring an Answer Tree Group

Answer Tree runs on any CA Automation Point workstation that has a notification server and a voice card installed. From Configuration Manager, run the Notification Setup Wizard at least once before setting up Answer Tree.

Setting up Answer Tree requires several procedures. Each of these procedures is explained in detail in this section.

- Configuring a Channel Group
- Selecting the Channel Group and Opening the Answer Tree Dialog
- Editing the Answer Tree Voice Prompt Menu
- Enabling Channel Group for Answer Tree

Configuring a Channel Group

To configure a channel group

1. From within the Configuration Manager, choose Expert Interface, Notification Services, Voice, Channel Groups.

The Channel Groups dialog displays.

2. In the Group Name field, select the ALL group and type over it, entering the name of the new channel group.

Note: The new channel group cannot be named ALL, NMANSWER, or NMCALL.

3. Uncheck the Interruptible check box.
4. Click Create.

5. Use the Insert and Remove buttons to insert channels into and remove channels from the group available from the card.
6. Click OK.

Selecting the Channel Group and Opening the Answer Tree GUI

To select the channel group and open the Answer Tree GUI

1. From the Configuration Manager main window, choose Expert Interface, Notification Services, Voice, Answer Tree.
2. Select Channel Group, Open.
A list of defined channel groups displays.
3. Click the channel group you want to configure.
The opened channel group name is displayed in a tree from which you can configure voice prompts and actions.

Editing the Answer Tree Voice Prompt Menu

To edit the Answer Tree Voice Prompt Menu

1. From within the Answer Tree dialog, use the Edit menu selections or right-click the main window to access controls for this dialog.
2. To edit a voice prompt for a menu, select or highlight the menu item on the Answer Tree.
3. Select Edit and Insert, Change, or Delete from the menu.
When inserting or changing a menu, the Insert Answer Tree data or Change Answer Tree data dialog displays.
4. Enter or edit the name, answer string, and REXX program fields.
The name displays in the Answer Tree window. The answer string is a voice prompt in the form of text or the file name of a prerecorded voice file. In addition to the answer string, you can optionally specify a REXX program to be executed.
Note: When specifying the text for the voice prompt, refer to the Voxmaint utility for the available words in a given voice word library.
5. When the voice prompt menu is complete, choose Channel Group Save to save, or choose Channel Group Exit to exit without saving the changes.

Enabling a Channel Group for Answer Tree

To enable a channel group for Answer Tree

1. Open the Answer Tree dialog with the channel group you want to enable for Answer Tree.
2. Check the box labeled Enable this Group for Answer Tree. At this time, you can also choose between the male or female voice word libraries or your own library. Answer Tree uses the library you select for all text strings specified in the Answer Actions edit box.
3. Choose Channel Group Save to save this information.

Note: If necessary, Configuration Manager automatically stops and restarts the services needed to enable these changes.

Configuring the Email Feature

You can use the Notification Server to initiate email notifications by submitting send mail requests to a mail server, provided Simple Mail Transfer Protocol (SMTP) over TCP/IP connectivity between the Notification Server and the mail server can be established. To configure Notification Server for email notification, you must specify the hostname for the mail server and the TCP/IP port number the mail server is listening on. If an authentication mechanism is in operation, you must also configure associated settings before the Notification Server can successfully establish an SMTP connection with the mail server.

If the Notification Server fails to successfully submit a send mail request, the Notification Server can be configured to resubmit send mail requests to the next mail server on a predefined list of available mail servers.

To configure the email feature of the Notification server

1. From the Configuration Manager main window, choose Expert Interface, Notification Services, Email.
The Email Settings dialog displays.
2. Select the Enable Mail Functions check box.
3. Create a list of mail servers with which the Notification Server can establish connections.

4. Add a server to the list of mail servers:
 - a. Click Add.
The Enter new server name dialog displays.
 - b. Enter a server name. Click OK.
 - c. Enter the host name for the SMTP mail server.
 - d. Enter the port number on which the SMTP mail server is listening for mail requests.

5. Repeat step 4 to add more servers.

6. Prioritize the mail servers.

The order in which the mail server names appear in the list is the order in which the Notification Server attempts to submit a mail request. The Notification Server first submits mail requests to the mail server at the top of the list. To assign priority, select a mail server in the list and click the up and down arrow buttons to position it in the list according to priority.

7. Enter a valid email address in the AP From mail address field. This address is the email address that displays in the From: section of an email message sent successfully by Notification Server.

8. Optionally, you can set the Server List retry value from 0 – 10.

When the following messages appear in the CA Automation Point Notification Services Console window, the notification server is ready to submit mail requests:

```
VOX5660I Mail handler is active.
```

Configuring the Text-to-Speech Feature

The CA Automation Point text-to-speech feature uses either the Microsoft Agent technology (on platforms that support this technology) or the Microsoft Speech API (SAPI) to provide text-to-speech capabilities for CA Automation Point messages sent over the TCP/IP network.

This feature works in a client-server model. You can send a text-to-speech message from the CA Automation Point server to a client computer over the TCP/IP network. The client computer must have the Speech Client component installed in order to properly receive the message in speech form.

Configuring the AP Server for Text-to-Speech

You can send a text-to-speech message from the CA Automation Point server using either a Speak Notification Manager method or the ALARMSAY rules keyword.

The AP server sends the text-to-speech messages to the client computer through a common TCP port number. The default port number is set at 5700 on both the server and client sides. If there is a port number conflict with other applications, you can change the sender port number on the AP Server using Configuration Manager.

To change the sender port number

1. From Configuration Manager, choose Expert Interface, Automation, Automation Point Desktop Settings,
2. Change the value in the Text-to-Speech Port Number field.

You must configure the new port number on all your text-to-speech client computers. Remember that the listener port must match the sender port.

Configuring a Text-to-Speech Client

Two programs make up the Speech Client component. These are the AP Speak program and the AP Listener program. The AP Speak program interfaces with either the Microsoft Agent (if available) or the Microsoft SAPI interface and announces the received text using the installed text-to-speech engine. The AP Listener program intercepts and manages messages requested by the CA Automation Point server and invokes the AP Speak program to deliver one text-to-speech message at a time.

The AP Listener program is installed as part of the startup programs group. It should be running when you log in and remain running so that you do not miss any text-to-speech messages. The AP Listener icon displays in the Windows status area that is usually in the bottom right corner of your screen. To access help for CA Automation Point text-to-speech notification, right-click this icon and choose Help.

Neither the AP Speak program nor the AP Listener program requires additional configuration. The defaults should work under most circumstances. However, if you should need to change the default settings for the AP Listener program, you can do so.

The syntax for invoking AP Listener at the command line is:

```
aplisten [options]
```

where *options* are:

/p tcp-port-number

Specifies the TCP port number to listen on. The default port is 5700.

Note: Use caution when setting the port number. For the listener client and the AP server to communicate, this port (the listener port) must match the sender port on the AP server computer.

/c max-number-msgs

Specifies the maximum number of messages to put on hold in the message queue before any are rejected because the queue is full. The default maximum is 25.

/a

Indicates whether to automatically dismiss a message after it has been delivered as speech. The default requires the user to explicitly dismiss the Microsoft Agent character by double-clicking it.

Troubleshooting Text-to-Speech Notification

If text-to-speech notification is not working on your client computer, follow this troubleshooting guide to determine the reason for the problem.

Ask these questions:	Check these solutions:
Does your computer have multimedia capabilities?	Do you need a sound card and/or speakers? Have you installed a driver for the card? Have you turned on the power for the speaker?
Is the Listener running? Do you see its icon in your desktop status tray?	AP Listener application should be part of the windows startup group and running at all times. If so, you will see it in your desktop status tray. If you do not see it, choose Start, Programs, Startup, AP Listener to run the program.

Ask these questions:	Check these solutions:
Does the Listener listening port match the CA Automation Point sending port?	These two ports must match for the programs to communicate with each other. The default configuration port is 5700. (Configure the Speech Notification TCP/IP Port option at the CA Automation Point server side by using Configuration Manager. Configure the client side by configuring the AP Listener command line options that are documented earlier in this section.)

Two-Way Paging

Two-way paging is a notification method that can be used inside or outside Notification Manager. Notification Manager supports two-way paging through the method NMPAGE2WAY.

Two-way paging lets you:

- Receive confirmation of message delivery
- Respond directly to monitored events using the same messaging or two-way paging device
- Use the ADDRESS VOX PAGE2WAY keyword to:
 - Send messages to two-way messaging devices
 - Query the paging service provider for confirmation of message delivery
 - Receive responses sent from two-way messaging devices and match these responses to the initial page request
- Use the Notification Manager method NMPAGE2WAY to:
 - Send the message specified in the TELL parameter to a two-way paging device
 - Receive a response to the question specified by the ASK parameter from the two-way page recipient
- Use a dialog to:
 - View status information for outstanding two-way page requests
 - Find out whether two-way page requests have been delivered to the intended recipients
 - Find out whether the recipient of a two-way page request read or responded to it

CA Automation Point supports two-way paging through both of the following:

- Simple Network Paging Protocol (SNPP)
- Wireless Communications Transfer Protocol (WCTP)

Both SNPP and WCTP use TCP/IP to transport page requests over the Internet to the paging service provider. This eliminates the need for modems and phone lines to produce alphanumeric pages. If the WCTP protocol is used, the paging service must support Version 1.1 of this protocol. The VOX command processor, PAGE2WAY, lets you submit two-way page requests using either the SNPP or WCTP communication protocol. Before you can use the PAGE2WAY command, you must enable and configure the CA Automation Point 2-Way Paging feature, using dialogs in Configuration Manager.

Configure Two-Way Paging

To configure two-way paging

1. From the main window of Configuration Manager, choose Expert Interface, Notification Services, Paging, 2-Way Paging, 2-Way Paging Setup.

The 2-Way Paging Setup dialog displays.

2. Click the Enable 2-way paging checkbox.

This enables two-way paging inside the Notification Server, and allows you to start adding symbolic names of paging service providers.

3. Right-click either the SNPP or WCTP provider tree and choose Add from the menu.

For example, if you want to add a new paging service provider, and this paging service supports the SNPP communications protocol, right-click SNPP Providers and click Add.

Note: For you to use the WCTP protocol, the paging service must support Version 1.1 of the WCTP protocol.

A new item displays under either SNPP Providers or WCTP Providers, depending on which protocol type you have chosen.

4. Enter a unique name for the provider.
5. In the fields under Remote Connection Settings, configure your connection settings, using the dialog help to guide you.

The connection fields that are contained within this group appear according to the connection protocol selected for the current provider.

Note: The Paging Service Authentication button displays when you have added an item under the WCTP Provider section. If the application on the paging service provider's Web server (used to service WCTP Client page requests) requires user authentication, use this dialog to specify the user ID and password to be sent to this server application. For more information, see the HTML help for this dialog.

6. In the fields under Local Performance Settings, configure your performance settings, using the dialog help to guide you. You can customize this information to either provide additional information or control how the Notification Server processes two-way page requests.
7. If you are planning to connect to one or more paging services implementing the WCTP protocol, you may want to click Advanced to specify your site-specific Internet proxy settings. For more information about this dialog, see the dialog help.
8. When you have finished configuring your connection and performance settings, click OK.

The VOX Command Environment

The CA Automation Point VOX command environment performs the following primary functions:

- By issuing VOX commands, it enables your REXX programs to request notification services from one or more notification servers
- It returns command result information to the calling REXX program

A VOX command environment does not have to reside on the same workstation as a notification server. A VOX command environment can reside on any workstations from which REXX-based VOX commands originate. (A workstation running a VOX command environment is network-connected by PPQs to one or more notification servers.)

What Is a VOX Command?

A CA Automation Point VOX command is an executable statement that performs a specific voice processing or voice-related action. All VOX commands have required or optional operands, or both.

A notification server services all voice processing requests that are received from a VOX client through the VOX command environment.

Requirements for Issuing VOX Commands

To issue a VOX command, use an ADDRESS statement in your REXX program to access the CA Automation Point VOX command environment, as shown:

```
ADDRESS VOX "voxcommand [operands...]"
```

If the VOX client does not reside on the same workstation as a notification server, the client workstation must be network-connected (through PPQs) to one or more notification servers.

Usage Example

In response to a specified system problem, CA Automation Point can execute a problem-escalation REXX program. This REXX program will take the appropriate action according to the success or failure of each attempt to notify a system administrator.

Avoiding Errors in Your REXX Programs

When creating REXX programs that issue VOX commands, avoid naming your variables with names reserved for VOX commands. For a list of VOX command names to avoid when naming your variables, see the section [VOX Command Summary](#) (see page 238) in this chapter.

The REXX variable `VOX.cmdseq` is set to the command sequence number assigned to the VOX command by the VOX client. Use this sequence number to identify and track the progress of the command on the CA Automation Point Notification Services Engine console.

VOX Command Summary

This section discusses the following types of commands:

- Notification Manager database maintenance
- Notification
- Voice processing
- Utility
- Commands that perform I/O operations

For command syntax, see the *Command and Keyword Reference Guide*.

Notification Manager Database Maintenance Commands

Use the following commands to manage the Notification Manager database.

ALTERENTITY

Alters the characteristics of an entity.

ALTERMETHOD

Alters the characteristics of a method.

ALTERPARM

Alters the characteristics of a parameter.

ALERTIME

Alters the characteristics of a time block.

CREATEENTITY

Creates an entity.

CREATELOGIN

Creates a login.

CREATEMETHOD

Creates a method.

CREATEPARAM

Creates a parameter.

CREATETIME

Creates a time block.

DESTROYENTITY

Destroys an existing entity.

DESTROYLOGIN

Destroys an existing login.

DESTROYMETHOD

Destroys an existing method.

DESTROYPARAM

Destroys an existing parameter.

DESTROYTIME

Destroys an existing time block.

EPWCHECK

Checks the password for an entity.

LISTENTITY

Lists all the data for one or more entities (except the password).

LISTFIND

Lists the time blocks that are active for one or more entities at a given date and time. The list is a tree that can contain any level of nesting.

LISTFORWARD

Lists all the time blocks that forward to a given entity (regardless of the time and date). The list is a tree that can contain any level of nesting.

LISTLOGIN

Lists all the data for one or more logins.

LISTMETHOD

Lists all the data for one or more methods.

LISTPARAM

Lists all of the data for one or more parameters.

LISTPERGRPS

Lists all personal groups for a specified contact.

LISTTIME

Lists all the data for one or more time blocks.

NMDBMERGE

Merges a previously exported copy of a database into the current Notification Manager database.

NMEXPORT

Creates a copy of the current Notification Manager database in a format that can be easily moved to another system.

NMIADDCALLER

Adds an entity to the list of entities that are allowed to call in on an item.

NMIANSWER

Sets or retrieves the answer for an item in the Notification Manager database.

NMICHECKCALLER

Checks to see whether an entity has been permitted to listen to/answer this item (by a call to NMIADDCALLER).

NMIGETITEM

Retrieves the ask text and tell text for an item from the Notification Manager database.

NMILISTANSWERS

Retrieves the answers that are specified using the NMFIND ASK parameter (if any) for a notification request item stored in the Notification Manager database.

NMILISTCALLERS

Retrieves status information about every notification attempt that is made during the processing of a notification request item stored in the Notification Manager database.

NMILISTITEMS

Retrieves detailed status information about notification request items stored in the Notification Manager database

NMIMPORT

Imports a copy of a Notification Manager database.

Notification Commands

Use the following commands for notification tasks.

PAGE

Issues an alphanumeric page to designated personnel using a modem-based alphanumeric paging service.

PAGE2WAY

Issues an alphanumeric page to designated personnel using an Internet-based alphanumeric paging service.

SENDMAIL

Generates an e-mail message to designated personnel.

Voice Processing Commands

Use the following commands for voice processing tasks.

ANSWER

Waits for an incoming telephone call on any available voice channel, any voice channel within a group, or a specific voice channel only.

ANSWERPLAY

Waits for an incoming telephone call (in the same way as the ANSWER command) and plays one or more prerecorded voice messages after answering.

CALL

Initiates a telephone call using an open, available voice channel.

CALLPLAY

Initiates a telephone call on an available voice channel and plays one or more prerecorded voice messages.

CLEAR

Clears the digit buffer of a voice channel or its call progress analysis (CPA) parameters.

GETCHANNEL

Serializes I/O activity on a voice channel by marking an available channel as in-use. The calling REXX program then has exclusive access to the channel.

GETCHANNELNUM

Identifies the physical voice channel number associated with a channel handle.

GETDIGITS

Retrieves tone digits-such as menu selections or access codes that a remote party enters from a telephone keypad in response to a voice prompt-from the digit buffer of a voice channel.

GETGROUP

Returns a text string containing a group name and a list of all physical voice channel numbers associated with the group.

GETSTATUS

Returns the current status of a voice channel.

GETSYSNAMES

Retrieves the system names of all connected notification servers in a distributed environment.

LOAD

Loads a voice file or voice word library index file into main memory for faster access.

PLAY

Plays a prerecorded voice message through a specified voice channel.

PLAYGETDIGITS

Plays a prerecorded voice message through a specified voice channel, and then retrieves tone digits that the remote party enters from the telephone keypad.

RECORDFILE

Records a voice message from the remote party and stores it in a disk file.

RELEASECHANNEL

Resets a voice channel marked as in-use by the GETCHANNEL command, making the channel available to other REXX programs issuing a GETCHANNEL command. Used with the GETCHANNEL command to serialize I/O activity on a channel.

SENDTONES

Sends tone digits through an already-open voice channel, useful for sending additional tones after CA Automation Point has successfully called and connected to the remote party.

SETGROUP

Associates a group name with one or more voice channels.

SETHOOK

Sets the hook state of a voice channel to on-hook or off-hook, useful in special circumstances such as multiple CALL operations within a single REXX program.

SETVOLUME

Adjusts the volume for current and subsequent PLAY operations on a specified voice channel.

STOP

Terminates an active I/O operation on a voice channel.

VERIFYUSER

Checks the validity of the user ID and password combination of the remote party.

WINK

Sends a brief "handshaking" protocol signal through a voice channel.

Utility Commands

The following commands are ADDRESS VOX utility commands.

GETTAPIDEVICELIST

Lists all the TAPI devices that are installed under Windows.

SETENGINE

Allows you to modify notification server settings.

SETMSGSTREAM

Duplicates the message stream from CA Automation Point to another queue that is accessible through PPQs, local or remote.

SETTRACE

Start and stop trace logging.

SLEEP

Causes the issuing REXX EXEC to enter a system sleep state.

STARTREXX

Starts another REXX program.

VER

Provides the version number and configuration information of the CA Automation Point notification services at your site

VOX Commands that Perform I/O Operations

I/O activity occurs when a VOX command gathers data from a voice channel (for example, by collecting tone digits entered by the remote party) or sends data out through a voice channel (for example, by placing an outgoing call).

The following VOX commands perform I/O operations:

Input Operations	Output Operations	Input and Output Operations
ANSWER	CALL	ANSWERPLAY
GETDIGITS	CALLPLAY	PAGE
GETSYSNAMES	PLAY	PLAYGETDIGITS
RECORDFILE	SENDTONES	
	WINK	

Example VOX Environment Applications

This section presents REXX programs that employ voice commands. The programs illustrate the following application types:

- Outbound voice notification with input from caller
- Outbound recording and voice message delivery
- Inbound application with input from caller

Although the REXX code shown may vary slightly from the code on your distribution media, you can use the examples as a basis for writing your own voice processing applications.

Each example REXX program is presented in the same format. Each example contains these elements in the order listed:

- A statement of the program's purpose, indicating what the program does and the primary operations that it illustrates
- A listing of the ADDRESS VOX commands used in the code
- A listing of the REXX source code
- A description of how the program works

Outbound Voice Notification with Input from Caller

The following REXX program implements an outbound voice notification with input from caller application. Within the UserApplication portion of the code, you can add specific logic that implements feedback from the connecting caller. Examples of UserApplication include, but are not limited to:

- Implementation of problem notification. This is done by informing the connecting caller of problems currently assigned and obtaining feedback whether the assigned problem has been resolved or is still outstanding.
- Implementation of message delivery. This is done by playing all messages that are to be delivered to the connecting caller.

VOX Commands Used

When coding a VOX environment application, use the following ADDRESS VOX commands:

CALLPLAY

Initiates the phone call and automatically play a voice file or words from a word library as soon as the caller picks up the phone. The advantage of using CALLPLAY instead of CALL then PLAY is that the VOX environment automatically loads the voice message segment to play prior to making the phone call, thereby improving performance during initiation of the call.

GETCHANNEL

Obtains an available channel using a hard-coded channel number or retrieving a channel from a group.

GETDIGITS

Obtains user's DTMF tones, assuming the voice message segment played with CALLPLAY has prompted the user appropriately.

PLAYGETDIGITS

Obtains subsequent DTMF tones by first prompting the user with a voice message segment. PLAYGETDIGITS is recommended over using PLAY then GETDIGITS, because it facilitates an improved reading and maintenance of your application.

RELEASECHANNEL

Releases the channel used by the phone call, making it available again.

VERIFYUSER

Verifies the caller as a valid user of the application.

REXX Code Listing

The following sample REXX code illustrates an outbound notification application:

```
/* REXX */
ChannelInUse = 'N'      /* Channel not obtained yet */
/* Setup Error Handler in the event */
/* program encounters Ctrl-Break */
SIGNAL ON HALT NAME DOEXIT

...

/* Logic to determine which phone number to call */
/* goes here. */

/* REXX variable PhoneNumber is set by user's */
/* application...it is hard-coded for */
/* illustration purposes. */

PhoneNumber = '9,1800-555-1212'

Call GetFreeChannel

Call MakeOutboundCall PhoneNumber

Call UserApplication

Call ReleaseTheChannel

exit

GetFreeChannel:
/*-----*/
/* Obtain an available voice channel */
/* Channel can be obtained by channel number */
/* or by Group, in this example, a Group of ALL */
/* is used..... */
/*-----*/
Address VOX "GETCHANNEL Group(All) Prefix(Handle)"
If (rc <> 0) Then Do
    say 'Unable to obtain an available channel. RC='rc
    exit 16
End
ChannelInUse = 'Y' /* remember we now have a channel */
return
```

```

MakeOutboundCall:
Arg OutboundNumber
/*-----*/
/* Call number and play preliminary greeting */
/* For improved performance, use VOX CALLPLAY */
/* since VOX message to be played is loaded into */
/* memory prior to making phone call. */
/*-----*/
OutboundNumber = STRIP(OutboundNumber)

GreetingToPlay = 'greeting.vox getuserid.vox'

Address VOX "CALLPLAY Channel("Handle") ",
           "ToneString("OutboundNumber") ",
           "File("GreetingToPlay") FILETYPE(NONINDEX)"
If (rc <> 0) Then Do
    say 'Problem calling number: 'OutboundNumber ' RC='rc
    return 16
End

/* Before returning to main line code, you may want to */
/* verify connecting caller, you can use VOX GETDIGITS, */
/* PLAYGETDIGITS and VERIFYUSER commands... */

IdLen = 6           /* Assume Userid is length 6 */
PinLen = 6         /* Assume Pin # is length 6 */
Address VOX "GETDIGITS Channel("Handle") ",
           "Count("IdLen") Prefix(CallerId)"

If (rc <> 0) Then Do
    say 'Problem obtaining caller userid RC='rc
    return 16
End

/* Get caller's pin number. */

Address VOX "PLAYGETDIGITS Channel("Handle") ",
           "FILE(GetPin.Vox) FILETYPE(NONINDEX)",
           "Count("PinLen") Prefix(CallerPin)"
If (rc <> 0) Then Do
    say 'Problem obtaining caller pin number RC='rc
    return 16
End

```

```
/* Verify the connected phone call */

Address VOX "VERIFYUSER Userid("CallerId") Pin("CallerPin)"
If (rc <> 0) Then Do
    say 'Unable to Verify Caller RC='rc
    return 16
End

/* Caller is OK, now we can return to main line */
return 0

UserApplication:
/*-----*/
/* User specific application can go here */
/*-----*/
    ...
    ...
return

ReleaseTheChannel:
/*-----*/
/* Release the channel obtained, this is done */
/* by calling VOX RELEASECHANNEL command */
/*-----*/
If (ChannelInUse = 'Y') Then Do
    Address VOX "RELEASECHANNEL Channel("Handle")
    If (rc <> 0) Then Do
        say 'Unable to release channel RC='rc
    End
    Else
        ChannelInUse = 'N'
End
return

DoExit:
/*-----*/
/* Perform any necessary cleanup such as releasing */
/* the channel by calling VOX RELEASECHANNEL command */
/*-----*/
Call ReleaseTheChannel
/* perform other application specific cleanup */
    ....
    ...
/* terminate application */
exit
return
```


How It Works

Typically, an outbound voice notification with input from a caller application involves the VOX environment making a phone call and retrieving user data from the connecting caller (in the form of DTMF tones). Depending on the data received, the application then takes any appropriate action or responds based upon the user's request.

An example of an outbound notification program could be calling support personnel and requesting what action should be taken when a product at a specific site needs to be recycled.

Outbound Recording with Voice Message Delivery

The following REXX program implements an outbound recording with voice message delivery application. Within the UserApplication portion of the code, you can add specific logic. Examples of UserApplication include, but are not limited to:

- Implementation of status update
- Implementation of message forwarding

VOX Commands Used

When coding such a VOX environment application, use the following ADDRESS VOX commands:

CALLPLAY

Initiates a phone call and automatically play a voice file or words from a word library when a caller picks up the phone. The advantage of using CALLPLAY instead of CALL then PLAY is that the VOX environment automatically loads the voice message segment to play prior to making the call, improving performance during initiation of the call.

GETCHANNEL

Obtains an available channel using a hard-coded channel number or retrieving a channel from a group.

GETDIGITS

Obtains user's user ID (in DTMF tones), assuming the voice message segment played with CALLPLAY has prompted the user appropriately.

PLAY

Used to notify the caller that recording is about to begin.

PLAYGETDIGITS

Plays voice messages with a prompt to retrieve the user's response to the prompt. We recommend you use PLAYGETDIGITS instead of PLAY then GETDIGITS because it facilitates an improved reading and maintenance of your application.

RECORDFILE

Records the caller's message.

RELEASECHANNEL

Releases the channel used by the phone call, making it available again.

VERIFYUSER

Verifies the connecting caller.

REXX Code Listing

The following sample REXX code illustrates an outbound recording with voice message delivery application:

```

/* REXX */
ChannelInUse = 'N'      /* Channel not obtained yet */
/* Setup Error Handler in the event program encounters Ctrl-Break */
SIGNAL ON HALT NAME DOEXIT

      ...

/* Logic to determine which phone number to call */
/* goes here. */
/* REXX variable PhoneNumber is set by user's */
/* application...it is hardcoded for */
/* illustration purposes. */
PhoneNumber = '9,1800-555-1212'
Call GetFreeChannel
Call MakeOutboundCall PhoneNumber
Call UserApplication
Call ReleaseTheChannel
exit

GetFreeChannel:
/*-----*/
/* Obtain an available voice channel */
/* Channel can be obtained by channel number */
/* or by Group, in this example, a Group of ALL */
/* is used..... */
/*-----*/
Address VOX "GETCHANNEL Group(All) Prefix(Handle)"
If (rc <> 0) Then Do
    say 'Unable to obtain an available channel. RC='rc
    return 16
End
ChannelInUse = 'Y' /* remember we now have a channel */
return

MakeOutboundCall:
Arg OutboundNumber
/*-----*/
/* Call number and play preliminary greeting */
/* For improved performance, use VOX CALLPLAY */
/* since VOX message to be played is loaded into */
/* memory prior to making phone call. */
/*-----*/
OutboundNumber = STRIP(OutboundNumber)
GreetingToPlay = 'greeting.vox getuserid.vox'

Address VOX "CALLPLAY Channel("Handle") ",
           "ToneString("OutboundNumber") ",
           "File("GreetingToPlay") FILETYPE(NONINDEX)"

```

```
If (rc <> 0) Then Do
    say 'Problem calling number: 'OutboundNumber ' RC='rc
    return 16
End

/* Before returning to main line code, you may want to */
/* verify connecting caller, you can use VOX GETDIGITS,*/
/* PLAYGETDIGITS and VERIFYUSER commands...          */

IdLen = 6          /* Assume Userid is length 6   */
PinLen = 6         /* Assume Pin # is length 6   */
Address VOX "GETDIGITS Channel("Handle") ",
            "Count("IdLen") Prefix(CallerId)"

If (rc <> 0) Then Do
    say 'Problem obtaining caller userid RC='rc
    return 16
End

/* Get caller's pin number. */

Address VOX "PLAYGETDIGITS CHANNEL("Handle") ",
            "FILE(GetPin.Vox) FILETYPE(NONINDEX)",
            "COUNT("PinLen") PREFIX(CallerPin)"
If (rc <> 0) Then Do
    say 'Problem obtaining caller pin number RC='rc
    return 16
End

/* Verify the connected phone call */

Address VOX "VERIFYUSER Userid("CallerId") Pin("CallerPin)"
If (rc <> 0) Then Do
    say 'Unable to Verify Caller RC='rc
    return 16
End

/* Caller is OK, now we can return to main line */
return 0

UserApplication:
/*-----*/
/* User specific application can go here */
/*-----*/
StatusMsg = 'SYS1STAT.VOX' /* actual system status VOX file */
TmpMsg = '$STAT.VOX' /* temporary hold for a VOX file */
PromptForRecord = 'StrRecd.Vox' /* Vox file to prompt */
/* for recording */
```

```

PromptUserOptions = 'UserOpts.Vox' /* Vox file to prompt */
                                /* user with options */
SupportMgrExt = '1555' /* Support Manager's Extension */
MgrMsg = 'STATCHG.VOX 'StatusMsg /* Vox files to play */
                                /* to support mgr */
    Msg = 'Problem has been detected by application '
ReRecord:
/* Prompt User before beginning recording session */
Address VOX "PLAY Channel("Handle") ",
            "File("PromptForRecord") FILETYPE(NONINDEX) "

Address VOX "RECORDFILE Channel("Handle") ",
            "File("TmpMsg") OverWrite(YES) ",
            "Interrupt(YES) "

If (rc <> 0) Then Do
    /* Tell connecting caller a problem has been encountered */
    say Msg ' RECORDFILE RC: ' rc
    Address VOX "PLAY Channel("Handle") ",
                "FILE(VOXF_A) FILETYPE(WORDLIB) VAR(Msg)"
    return 16
End
PromptAgain:
UserSelection = ''
/* Present User with Options for the recording */
Address VOX "PLAYGETDIGITS Channel("Handle") ",
            "File("PromptUserOptions") FILETYPE(NONINDEX) ",
            "Count(1) Prefix(UserSelection) "

If (rc <> 0) Then Do
    /* Tell connecting caller a problem has been encountered */
    say Msg ' PLAYGETDIGITS RC: ' rc
    Address VOX "PLAY Channel("Handle") ",
                "FILE(VOXF_A) FILETYPE(WORDLIB) VAR(Msg)"
    return 16
End

/* Process User's Response */
/* 1 - Accept new recording and terminate */
/* 2 - Rerecord the message */
/* 3 - Terminate without saving recording */
Select
    When (UserSelection == 1) Then Do
        /* Update the system status message */
        'COPY 'TmpMsg StatusMsg
    End
    When (UserSelection == 2) Then Do
        /* Prompt user to rerecord the message */
        Signal ReRecord
    End
    When (UserSelection == 3) Then Do

```

```

        /* don't update the system status message */
        nop
    End
    Otherwise
        /* User Specified unrecognized option */
        Signal PromptAgain
    End
    Msg = 'Goodbye'
    Address VOX "PLAY Channel("Handle") ",
        "FILE(VOXF_A) FILETYPE(WORDLIB) VAR(Msg)"

    /* Inform Support Manager that System Status has been updated */
    /* So, hangup this call then call the support manager */
    Address VOX "SETHOOK CHANNEL("Handle") HOOKSTATE(ONHOOK)"
    Address VOX "CALLPLAY CHANNEL("Handle") TONESTRING("SupportMgrExt)" ,
        "FILE("MgrMsg") FILETYPE(NONINDEX)"

    If (rc <> 0) Then Do
        /* Unable to call support manager .... */
        say Msg ' CALLPLAY RC:' rc
        return 16
    End
    return 0

ReleaseTheChannel:
/*-----*/
/* Release the channel obtained, this is done */
/* by calling VOX RELEASECHANNEL command */
/*-----*/
    If (ChannelInUse = 'Y') Then Do
        Address VOX "RELEASECHANNEL Channel("Handle")
        If (rc <> 0) Then Do
            say 'Unable to release channel RC='rc
        End
    Else
        ChannelInUse = 'N'
    End
    return
DoExit:
/*-----*/
/* Perform any necessary cleanup such as releasing */
/* the channel by calling VOX RELEASECHANNEL command */
/*-----*/
    Call ReleaseTheChannel
    /* perform other application specific cleanup */
        ....
        ...
    /* terminate application */
    exit

```

return

How it Works

Typically, an outbound recording with voice message delivery application involves the VOX environment making a phone call and retrieving user security information prior to allowing a user to record a voice message. Depending on the application, voice messages can be deleted, recorded again, or forwarded to others.

An example of an outbound recording with voice message delivery could be when maintaining system status on a daily basis, where an individual within a support group could be responsible for updating the system status every hour.

Inbound Application with Input from Caller

The following REXX program implements an inbound with input from caller application. Within the UserApplication portion of the code, you can add specific logic that implements feedback from the connecting caller. Examples of UserApplication include, but are not limited to:

- Inbound status checking, where any caller can call to retrieve latest status
- A help desk application

VOX Commands Used

When coding such a VOX environment application, use the following ADDRESS VOX commands:

ANSWERPLAY

Begins the application in an ANSWER state. When a call comes in, the application can play a voice file or words from a word library. The advantage of using ANSWERPLAY instead of ANSWER, then PLAY, is that the VOX environment automatically loads the voice message segment to play prior to answering the phone call, thereby improving performance during the initial connection.

GETCHANNEL

Obtains an available channel using a hardcoded channel number or retrieving a channel from a group (see the note after the table).

GETDIGITS

Obtains a user's user ID (in DTMF tones), assuming the voice message segment played with ANSWERPLAY has prompted the user appropriately.

PLAYGETDIGITS

Plays voice messages with a prompt to retrieve the user's response to the prompt. PLAYGETDIGITS is recommended instead of PLAY then GETDIGITS because it facilitates an improved reading and maintenance of your application.

RELEASECHANNEL

Releases the channel used by the phone call, placing the channel in an available state for the next caller.

VERIFYUSER

Verifies the caller as a valid user of the application.

Note: When coding an inbound application, you can eliminate the GETCHANNEL command by using the ANSWERPLAY command with the GROUP keyword (waiting for a phone call from a group of channels); when a call comes in, the ANSWERPLAY command returns the appropriate channel handle to be used on subsequent ADDRESS VOX commands.

REXX Code Listing

The following sample REXX code illustrates a sample inbound status application:

```
/* REXX */
ChannelInUse = 'N'          /* Channel not obtained yet */
Terminate = 'N'
/* Setup Error Handler in event program encounters Ctrl-Break */
SIGNAL ON HALT NAME DOEXIT

0 ...
Call GetFreeChannel

Do While (Terminate <> 'Y')
  Call WaitForACall
  If (result <> 0) Then
    Terminate = 'Y'
  Else
    Do
      Call UserApplication
      If (result <> 0) Then
        Terminate = 'Y'
    End
End
Call ReleaseTheChannel

exit
```



```

GetFreeChannel:
/*-----*/
/* Obtain an available voice channel */
/* Channel can be obtained by channel number */
/* or by Group, in this example, a Group of ALL */
/* is used..... */
/*-----*/
Address VOX "GETCHANNEL Group(All) Prefix(Handle)"
If (rc <> 0) Then Do
    say 'Unable to obtain an available channel. RC='rc
    exit 16
End
ChannelInUse = 'Y' /* remember we now have a channel */
return

WaitForACall:
/*-----*/
/* Waiting for a call on a specific channel before returning */
/* For improved performance, use VOX ANSWERPLAY */
/* since VOX message to be played is loaded into */
/* memory prior to making answering phone call. */
/*-----*/
GreetingToPlay = 'greeting.vox getuserid.vox'

Address VOX "ANSWERPLAY Channel("Handle") ",
           "File("GreetingToPlay") FILETYPE(NONINDEX)"
If (rc <> 0) Then Do
    say 'Problem answering call RC='rc
    return 16
End

0/* Before returning to main line code, you may want to */
/* verify connecting caller, you can use VOX GETDIGITS,*/
/* PLAYGETDIGITS and VERIFYUSER commands... */

IdLen = 6 /* Assume Userid is length 6 */
PinLen = 6 /* Assume Pin # is length 6 */
Address VOX "GETDIGITS Channel("Handle") ",
           "Count("IdLen") Prefix(CallerId)"

If (rc <> 0) Then Do
    say 'Problem obtaining caller userid RC='rc
    return 16
End

```

```
/* Get caller's pin number. */

Address VOX "PLAYGETDIGITS Channel("Handle") ",
            "FILE(GetPin.Vox) FILETYPE(NONINDEX)",
            "Count("PinLen") Prefix(CallerPin)"
If (rc <> 0) Then Do
    say 'Problem obtaining caller pin number RC=' rc
    return 16
End

/* Verify the connected phone call */

Address VOX "VERIFYUSER Userid("CallerId") Pin("CallerPin)"
If (rc <> 0) Then Do
    say 'Unable to Verify Caller RC=' rc
    return 16
End

/* Caller is OK, now we can return to main line */
return 0

UserApplication:
/*-----*/
/* User specific application can go here */
/*-----*/
/* Can use VOX PLAY command to play */
/* stock quotes, system status etc., */
    ...
    ...
/* Make sure phone is back on hook before returning to mainline */
Address VOX "SETHOOK Channel("Handle") HOOKSTATE(ONHOOK)"
If (rc <> 0) Then Do
    say 'Unable to put phone back onhook RC = 'rc
    return 16
End
```

```

return 0

ReleaseTheChannel:
/*-----*/
/* Release the channel obtained, this is done */
/* by calling VOX RELEASECHANNEL command */
/*-----*/
If (ChannelInUse = 'Y') Then Do
  Address VOX "RELEASECHANNEL Channel("Handle")
  If (rc <> 0) Then Do
    say 'Unable to release channel RC='rc
  End
Else
  ChannelInUse = 'N'
End
return

DoExit:
/*-----*/
/* Perform any necessary cleanup such as releasing the */
/* channel by calling VOX RELEASECHANNEL command */
/*-----*/
Call ReleaseTheChannel
/* perform other application specific cleanup */
....
...
/* terminate application */
exit
return

```

How it Works

Typically, an inbound notification application remains in an ANSWER state until a call is received. Once the VOX environment receives an inbound phone call, the application can provide status updates, acknowledge the receipt of a message, or provide various MIS services.

Chapter 11: Using Notification Manager

This section contains the following topics:

- [Notification Manager Overview](#) (see page 261)
- [Notification Manager Database](#) (see page 265)
- [Notification Website Application](#) (see page 266)
- [Optional Configuration Tasks](#) (see page 268)
- [Notification Manager Concepts](#) (see page 275)
- [Creating Your Own Invocation Programs](#) (see page 297)
- [How You Populate the Notification Manager Database](#) (see page 304)
- [How You Use the Notification Website](#) (see page 304)
- [Troubleshooting the Notification Website](#) (see page 307)
- [Secure the Notification Website](#) (see page 309)
- [Understanding Notification Manager Log Files](#) (see page 327)
- [JolBeep Panel Emulation](#) (see page 330)

Notification Manager Overview

The Notification Manager component can help you implement automated notification policies and procedures in your operations. Notification procedures can include the following:

- Numeric paging
- Alphanumeric paging
- Two-way paging
- Networked text-to-speech messages
- Email
- Voice notification
- Solicitation of input through DTMF (telephone keypad) tones
- Pre-recorded or dynamically built messages
- Recording and forwarding of messages
- Various combinations of these methods.

At its highest level, Notification Manager exposes a single program interface for use in notification. This interface is facilitated through the NMFIND.REX REXX program. By calling the NMFIND program with the appropriate parameters from CA Automation Point rules or your own REXX programs, you can perform notification functions.

Notification Strategy

With Notification Manager, you can implement a sophisticated notification strategy without any REXX programming. Your strategy can include:

- Choosing the method (for example, voice, pager) used to notify people based on the time or day
- Choosing the behavior of the method (for example, number of times to let the phone ring, number of times to retry on busy) based on the individual being notified or on the time or day
- Providing multiple (for example, backup) methods of notifying individuals
- Assigning responsibilities based on the time or day
- Assigning backup responsibilities based on the time or day
- Providing for temporary or periodic reassignment of responsibilities
- Grouping people by common job function or any other criteria
- Notifying just one or all the people in a group about a situation
- Automatic escalation, or forwarding, when a person or group cannot be notified

Objects Used by Notification Manager

The basic purpose of Notification Manager is to find and notify someone using specified methods, tell them something, and (optionally) ask them something. You can take advantage of this functionality directly from rules, from within a REXX script that you write, from a command prompt, or by using the Notification Website. Notification Manager uses five different kinds of objects to carry out this function:

Login

Identifies you to Notification Manager that you are who you claim to be before any tasks can be carried out.

Contact

Specifies the person or group that you want to modify.

Method

Determines the type of method to be used to notify a particular contact. Some people have numeric pagers, some alphanumeric pagers, and some no pager at all. Notification Manager allows you to use different methods of notification depending on the person you are contacting and on the current time and day.

Time block

Describe how to reach a contact during a particular block of time. For example, when a person is scheduled to be in the office you may want to send them a voice message, but when they are on the road, you may want to use a pager to contact them.

Parameter

Establish and define method actions. For example, when you use a pager, the phone number of the pager service, the pager number, and the maximum length this pager service allows for messages are all parameters.

Note: Any messages generated by notification (NMFIND) requests are displayed in the CA Automation Point Notification Messages window.

Notification Manager Scenarios

Here are some examples of Notification Manager's capabilities.

Example 1:

Suppose you determine that when a particular JES is having difficulties, you need to notify the lead JES systems programmer, Jim Smith. The CA Automation Point rule that trapped the error message from JES would contain this clause:

```
REXX(NMFIND PERSON(JIM SMITH) TELL('JES is down'))
```

Notification Manager uses its database technology (which is based on a relational database) to determine which communications method it should use to contact Jim Smith based on the time and day. Notification Manager proceeds to contact Jim Smith and relay the message according to the following:

- If the notification method was CA Automation Point voice technology, Jim receives a phone call at the phone number pointed to by his notification schedule.
- If the notification method was a numeric pager, Notification Manager pages Jim with the numeric message consisting of the phone number that he needs to call and an ID number authorizing him to receive the voice message.
- If the appropriate notification method was an alphanumeric pager, the message displays on Jim's pager with a phone number and ID that he can use to get any information that was not sent by his pager. (For example, the message to be sent may be longer than his paging service supports.)
- If the notification method was CA Automation Point email technology, Jim receives an email message at the email address pointed to by his notification schedule.

Important! You can support longer text notification with the email method than you can support with the voice and paging methods.

Example 2:

Extending the previous example, suppose you have written a REXX program that obtains control whenever JES is down and you have determined that you do not know how to handle the situation. Thus, you want to allow Jim to specify what should be done about JES being down. You can code a call to Notification Manager within your REXX program as follows:

```
CALL NMFIND.REX "PERSON(JIM SMITH) TELL('JES is down')",  
  "ASK('What should I do', 'WARM START', 'COLD START')"
```

Your program returns a value of 1 from NMFIND if Jim wants it to warm-start JES (because WARM START is the first answer) and a value of 2 if Jim wants it to cold-start JES (because COLD START is the second answer).

Example 3:

Alternately, you may want to code REXX programs that handle various situations, and then code your automation so that Notification Manager invokes those programs based on the response it gets from the person it calls. For example, you could code the following invocation of Notification Manager in your rules file:

```
REXX(NMFIND PERSON(JIM SMITH) TELL('JES IS DOWN')  
  ASK('What should I do,  
      WARM START :: JESSTART.REX WARM,  
      COLD START :: JESSTART.REX COLD,  
      IPL THE SYSTEM :: IPLSYS.REX'))
```

Note: The previous example is split across lines for presentation on the page. However, when you enter this code, it must be on a single line.

If Jim asks for a warm start, Notification Manager will run the JESSTART REXX program with a parameter of WARM. If Jim asks for a cold start, Notification Manager will run the JESSTART REXX program with a parameter of COLD. If Jim asks for a system IPL, Notification Manager will run the IPLSYS REXX program.

Notification Manager also supports group definition and notification. For example, if you define a group called CICS_SYS_PROGS, you could replace the PERSON(JIM SMITH) in the previous examples with GROUP(CICS_SYS_PROGS). Instead of trying to contact just Jim Smith, Notification Manager systematically attempts to contact each member of the CICS_SYS_PROGS group until someone is successfully contacted. You can also tell Notification Manager to notify all members of the group by defining the group as a broadcast group.

Notification Manager Database

This section describes the steps you need to perform to define the Notification Manager database for use at your site.

Configuration Steps

Important! We *highly* recommend that you use the Wizard Interface to initially configure the CA Automation Point essential facilities and services. After you have used the Wizard Interface for initial configuration, you can expediently modify a specific component by using the Expert Interface. See the chapter "[Configuring CA Automation Point](#) (see page 19)" for more information on the Wizard and Expert interfaces.

To configure Notification Manager.

1. From Configuration Manager, choose Expert Interface, Notification Services, Notification Manager, NM Setup.
The NM Setup dialog displays.
2. Follow the instructions on the dialog to establish your connection to a Notification Manager database.

The machine that you select on this dialog as the Notification Manager database server (either local or remote) must already have a supported version of Microsoft SQL Server installed. The selected Microsoft SQL Server instance must also have the TCP/IP database protocol enabled. You can use the SQL Server Configuration Manager application that ships with Microsoft SQL Server to view the enabled status of this protocol. If the selected Notification Manager database server is remote, you must first ensure that the Notification Manager database has been installed on this remote machine before completing this configuration. A Notification Manager database cannot be installed or upgraded remotely.

After you have configured the Notification Manager database, you can enable Web access to Notification Manager using the NM Website dialog. This dialog allows you to specify the required configuration information to make Notification Manager accessible from a Web browser.

Note: For more information, see the section [Notification Website Application](#) (see page 266).

To configure security constraints that apply to the use of Notification Manager from the Web, you can use the NM Security dialog. This dialog allows you to specify the type of user authentication, the type of security, and the passwords for the built-in Notification Manager user accounts.

Notification Website Application

The Notification Website comprises two parts:

- The Notification Manager web application, which executes inside the JSP/Servlet environment that is selected to host the Notification Website. The Notification Manager web application is responsible for creating the HTML pages that are used to display the gathered Notification Manager data.
- The Notification Manager Gateway component, which is a service that is installed on the CA Automation Point server machine. This component is responsible for retrieving data from the Notification Manager database that is initially requested by the Notification Manager web application.

Note: Before you can begin to configure the Notification Manager Website, you must first configure Notification Manager. You do this on the NM Setup dialog.

The Notification Manager Website can be set up in two different ways:

- On the same machine where you installed the CA Automation Point Server
- On another machine hosting a Java Servlet environment

These situations are described in detail in the following sections.

Install the Notification Website Application on the Same Machine as the CA Automation Point Server

The Notification Website is included when you choose to install the Server Features from the CA Automation Point installation program.

To enable the Notification Website Application.

From the main window of Configuration Manager, choose Expert Interface, Notification Services, Notification Manager, NM Website.

The NM Website dialog displays.

1. Check the Enable Access to Notification Website check box.
2. The Notification Website application must run in a JSP/Servlet environment. Therefore provide your own JSP/Servlet environment or install a redistributed version of the Apache Tomcat Server. The required version of the JRE must be installed on your machine.
3. The Java/Tomcat Installation dialog displays if either a Java Runtime Environment (JRE) or the Apache Tomcat JSP/Servlet software is needed on your computer.

How You Set Up to Use the Apache Tomcat Server

To use the Apache Tomcat server that is redistributed with CA Automation Point as the web server for the Notification Website.

Follow these steps:

1. Check the Use Local Java Servlet Environment check box.
2. Choose an appropriate context name for the Notification Manager website application and select a previously unused TCP/IP port number for the Notification Manager Gateway service.
3. Confirm your configuration choices by clicking the OK button. The Java/Tomcat Installation dialog displays. This dialog guides you through the process of installing the Apache Tomcat software and Java Runtime Environment.

The Notification Website configuration step is complete. Continue with the [Optional Configuration Tasks](#) (see page 268).

How You Provide Your Own JSP/Servlet Environment

When providing your own JSP/Servlet environment, be sure to leave the Use Local Java Servlet Environment check box unchecked.

The Notification Manager Gateway service requires a compatible version of the Java Runtime Environment (JRE) for the Notification Manager web application to have access to the Notification Manager database. You can also select a previously unused TCP/IP port number for the Notification Manager Gateway service.

After you have enabled the Notification Website, manually deploy the Notification Manager web application into this JSP/Servlet environment. The Notification Manager web application is packaged into a web application archive file (a WAR file) called `caapnfy.war`. This WAR file resides in the CA Automation Point CLASSES directory (for example, `installDir\classes\caapnfy.war`). See your JSP/Servlet environment documentation for instructions on how to deploy this application. After both enabling the Notification Website functionality and manually deploying the Notification Manager web application, see the section [Optional Configuration Tasks](#) (see page 268).

Confirm your configuration choices. When the Java/Tomcat Installation dialog, displays it guides you through the process of installing a compatible version of the Java Runtime Environment.

Install the Notification Website Application on a Non-CA Automation Point Machine

If you decide to run the Notification Website application on a machine other than the CA Automation Point server machine, you must manually deploy the web application archive file (.WAR file) into your JSP/Servlet engine. The Notification Website application is packaged in the `caapnfy.war` file, which can be found in the `installDir\classes` directory on the CA Automation Point server machine. Refer to your JSP/Servlet environment documentation for instructions on how to deploy this application.

Optional Configuration Tasks

This section describes the optional procedures that may be needed to complete the setup of Notification Manager.

How You Create Or Update a Notification Manager Database on a Non-CA Automation Point Server

A Notification Manager database can reside on a server where CA Automation Point is not installed. This lets you centralize your databases and minimizes the number of Microsoft SQL Server licenses your site requires. CA Automation Point can remotely connect to a Notification Manager database created on another system.

You must install a supported version of Microsoft SQL Server on any server machine where the Notification Manager database is to reside.

Note: For supported releases of Microsoft SQL Server, see the chapter "System Information" in the *CA Automation Point Release Notes*.

After you install (or upgrade) a CA Automation Point server on one machine, you need to install (or upgrade) a remote database.

To install a remote database

1. On the remote server, login as a user who has the SQL privilege to update SQL databases using Windows authentication. Make sure that the SQL command-line utility `osql.exe` is on that user's PATH.
2. Start the CA Automation Point installation program from the CA Automation Point DVD.
3. Select Setup Remote NM database.
4. Enter the appropriate values into the fields on the dialog.
5. Click Setup to create or update the NM database schema on the local system.

6. Return to your CA Automation Point server, and navigate to Configuration Manager, Expert Interface, Notification Services, Notification Manager, NM Setup.
7. Establish a connection from CA Automation Point to the remote NM database. This populates the database with initial values. If you already established access to a remote database in your previous CA Automation Point release, disable access, click OK, and then re-enable access. This confirms that CA Automation Point can establish a connection and properly recognize the recently updated remote database.

Configure Voice Notification and Call-in Features for Notification

To enable the voice notification and call-in features for Notification Manager, you create the Notification Manager voice groups NMCALL and NMANSWER.

Note: These features are available only to a notification server that has a voice card installed.

- **NMCALL Group**

When you define the NMCALL group, the voice notification feature is activated. The voice notification feature allows you to notify a contact through a phone call or a voice message, and optionally to prompt the contacted person to respond to a question using the phone's touch-tone pad.

Before configuring the NMCALL group, you must determine the number of channels you want to assign to it. You base this number on the amount of notification traffic that you expect Notification Manager to generate. You can adjust the number at a later time when the notification traffic is more clearly defined.

- **NMANSWER Group**

When you define the NMANSWER group, the call-in feature is activated. The call-in feature allows the contacted person to call Notification Manager to acknowledge receipt of the notification, and to address any reported problem or event through the use of the phone's touch-tone pad. This feature provides notification methods that are inherently "outbound only" with a way to acknowledge notifications or to respond to questions through the use of touch-tone pad entries. Examples of such methods are email and paging.

Before configuring the NMANSWER group, you must determine the number of channels you want to assign to it. As with the NMCALL group, the number of channels you assign to the NMANSWER group is based on the expected call load. You must also specify the phone numbers of the channels assigned to the NMANSWER group.

To configure the NMCALL and NMANSWER groups

1. From the Configuration Manager main window, choose Expert Interface, Notification Services, Notification Manager, NM Voice Groups.

The dialog NM Notification Server Voice Channel Groups displays.

2. Enter the name of a notification server that has the "Use Voice Card" option enabled.

The notification server can be either local or remote. If the notification server service is remote, the local notification server and the the remote notification server must be network-connected using PPQs. Both local and remote notification servers must also be configured to use the same NM database.

3. Choose Edit.

The dialog that displays has two tabs—NMANSWER and NMCALL—and displays a list of all the channels available to the voice card.

At this point, you can designate which channels are to be used only to call out (NMCALL) and which channels can be used only to call in (NMANSWER). For you to enable the call-in feature, at least one channel must be assigned to the NMANSWER group along with at least one phone number. If more than one channel is part of the NMANSWER group, CA recommends that you provide a phone number for each channel. One of the phone numbers entered here will be part of the message Notification Manager issues to a contact when the contact is required to call in.

4. Click OK to commit the changes to the database.

If the notification server service is local, the service is recycled.

If the notification server service is remote, a dialog box displays stating that the service must be recycled on the remote box before the changes cantake effect. Go to the remote workstation and manually recycle the notification server service to ensure the changes are activated.

When the service is recycled, Notification Manager starts one NMANSWER program for each channel in the NMANSWER group. The NMANSWER program secures an NMANSWER group channel and awaits a call-in from a contact.

Note: The individual notification methods (for example, Alphanumeric Pager 1 or Email 1) do not need any configuration changes (such as a change to the Usage setting) for you to use the call-in feature.

Notification Website Search Capabilities

We recommend that Notification Website users specify at least a one- or two-letter search pattern when conducting a search of contacts or logins. This is especially important on systems with large databases. Limiting the search in this way provides for fast data retrieval and produces fewer records to scroll through.

If users of the Notification Website do not follow this practice of limiting searches, the general searches for contacts or logins that are conducted may require a large number of records to be retrieved. The time it takes to retrieve these records may lead to delays due to network performance issues. To minimize the delays, the Notification Manager administrator can configure the Notification Manager Gateway Server to limit the number of records retrieved for each search. If the limit is reached, the user is advised that not all records were returned and that he or she can use a more specific search pattern to lower the number of records to be retrieved. The default value for this limit is 50. This value can be increased or decreased depending upon the throughput of the network.

Consider this example. A Notification Website user wants to modify the contact James Smith, and decides to search for this contact. The user enters the wildcard (*) in the search pattern. The database contains a thousand contacts. Because the wildcard matches all contacts, all 1000 records are returned. A long time goes by before all these records are returned to the user, and when they are, he or she must scroll through 1000 records to find James Smith. If the system is limited to returning only 50 records, 50 records are returned quickly and the user is advised to provide a more specific search pattern. James Smith might be included in the 50 records returned, but this is not guaranteed. If the user puts Ja* in the search pattern, it is likely that far fewer than 50 records would be returned, and the scrolling to find the contact would be much easier. For future searches, the Notification Website user would realize that the best way to quickly retrieve a record is to provide as much detail as possible in the search pattern.

To limit the number of records retrieved on searches

1. From Configuration Manager, choose Expert Interface, Notification Services, Notification Manager, NM Website.

The NM Website dialog displays.

2. Change the value in the Maximum Number of Search Records field.

The default value is 50.

The NM Gateway Server service does not need to be recycled after the value is changed. The new setting takes effect as soon as the NM Website dialog is closed.

Security Options for Notification Manager

To establish security for the Notification Website, choose Expert Interface, Notification Services, Notification Manager, NM Security from the Configuration Manager main window. Using this expert interface, you can configure the security options for the Notification Website as follows:

- Establish authentication by choosing a login authentication mode.
- Control authorization by enabling or disabling the advanced LDAP permissions.
- Change the passwords of the NM built-in login names.

Note: For a description of Notification Website security and more information about Lightweight Directory Access Protocol (LDAP), see [Secure the Notification Website](#) (see page 309).

Login Authentication

The Notification Manager website supports three modes of login authentication, which are represented as options on the NM Security Options dialog. The three modes are:

- No authentication
- Windows authentication
- LDAP authentication

The modes and the options you choose for them on the NM Security Options dialog are:

No Authentication

To forgo any authentication for website users, choose the None option on the NM Security Options dialog. When you choose None, passwords are not checked. This option is the default and is the same as what was done in previous releases. However, before using this option long term, you should thoroughly evaluate your security situation.

Windows Authentication

To specify that user names and passwords be checked against the Windows domain server, choose Windows Default Domain Server in the NM Security Options dialog. If you do not specify a name for the default domain server, the CA Automation Point server is used.

If you use Windows authentication, the CA Automation Point Server box must be running either in the domain where the authentication occurs, or in a domain that shares a trust relationship with the domain where the authentication occurs.

Note: The user can optionally specify an alternate Windows domain server when logging on to the website. This is done by using the backward slash character as a delimiter. For example, a user could specify the login user name as follows:

```
windomain_2\user_name
```


The login user name submitted for Windows authentication must match what is specified in the Notification Manager database. In the preceding example of specifying alternate domain, the login name specified in the Notification Manager database should also be `windomain_2\user_name`.

LDAP Authentication

For the user name and password to be checked against an LDAP-compliant directory server, choose LDAP Default Login Server and specify the following options on the NM Security Options dialog:

LDAP Default Login Server

The host name or host TCP/IP address of the default LDAP server (Required)

Port Number

The host TCP/IP port number of the default LDAP server (Required)

User DN Prefix

The user DN prefix used to make up the final user DN's to authenticate the login with the server

User DN Suffix

The user DN suffix used to make up the final user DN's to authenticate the login with the server

Note: A DN (or Distinguished Name) is a unique identifier used by the LDAP server to authenticate logins. The final user DN's are in the concatenated form of the user DN prefix, the login user name supplied by the user, and the user DN suffix.

The user can optionally supply an alternate LDAP server and port number when logging on to the website. This is done by using the backward slash character as a delimiter. For example, the user could specify the user name as follows:

```
ldapservers_2:port_2\user_name
```

The login user name submitted for LDAP authentication must match what is specified in the Notification Manager database. In the preceding example of an alternate LDAP server and port number, the login name specified in the Notification Manager database should also be `ldapservers_2:port_2\user_name`.

NM Built-in Login Names

Notification Manager provides two built-in login names that you can use initially to start up your site, as well as for subsequent transitions. These login names and their initial passwords are:

NmAdmin

Identifies the built-in administrative login name, with a password "nadmin"

NmGuest

Identifies the built-in guest login name, with a password "nmguest"

You can use these login names as a basis for authorization as well as authentication. For detailed information on these login names, see [Secure the Notification Website](#) (see page 309)

Important! We strongly recommended that you change these initial passwords before publishing the website.

Advanced LDAP Permissions

The Notification Website supports two modes of user authorization, Basic Privileges and Advanced LDAP Permissions. Basic Privileges authorization is always enabled. Optionally, you can enable the advanced mode of permissions with a third-party LDAP-compliant directory server. In this case, the LDAP server is queried for additional user permissions data. If you want to enable advanced LDAP Permissions, specify the following options on the NM Security Options dialog. All of these options are required.

Enable

Enables, or disables, the Advanced LDAP Permissions user authorization option. The default is disabled.

LDAP Permission Server Name

Configures the host name or host TCP/IP address of the LDAP server that has the advanced permissions data.

Port Number

Configures the host TCP/IP port number of the LDAP server.

AP Connection User DN

Configures the user DN that the Notification Website uses to authenticate with the LDAP server to query permissions data.

AP Connection Password

Configures the password that the Notification Website uses to authenticate with the LDAP server to query permissions data.

NM Base DN

Configures the base DN that identifies the root of the NM LDAP tree within the larger LDAP "forest." This root entry is the directory from which all NM searches begin. All NM related advanced user permissions data should reside under this base DN. For example, it may appear as follows:

```
ou=CAAP,o=your_company,c=your_country
```

Note: For more information about LDAP, see [Secure the Notification Website](#) (see page 309).

Note: The Notification Website does not use a directory's internal access controls for the implementation of the advanced LDAP permissions; instead, it follows the directory security policies set by the LDAP server, as all other users of LDAP do. The Notification Website application connects with the LDAP server using the AP Connection User DN and password that you specify, queries the LDAP server for the NM permissions data, and then grants or denies the user permissions that are requested.

Notification Manager Concepts

The core of Notification Manager is more of a control center and database handler than an actual method of contacting people. It allows you to create any number of methods to be used to contact people. CA provides you with a starter set consisting of voice paging, email, alphanumeric paging, and numeric paging. You can add any number of your own methods. Currently, the method must be a REXX script.

Notification Manager makes it possible to notify contacts through different methods. You almost always want to use a different method to contact people depending on where they typically are at a particular time of day. If they always wear a pager, you can always have them paged; however, there are other considerations.

The methods that Notification Manager uses to notify a contact fall into two categories:

One category notifies a contact by relaying a message and by soliciting a response to a question. The method accepts the response and passes the data along for site automation. For example, a contact may be notified by a voice phone call that will play prerecorded messages, record contact initiated messages for forwarding to all subsequent people handling the problem, ask a question, and wait for a response.

Methods that fall into the second category simply relay a message. Such methods when combined with the Notification Manager Call-in feature can be used to notify, confirm notification receipt, and solicit a response to a question. For example, whenever Notification Manager notifies a contact using a pager, Notification Manager displays a call-in phone number and item ID on the pager display. Upon receipt, the contact dials the displayed call-in number. After a voice call connection is established, Notification Manager prompts the contact to enter the item ID, relays a message, asks a question, awaits the response by the contact, and passes the response along for further automation.

Notification Manager supports a number of communication methods directly, including:

- TAP-compliant pager service
- Voice notification using the ADDRESS VOX environment
- Email
- Two-way paging notification using a paging service that supports either the SNPP protocol or the WCTP protocol (version 1.1)
- Text-to-speech network communications to a Windows desktop PC

Because Notification Manager is an open application, you can also define your own custom methods to use in particular circumstances. Examples of custom methods include generating an operator message, sending a FAX, opening a problem ticket, and putting a message on a CA Automation Point PPQ.

Backup Methods

Notification Manager makes it possible to notify contacts through several different methods, one after another. If the initial notification attempt fails, Notification Manager reverts to backup methods (which you have defined) to notify the contact.

Suppose that Joe is responsible for the very important payroll application. Notification Manager makes it easy to ensure that every possible means of contacting Joe is used.

Note: If the current time is within the time(s) covered by a schedule entry, the time is said to be *active*; otherwise, it is said to be *inactive*.

If you want Notification Manager to use more than one method when attempting to notify Joe, simply define multiple schedule entries covering the same period of time and assign a different method to each one. For a very important application, you can tell Notification Manager to try to contact Joe by phone whenever possible (for the reasons covered previously), but you would also want to tell it to use a pager if Joe cannot be notified by phone.

Another reason to use backup methods is when you cannot be certain where to contact a person at a particular time. For example, Joe leaves the office between 4:00 p.m. and 6:00 p.m. each day to make his one hour commute home. In that case, you would want Notification Manager to try to notify Joe:

- At work up to 6:00 p.m.
- In the car from 4:00 to 7:00 p.m.
- At home from 5:00 p.m. onward

Notification Manager will then try to notify him in all three places from 4:00 to 6:00. When you tell Notification Manager to use backup methods, it provides you with a means, called *priority*, of telling it which method to try first.

Overrides

Because a contact may have a standard weekly schedule, Notification Manager provides you with day of the week (DOW) scheduling using time blocks. This allows you to input time entries that do not vary from week to week. The time blocks may only have a single start and stop time but can cover any number of the days of the week.

For example, Tim takes his lunch break from 12:00 p.m. to 1:00 p.m., Monday through Friday. In this case, use a single time block to specify his lunch breaks. As another example, Veronica has her planning meetings every Tuesday and Thursday from 9:00 a.m. to 11:00 a.m. Again, specify her schedule with a single time block. On the other hand, if Veronica's planning meetings were from 9:00 a.m. to 11:00 a.m. on Tuesday, but from 10:00 a.m. to 12:00 p.m. on Thursday, you would have to use two different time blocks because the start and stop times are different. The completed DOW schedule consists of entries telling Notification Manager what to do on a certain day or days of the week.

On the other hand, if you have a special assignment that changes how you should be contacted on a particular Tuesday, you would not want to change the time entries for Tuesday to handle that special event. Notification Manager provides date scheduling to make it easy to override the Tuesday schedule with a schedule that is specific to the Tuesday that falls on a particular date. Note that the date schedule does not *replace* the day of the week schedule. In this case, Notification Manager uses the methods specified for the specified date, and then if those fail, the methods for Tuesday.

Order of Events When Using Backups and Overrides

A date-specific schedule overrides a day of the week schedule solely because Notification Manager has a built-in algorithm for determining the order to use when performing processing for a time that has multiple active time blocks.

That algorithm is as follows:

1. Time blocks that specify a date are always performed before time blocks that specify a day of the week. (A time block that is specific to Tuesday, July 3 is probably more accurate than one that covers every Tuesday of the year.)
2. The highest priority time block is selected. (The user knows best.)
3. If two time blocks that specify a date are active, the one whose beginning date is closest to the current date is performed first.
4. If two day of the week time blocks are active, the one whose first active day of the week is closest to the current day of the week is performed first. (A time block that covers July 15 through July 18 is probably more accurate on July 16 than one that covers July 1 through July 31. Similarly, a time block that covers Wednesday through Friday is probably more accurate on Thursday than a time block that covers Sunday through Saturday.)

5. If there is still a tie, the time block whose start time is closest to the current time is performed first. (A time block that starts at 9 a.m. is probably more accurate at 10 a.m. than one that starts at 1 a.m.)
6. If there is still a tie, the time block whose end time is closest to the current time is performed first. (A time block that covers from 9 a.m. to 11 a.m. is probably more accurate at 10 a.m. than one that covers from 9 a.m. to 9 p.m.)
7. If there is still a tie, a time block that forwards calls to another person is used after one that tries to reach the person Notification Manager is currently trying to reach.
8. If there is still a tie, the order is random/undefined.

Forwarding

Notification Manager allows you to forward notification to a predetermined contact in the event that the primary contact is unavailable. This feature is useful for situations that are planned to occur on a regular basis, or for last-minute schedule changes that require one contact to cover for another.

For example, Joe may be the person who is primarily responsible for the payroll application, but he wants to go on vacation. You have over 100 CA Automation Point rules that make calls to Notification Manager to contact Joe for problems with this application. Do you have to change them all? No. Joe can simply forward them to another contact. When Notification Manager is told to notify a contact, it looks at the method it is supposed to use. If that method is to forward to another contact, Notification Manager looks at the schedule of that other contact for the method to use.

Forwarding can also be used to handle lunch breaks, meetings, travel times between home and office, or anything that involves one person needing to hand over responsibility for something to another person for awhile. With forwarding, each person can maintain an individual, personal schedule of notification methods. In other words, when Joe forwards his calls to Mary from 12 p.m. to 1 p.m. for his lunch break, he does not need to know how to notify Mary during that time. He simply forwards his calls to her and Notification Manager looks up how to notify Mary, based on her schedule.

Create an Alias

Another powerful use of forwarding is to create pseudo-contacts, or aliases, that describe responsibilities. These aliases allow you to isolate jobs from the people who are currently filling them.

For example, as in the previous section, there might be over 100 rules for the payroll application that need to notify this application's administrator. If you set up those rules to notify Joe directly, when he moves into another position you must recode all the rules to notify someone else. A much more maintainable method would be to define a contact such as a "payroll administrator" to act as an alias. This alias will always be notified when there is a problem with the payroll application. Then, you simply create a single 24x7 time block for that contact that forwards notification to Joe. Joe can change his personal schedule any way he desires and Notification Manager can still find him when it needs to get in touch with the "payroll administrator." Also, when Joe moves on and Mary takes his place, all you need to do is change the payroll administrator contact to forward notification to Mary instead of Joe.

Forwarding can also be used to manage groups of people who take turns being responsible for something. For example, Joe actually shares the role of the payroll administrator with Susan and Tracy. You can define time blocks for the payroll administrator that divide the week or year up any way you want and make one of the three responsible for the payroll application at each particular time.

Priority Forwarding

Extending the concept of backup methods discussed previously, you can use priorities to make sure that Notification Manager tries multiple people anytime there is a problem, but tries them in a different order, depending on the time.

For example, you can define three schedule entries for the payroll administrator that cover 8:00 a.m. to 5:00 p.m., Monday through Friday, assigning them priorities of 3, 2, and 1. Then, you make the priority 3 time block forward to Joe, the priority 2 time block forward to Susan, and the priority 1 time block forward to Tracy. Next, you create three schedule entries that cover 5:00 p.m. to 8:00 a.m., Monday through Friday, and forward to Susan from the priority 3 time block, Tracy from the priority 2 time block, and Joe from the priority 1 time block.

When Notification Manager needs to notify the payroll administrator between 8:00 a.m. and 5:00 p.m. on Monday through Friday, it tries to notify Joe, then Susan, then Tracy. If it needs to notify the payroll administrator between 5:00 p.m. and 8:00 a.m. on Monday through Friday, it tries Susan, then Tracy, then Joe.

Nested Notification Forwarding

When one contact forwards notification to another contact, that contact can, in turn, forward to yet another, creating nested forwarding. This nesting can be to any depth.

Availability

Notification Manager allows you to change the availability status assigned to an individual contact. The availability status of an individual determines how Notification Manager processes notification requests for this individual. If the individual is marked as unavailable, Notification Manager does not directly attempt to notify that individual. However, because forwarding methods point to a contact other than the selected individual, Notification Manager does consider forwards during the notification request.

You can use the Modify Individual Contact page within the Notification Website to change the availability of an individual contact. In addition, you can use this web page to specify the reason that this individual is currently unavailable. This reason is displayed on the Send a Notification web page, after the unavailable contact has been selected for notification.

Although Notification Manager processes forwards currently assigned to an unavailable contact, the Send a Notification web page can only be used to send notification requests to those contacts that have at least one non-forward (direct) method currently active. For this reason, you cannot notify an unavailable contact using this Send a Notification web page.

Groups

Notification Manager enables you to create groups of people. The grouping can be based on any criteria that you choose. Membership in a group can be on a 24 hour a day, 7 day a week basis, or it can be based on a person's availability at various times of the day. By setting up time blocks according to the availability of each person in the group, you can create a membership directory that will contact the appropriate person at the appropriate time.

Note: Individual escalations and forward-tos do not occur if the contact they escalate or forward to is a member of the group. For example, Jack and Jill are part of a group, and Jack escalates to Jill. When the group is notified and the calltree constructed, Jack's escalation to Jill is suppressed because Jill is a member of the group.

Broadcasting

Notification Manager's default behavior is to stop processing as soon as it successfully notifies one person or one member of a group. However, it can also be used to broadcast information to an entire group or send information to an individual in more than one way.

Example:

This example uses broadcasting to call a department meeting.

1. Create a group called Dept7.
2. Have Dept7 refer to each person in Department 7 for 24 hours a day, 7 days a week.
3. Tell Notification Manager that it should use all active time blocks and notify all active members. (Recall that a time block is active if the current time is included within the time span covered by the time block.)

The Dept7 group can now be used to notify all the members of the group.

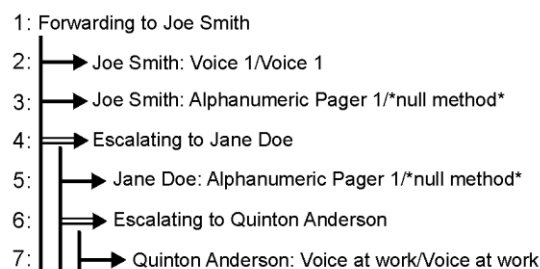
Escalation

Sooner or later, Notification Manager will be unable to notify the contact that you request through any of the specified methods. To help you handle this case, Notification Manager provides the concept of escalation. **Escalation** allows you to forward notification from one contact to another. Escalation is always active but is only used when all of the other methods that you specified have failed.

The following examples illustrate the escalation functionality.

Example 1:

Joe Smith is a contact who can be contacted by the Voice 1 and the Alphanumeric Pager 1 methods. Joe escalates to Jane Doe. Jane can be contacted by the Alphanumeric Pager 1 method. Jane escalates to Quinton Anderson. Quinton can be contacted by the Voice at Work method. Quinton does not escalate to anyone. The call tree for this example would look as follows:



A notification is issued for Joe Smith (for example, NMFIND NAME(Joe Smith) TELL('JES is down')). Notification Manager attempts to call Joe using Voice 1, but cannot reach him. It next attempts to use Alphanumeric Pager 1 to contact him, but this also fails. Left with no more methods to attempt for Joe, Notification Manager escalates to Joe's escalate person. In this case, the escalate person is Jane Doe. The attempt to contact Jane using Alphanumeric Pager 1 works. At this point the NMFIND is successful and exits without an attempt to contact Quinton Anderson.

Example 2:

Using the same scenario as before, assume Joe Smith cannot be contacted. He escalates to Jane Doe. Jane Doe cannot be contacted using the Alphanumeric Pager 1 method. Because no other methods are defined for Jane, Notification Manager escalates to Quinton Anderson. Notification Manager attempts to contact Quinton using the Voice at Work method. This method fails. Because Quinton has no more methods to attempt, Notification Manager attempts to escalate. However, Quinton did not define a contact to escalate. Escalation has reached the termination point and NMFIND.REX completes without any successful contacts.

Loops

Because forwarding can be nested, you can accidentally create a situation where forwards form a loop so that no one can be notified. Notification Manager treats this case as a failure of all regular methods and tries to notify the contact's escalation contact.

Duplicate Entities

Forwarding can also cause a situation where Notification Manager is supposed to use the same schedule entry (the same actual method and parameter set for a particular contact) more than once. With Notification Manager's current architecture, these duplicate notification attempts are suppressed (not performed).

Local Escalation

Notification Manager gives each contact a choice concerning escalation paths when the contact forwards notification. The contact must choose whether the escalation path of the contacts that it forwards to should be used. For example, Joe can specify a schedule entry that forwards notification to Mary. Notification Manager (at the appropriate times) uses this information to notify Mary in Joe's absence. However, Notification Manager may be unable to notify Mary and thus need to escalate. In this case, Joe must decide whether Notification Manager should use Mary's escalation path.

The decision is *not* whether to use Joe's escalation path. Joe's escalation path is always active when Joe is supposed to be notified. When Joe forwards notification to Mary, he must decide whether he wants to allow Mary's escalation path to be tried before his is tried. If Mary's escalation path is tried and actually succeeds in notifying someone, then Joe's forward to Mary is considered successful.

Local Forwarding

Another decision that an individual contact must make is whether to allow Notification Manager to use the forwards of the contacts that it forwards to.

To continue with the same example, a decision Joe must make when forwarding to Mary is whether to allow Notification Manager to use the forwards that Mary has specified for her personal schedule. If Joe does not allow Notification Manager to use Mary's forwards and it cannot find any other methods to notify Mary, the attempt to notify Mary is considered to have failed and Notification Manager uses Mary's escalation (provided that Joe allows it). If Joe does not allow Notification Manager to use Mary's escalation, Notification Manager considers the forward to Mary to have failed and proceeds as appropriate.

REXX Programs

This section covers the REXX programs that Notification Manager uses to implement different tasks involved in the maintenance and administration of Notification Manager. Among these, NMFIND is of special interest, as it is the REXX program that you will invoke from your automation when you want Notification Manager to notify someone for you. The programs follow:

- The NMFIND REXX program is used by anyone that wants to use Notification Manager to notify someone. Most likely, those people are the ones that write automation at your site. However, it could also be a manager who wants to locate an employee quickly.
- The NMANSWER REXX program provides Notification Manager's call-in feature. At installation time, this feature is installed on a single client machine of each notification server that has an NMANSWER group defined. It spawns itself on that computer until there is one NMANSWER running for each phone line in the NMANSWER notification server group.
- A Notification Manager method is a REXX program invoked by Notification Manager using parameters and settings defined in the Notification Manager database. The invocation program is any REXX program that follows Notification Manager's architectural parameter requirements. For your convenience, CA Automation Point provides invocation programs that are architecturally compatible with Notification Manager along with sample Notification Manager method definition data. The sample method data predefines parameters and their default settings. You customize these method parameter settings with site-specific values. These REXX programs issue page, email, and voice notifications through the use of the ADDRESS VOX command environment whenever the notification server is enabled. You can extend or modify these REXX programs to create a Notification Manager method to meet the requirements of your site.

Modifying or Extending Notification Manager

Due to the open architecture of Notification Manager, you can modify or extend Notification Manager in these ways:

- By giving an existing REXX program a new personality (using parameters)
- By creating your own REXX programs that extend the ways in which Notification Manager can communicate with the outside world beyond the voice, email, DTMF, and TAP paging services that CA provides. These REXX programs can open problem tickets, send faxes, and non-TAP pages, or perform any other type of communication you need.

This section covers the conceptual basics of these processes. See the *Command and Keyword Reference Guide* for details of the Notification Manager API.

Defining Methods to Notification Manager

Methods are aliases for a REXX program and a set of parameters. For Notification Manager to notify someone, it must use REXX programs that actually know how to communicate with the outside world. For example, the NMVOICE program knows how to send speech over a phone line and accept DTMF input, and the NMPAGE program knows how to send a page to a TAP-compliant paging service.

CA provides you with a starter set of the following REXX programs to use when creating your methods:

- NMPAGE, which allows you to issue TAP-compliant alphanumeric and numeric pages through CA Automation Point's notification services
- NMTAP, which allows you to issue TAP-compliant alphanumeric and numeric pages without the use of CA Automation Point's notification services. The NMTAP REXX program can therefore be user customized to interface with paging services that are not fully TAP-compliant.
- NMPAGE2WAY, which allows you to issue a page request and receive a response to this page request using a paging service that supports either the SNPP protocol or the WCTP protocol
- NMMAILPG, which allows you to issue a page using the local email system
- NMMAIL, which allows you to notify a person or group using the local email system
- NMSPEAK, which allows you to perform speech notification using the TCP/IP network
- NMVOICE, which allows you to perform voice notification
- NMNETSND, which allows you to perform NET SEND notifications from one workstation to another using the TCP/IP network

Using these programs, CA creates the following methods in the sample Notification Manager database that is shipped with the Notification Manager component:

- Voice at work
- Voice in the car
- Voice at home
- Voice 1
- Voice 2
- Alphanumeric Pager 1
- Alphanumeric Pager 2
- Numeric Pager 1
- Numeric Pager 2
- 2-Way Pager 1
- 2-Way Pager 2
- TAP 1
- TAP 2
- EMail 1
- EMail 2
- Email pager via Arch
- Email pager via SkyTel
- Speak 1
- Speak 2
- Net Send 1
- Net Send 2

The open architecture of Notification Manager allows you to create your own REXX programs and use them to define additional methods that Notification Manager can use when attempting to notify people.

Consider the following points as you decide on the set of methods your Notification Manager installation will have:

- Method names are independent of the actual invocation string. Thus, you can choose any name for the methods you create (for example, you do not need to name a method that invokes the NMVOICE REXX program NMVOICE).
- A particular REXX program can be used to create any number of methods (usually with a different set of parameters or parameter values for each method).

While you can simply define a single method that is always used to invoke NMVOICE and another that is always used to invoke NMPAGE, it is probably not in your best interest to do so. Rather, you should use the capabilities of Notification Manager to create methods that have "personality." For example, the NMVOICE program has several parameters that specify the number of times to call a person, the number of times to let the phone ring on each call attempt, and the amount of time to wait between attempts.

You may want Notification Manager to persistently try to notify a particular person, or to run quickly through a whole list of people because you need an answer quickly. You can accomplish this by creating the following methods:

- LongVoice, which invokes NMVOICE and tells it to try to call the person 20 times at intervals of 5 minutes and let the phone ring 10 times on each try
- QuickVoice, which invokes NMVOICE and tells it to try to call the person only 1 time and to let the phone ring only 3 times

It is much easier to specify the LongVoice method during a particular time block than it is to specify the NMVOICE method and the parameters and values that make it behave like the LongVoice method.

Notification Manager Parameter Levels

Notification Manager must have a very general view of parameters. To Notification Manager, parameters are simply strings that are passed to the appropriate method.

The most important consideration with parameters is the assigning of default values. Default values save typing, reduce data entry errors, and make implementing changes easier. On the other hand, some parameters cannot have a default (for example, the phone number for a particular contact). Notification Manager handles this diversity by providing the following three-level scheme in which to store parameter values:

- Method level
This is the highest level. It stores parameters that are associated only with a particular method.
- Personal level
This is the next level down. It stores parameters that are associated with a particular method and contact.
- Time block level
This is the lowest level. It stores parameters that are associated with a particular method, contact and time block.

Specifying Parameters

When specifying parameters to Notification Manager, place them at the highest possible level. For example, if everyone at your site uses the same pager service phone number, specify that number at the method level. Similarly, specify a contact's home phone number at the personal level rather than at the time block level.

In general, a value specified at a more specific level overrides a value specified at a less specific level.

Uses of the Three-Level Structure

The three-level structure for parameters is useful for reducing the number of times you have to specify a parameter's value. For example, consider the NMPAGE program. You can define a method for each different pager service you use. Therefore, for each pager service, you only need to define parameters like the pager service phone number, baud rate, and modem initialization string once. Also, you only need to define the pager ID for each person once at the personal level, rather than in every time block that needs to page the person. On the other hand, if the person needs to be notified using a different paging service or pager ID during a particular time block, you can specify those parameters at the time block level and they will override the method and personal level values.

Combined with the method concept, parameters also allow you to minimize data entry by giving parameters a more specific meaning. For example, recall that the voice at work and voice at home methods both use the NMVOICE program. The NMVOICE program does not have a HomePhoneNumber and a WorkPhoneNumber parameter; it has a single Phone() parameter, which is the phone number to call. However, because you have defined two different methods that point to the NMVOICE program, you only need to specify your home and work phone numbers once at the personal level for the two methods.

As you can see, defining a single method to be used when invoking a particular program usually means that you have to specify many parameters at the personal or time block level. Giving a program multiple "personalities" by creating more than one method to be used when invoking it allows you to fix parameters at the method or personal level, thereby letting you specify far fewer parameters at the more specific levels. Consequently, your maintenance overhead is greatly reduced.

On a separate note, you should be aware that when you are creating a method to specify a parameter at the time block or personal level, you must specify it at the method level (even if you specify a null or dummy value at that level). An example of this rule is the PHONE parameter of the NMVOICE program. It is highly unlikely that you will create a unique method for each phone number you want to call. Nevertheless, you must specify the PHONE parameter whenever you create a method that uses the NMVOICE program; otherwise, you cannot specify the phone number to use for a particular time block or contact.

Method Types

You can use method types in conjunction with methods to make notification requests more specific. Each Notification Manager method is associated with a method type. Method types allow you to group your notification methods into distinct categories, which you can use to limit the number of methods Notification Manager will consider when initiating a notification request.

Each method type (or category) is designated by a single letter from B through W. Notification Manager reserves the letter "A" to represent the built-in category containing all defined methods. Some method types are pre-assigned to CA-supplied methods. These are shown in the following table:

Website Method Type Description	Method Type Code Assignment	Method Name
Pager	P	TAP 1 TAP 2 Alphanumeric Pager 1 Alphanumeric Pager 2 Numeric Pager 1 Numeric Pager 2 Email Page via Skytel Email Page via Arch
Two-Way Pager	T	2-Way Pager 1 2-Way Pager 2
Email	E	EMail 1 EMail 2
Network Message	N	Net Send 1 Net Send 2 Speak 1 Speak 2
Voice Telephone Call	V	Voice 1 Voice 2 Voice at home Voice at work Voice in the car

The remaining method type designators (B, C, D, F, G, H, I, J, K, L, M, O, Q, R, S, U, and W) are available for your use to create new method types either to categorize site-specific methods or to change the default grouping of existing methods.

You can set the method type code for a specified method in two ways:

- Specify a custom method type code using the Create Method or Modify Method web page on the Notification Website
- Specify a TYP value when issuing either of the ADDRESS VOX commands CREATMETHOD or ALTERMETHOD to represent a type of method.

Using Method Types

You can use method types in your notification strategy to help Notification Manager determine how to notify a contact. For example, suppose that John Smith has two active schedules. One schedule is assigned to use the Email 1 method and the other to use the Alphanumeric Pager 1 method. If you have an urgent problem that requires John Smith's immediate attention, you can use the Pager method type (P) on the notification request to force Notification Manager to use only those methods assigned to this method type. This would ensure that John Smith would be notified using the Alphanumeric Pager 1 method instead of the Email 1 method.

Specifying Method Types Using the Notification Website

If you are using the Notification Website to send notification requests, you can use the Acceptable Types of Notification area of the Send a Notification page to select which types of notification methods should be considered for use on a particular notification request.

Using the MTUP Parameter on the NMFIND Command

If you are using the NMFIND program to directly issue a notification request, you can use the MTUP (Methods to Use Profile) parameter to specify the types of methods to use for the request. You do this by specifying the method type codes assigned to the methods to be considered for a particular notification request.

To view all currently defined method type codes assigned for all methods in the Notification Manager database, run the Notification Manager utility program, listmeth.rex. You can find this utility in the subdirectory SAMPLE\NM in your CA Automation Point installation directory. The method codes are listed as the TYP settings. For example, the following is an entry in this file.

```
=====  
Method name      :      Email 1  
Key              :      200001010  
Tmflags         :      00000000  
Use              :      TELL  
Typ setting     :      E  
Invoke          :      NMAIL  
=====
```

The code assigned to the Email 1 method is E. You could use this value on the MTUP parameter to specify that the NMFIND attempt only active schedules whose methods are defined as method type E.

Alternatively, you can view currently defined method type codes using the Notification Website. Choose to create or modify a method and select a method name to view the method type setting. If a custom method type code was defined, that method code will be displayed.

Note: For more information on the MTUP parameter, see the *Command and Keyword Reference Guide*.

Telling and Asking

Notification Manager supports the following basic types of communication with personnel using the NMFIND program:

- Telling them something (specify using the TELL parameter of the NMFIND program)
- Asking them a question and then giving them a set of options from which to choose (specify using the ASK parameter of the NMFIND program)

Note: Notifications issued from the Notification Website are TELL only. For more information, see [How You Use the Notification Website](#) (see page 304).

When you define a method, you tell Notification Manager which of these capabilities the underlying REXX program has. For example, a method that sends a FAX can tell, but it cannot ask because there is no way to get a response back from the person by a FAX machine.

For each method you define, you tell Notification Manager that you want to allow the method to be used to tell or both tell and ask. Notification Manager then enforces this selection when the method is used by a time block. In other words, if the method is defined as only supporting the TELL parameter of NMFIND, Notification Manager does not allow the method to be specified as the ask method for a time block.

If the method is specified as supporting both the TELL and ASK parameters of NMFIND, you are allowed to specify it as just the TELL method for a time block, or as both the TELL and ASK methods.

NMANSWER and the Call-in Feature

Many methods (for example, pager and email) can only be used to tell a contact about an event. In many situations it is desirable for the contact to acknowledge that a notification was received or to take an action to address a problem. Notification Manager supports a call-in feature that allows every method to support the ASK function. See the section [Configure Voice Notification and Call-in Features for Notification](#) (see page 269) earlier in this chapter for details on how to activate this feature.

Using the Call-in Feature

The following scenario describes how a contact could use the call-in feature.

A contact receives a notification and wants to use the call-in feature to either acknowledge that the notification has been received or to reply to the question raised by the ASK function. To do this, the contact needs four pieces of data. The first two are his or her contact ID and contact password. This information, needed for security purposes, can be obtained from the Notification Manager administrator or from the Notification Manager Website. The third piece of information is the item number of the notification to which the contact is responding. This four-digit number is part of the notification sent to the contact. The fourth piece of information is a valid NMANSWER phone number. The notification sent to the contact provides one of the valid NMANSWER phone numbers.

Note: The user is not limited to calling in on just the NMANSWER phone number provided in the notification. Any of the phone numbers that the NMANSWER group represents can be used.

When the contact calls in, an NMANSWER channel picks up, and the NMANSWER program processes the call. After an introduction message, the user is prompted to type his or her contact ID and contact password. Then the system prompts for the item number the contact is responding to. After confirming the contact is permitted to handle the supplied item number, the message is played with a list of actions the user can take. If someone else has already called in and taken action on this item number, the system communicates that the item has already been handled and tells the caller what action was taken.

Notification Manager Commands

The following list describes the Notification Manager commands. See the *Command and Keyword Reference Guide* for command syntax.

NMANSWER

Invokes the external interface for the call-in feature.

NMFIND

Invokes the external interface used for invoking Notification Manager to call personnel.

Notification Manager Method Programs

The following list describes Notification Manager method programs.

NMMAIL

Provides a means for NMFIND to notify a person or group using the local email system.

NMSPEAK

Provides a means for NMFIND to notify a person or group through speech using the TCP/IP network.

NMMAILPG

Provides a front-end to the NMMAIL program. NMFIND uses this command to notify a person or group by initiating an alphanumeric page using the email system.

NMNETSND

Provides a means for NMFIND to notify a person or group via the NET SEND command using the TCP/IP network.

NMPAGE

Allows NMFIND to notify someone using a TAP alphanumeric or numeric pager.

NMPAGE2WAY

Allows NMFIND to notify someone using a 2-way messaging device. If a question was specified, the user may also respond to the question by replying to the page request.

NMTAP

Allows NMFIND to notify someone through a TAP alphanumeric or numeric pager. It differs from NMPAGE in that the REXX code talks directly to the modem instead of using the ADDRESS VOX PAGE command to perform the page. NMTAP supports the batching of pages (that is, sending multiple pages to a paging service with a single phone call). NMTAP is designed so that you can easily modify it to support a different protocol than TAP. For more information, see the NMTAP.txt file in the Distrib directory.

NMVOICE

Allows NMFIND to notify someone using a voice card and, optionally, ask them to make a selection from a set of options.

Note: For more information, see the *Command and Keyword Reference Guide*.

Basic Rules for Creating a Personal Schedule

The Notification Manager architecture allows you to create a robust notification strategy using a few basic rules:

- Any number of time blocks (notification schedules) can be specified for an individual or group.
- Two or more time blocks can span:
 - The exact same period of time. For example, you can use multiple active time blocks to cause Notification Manager to attempt to contact you by phone, and if that fails, to:
 - Attempt to contact you by phone at a different number
 - Attempt to contact you by pager
 - Forward to someone else who is acting as a backup for you during a particular period of time
 - Overlapping periods of time. You can use this to tell Notification Manager that your schedule is somewhat flexible. For example, suppose you leave work anywhere from 4:00 to 6:00 in the evening. You would tell Notification Manager to try you at work from 8:00 a.m. to 6:00 p.m., in the car from 4:00 p.m. to 7:00 p.m., and at home starting at 5:00 p.m.
 - Completely separate periods of time.
- Time blocks can be specified either to cover a date range or to span a subset of the days of the week. If two time blocks cover the same period of time and have the same priority value, Notification Manager prefers those time blocks that specify a date range to those that specify a weekly schedule. Time blocks that are based on a date range are useful when you need to define a temporary change to your weekly notification schedule for events such as vacations.

- Time blocks can cross midnight. This is determined by the values specified for BeginTime and EndTime. When the EndTime is earlier in the day than the BeginTime, the time block is interpreted as crossing midnight into the next day. Consider the following time block:

BeginTime: 11:00 PM

EndTime: 7:00 AM

This time block starts at 11:00 at night, crosses midnight, and ends at 7:00 the next morning.

Similarly, when the EndTime is the same as the BeginTime, the time block is interpreted as crossing midnight into the next day. Consider the following time block:

BeginTime: 9:00 AM

EndTime: 9:00 AM

This time block starts at 9:00 in the morning, crosses midnight, and ends at 9:00 the next morning.

When scheduling by date, you cannot specify the same BeginDate and EndDate when your times specify that your timeblock will cross midnight. For example, you will receive an error if you specify 7/10 for both BeginDate and EndDate in either of the two previous examples. Both of those examples begin on one day, cross midnight, and end the next day. Therefore, the appropriate EndDate would be 7/11.

In general, you can interpret that notifications will stop at the EndTime on the EndDate. The one exception is midnight. Notification Manager interprets an EndTime of 12:00 AM as the end of the current day, not the beginning of the next day. Consider the following time block:

BeginTime:12:00 AM

EndTime: 12:00 AM

BeginDate: 7/10

EndDate: 7/10

This time block is interpreted as starting as the first thing in the morning on July 10, and ending as the last thing at night on July 10. Because a time block which ends at 12:00 AM, never actually crosses midnight, the specified EndDate should not be the date of the next day.

- A specified BeginTime/EndTime constraint is enforced separately on each individual day of a multi-day timeblock. Consider the following time block:

BeginTime: 5:00 PM

EndTime: 12:00 AM

BeginDate: 7/1

EndDate: 7/5

With this time block specification, the person to be notified would be on-call from 5:00 in the afternoon until midnight on each of the five days of the time block. On each of those 5 days, the person will not be notified in the morning or afternoon, until 5:00 PM.

- One way to place someone continuously on-call from the StartTime on the StartDate to the EndTime on the EndDate is to use a 24 hour time block. This is accomplished by specifying the same time for the BeginTime and EndTime. Consider the following time block:

BeginTime: 10:00 AM

EndTime: 10:00 AM

BeginDate: 7/1

EndDate: 7/5

With this time block specification, the person to be notified would be on-call continuously from 10:00 on the morning of July 1st until 10:00 on the morning of July 5th. If your BeginTime and EndTime are not the same time, you have to use multiple time blocks to achieve continuous notifications.

- Time blocks can cross into the next year. Because a year is not specified as part of the value for either the BeginDate or EndDate, if the EndDate is earlier than the BeginDate, the time block will cross into the next year. Consider the following time block:

BeginDate: 10/1

EndDate: 2/15

This time block will start in October, and continue until February of the next year.

- Time blocks react to leap years. If you specify a BeginDate or EndDate of 2/29, that exact date will only occur every four years. On the years which do not have 2/29, Notification Manager interprets the other dates within the time block as if a 2/29 did exist. The difference is most noticeable when such a time block also crosses midnight. Consider the following time block:

BeginTime: 8:00 PM

EndTime: 3:00 AM

BeginDate: 2/15

EndDate: 2/29

Every year, the person will always be notified on 2/28 from 12:00 in the morning until 3:00 AM, and again from 8:00 PM until midnight that same night. During leap years, the person will also be notified on 2/29 from 12:00 in the morning until 3:00 AM.

This behavior is exactly what was specified by the time block. However, you must recognize that the person will be notified from 8:00 PM until midnight on the last day of February on non-leap years, but not on leap years.

Similarly, consider the following time block:

BeginTime: 8:00 PM

EndTime: 3:00 AM

BeginDate: 2/29

EndDate: 3/5

Every year, the person will always be notified on 3/1 from 12:00 in the morning until 3:00 AM, and again from 8:00 PM until midnight that same night. During leap years, the person will also be notified on 2/29 from 8:00 PM until midnight.

Again, the behavior is exactly what was specified by the time block. However, you must recognize that the person's notifications will start at 8:00 PM on the *last day of February* during leap years, but will not start until 12:00 AM on the first day of March during non-leap years.

Note that you should have fewer special considerations if your times do not cross midnight. There are also no special considerations if your date range crosses 2/29, as long as neither your BeginDate nor EndDate is specifically 2/29.

- If it is important to your notification policy to specify the *last day of February*, and the behavior described previously is not acceptable, you can use a second time block for just the date of 2/29 to override the default behavior.

Creating Your Own Invocation Programs

This section explores some specific functions of Notification Manager that are helpful when using REXX programs.

The open architecture of Notification Manager allows you to create your own REXX programs, and then specify them as the invocation strings for methods in the Notification Manager database so that they can be used when NMFIND needs to notify someone. When you create your own program, its parameters fall into the following classes:

- Architected parameters, which are the interface between NMFIND and any program it calls
- Method-specific parameters, which are unique to your program
- Runtime parameters, which need to be passed between successive calls to your code (to notify different people) by a particular invocation of NMFIND

Sample Code

Several sample programs are installed with CA Automation Point and are located in the SAMPLE\NM\ directory. Here is a list of the files:

Program	Description
listanswers.rex	Lists supplied answers for the specified notification request item
listcallers.rex	Lists contacts notified for the specified notification request item.
listent.rex	Lists contacts in the database
listfind.rex	Displays a call tree for a contact
listitems.rex	Lists status information for issued notification request items
listlogin.rex	Lists logins in the database
listmeth.rex	Lists methods in the database
listMTUP.rex	Displays a call tree using the MTUP operand
listparm.rex	Lists parameters in the database
listpergrps.rex	Lists either the members of a particular group or the groups that are associated with a particular individual.
listtime.rex	Lists time blocks in the database
listforto.rex	Lists forwards to for a contact
nmdupmet.rex	Duplicates a method

Program	Description
nmdupper.rex	Duplicates a person
nmnewper.rex	Creates a person
nmshell.rex	Template for creating in-house methods
nmshell.txt	Explains how to use nmshell.rex to create in-house methods.
build_db.rex	Builds a database from a text file

Note: Certain .rex files have a corresponding .txt file that contains more information.

Architected Parameters

Notification Manager always passes the following parameters to your program. Do *not* place these parameters in the Notification Manager database if possible.

Ask

Specifies question and answer arguments. For a detailed description of this parameter, see the *Command and Keyword Reference Guide*.

Debug

Determines whether your program produces debugging messages in the ASOTRACE log. Values are:

YES—Your program should produce debugging messages in the ASOTRACE log. For an example of how to do this from a REXX program, see the DEBUGMSG subroutine in either the NMPAGE or the NMVOICE program. You can also use the ERRORMSG subroutine found in either of these programs.

NO—Your program should not produce debugging messages in the ASOTRACE log.

Item

Each time NMFIND is invoked, it obtains a unique four-digit number (called its *item* number) that distinguishes it from all other active NMFIND requests. This is that number. Only specify this parameter if you intend to allow your target person to use the call-in feature.

ItemDepth

When NMFIND is invoked, it builds a tree that contains all the time blocks that are active for the contact and time specified. This number specifies the depth of the tree node containing the time block that is causing your code to be invoked. Only use this parameter if you intend to pass parameters to other invocations of your code.

ItemIX

When NMFIND is invoked, it builds a tree that contains all the time blocks that are active for the contact and time specified. This number specifies the index of the tree node containing the time block that is causing your code to be invoked. Only use this parameter if you intend to pass parameters to other invocations of your code.

Tell

The string or voice file that you want to tell the notified personnel. The maximum length of the TELL string is 250 characters.

Use

Determines whether your program should handle the ASK parameter, the TELL parameter, or both parameters on this particular invocation. Values are:

ASK—Your program should only handle the ASK parameter.

TELL—Your program should only handle the TELL parameter.

BOTH—Your program should handle both the ASK and the TELL parameters.

Method-specific Parameters

Method-specific parameters are parameters unique to your code. These parameters must be stored in the Notification Manager database because this is the only way that Notification Manager can pass them to your code when NMFIND invokes it. The following rules apply to method-specific parameters:

- Parameter names must be 32 characters or less and consist of A-Z, 0-9, and underscore.
- Parameter values must be 250 characters or less.
- If a parameter value contains left or right parentheses, the value must be enclosed in single or double quotes when it is stored in the database.

Runtime Parameters

NMFIND allows the programs it calls to hand it parameter strings to be stored and then passed as part of the parameter list to every subsequent program it calls. These parameters are called *runtime parameters*. The strings passed to NMFIND must adhere to the standard REXX program parameter format, as follows:

- They must be in the form *keyword* or *keyword(value)*
- If the value specified contains parentheses, the entire value string must be enclosed in single or double quotes

There are no further restrictions on the values passed to NMFIND. NMFIND does not do anything with the values. It simply stores and forwards them.

This feature is used, for example, by the NMVOICE program to pass the name of the file that contains the record and forward .vox file. Recall that NMVOICE provides the ability for a person called by Notification Manager to record a message to be heard by all subsequent people that Notification Manager calls during a particular invocation of NMFIND. When a person asks to record a message, NMVOICE generates a unique file name and then stores the message in it. To have subsequent calls to NMVOICE (by NMFIND) find this file, NMFIND provides NMVOICE with a facility that enables it to pass the file name to every subsequent call to NMVOICE. NMFIND does not even know that it is storing the name of a voice file. It simply stores a parameter string that it knows should be passed on to all subsequent calls it makes.

If you plan to use this feature in your own programs, be aware of the following:

- NMFIND passes these values to all subsequent programs it calls. Thus, you should expect to see parameters in your program that you did not explicitly define to Notification Manager.
- The parameter strings are cumulative. For example, if program A passes back APARM(A) and program B passes back BPARAM1(B1) and BPARAM2, all subsequent programs called by NMFIND see the string APARM(A) BPARAM1(B1) BPARAM2.
- NMFIND does not allow you to modify strings after you have passed them back. Thus, you cannot change a parameter value once it is passed to NMFIND—you can only override or ignore it. For example, if NMFIND calls program A and it passes back APARM(A1), and then it calls A again and A decides that the value of APARM should be A2, it cannot tell NMFIND to dispose of the APARM(A1) value. It can only tell NMFIND to add another string, APARM(A2), and then code its handling for APARM to only use the last value found.

Your code must use the REXX external data queue (REXX XDQ) facility to pass parameter strings back to NMFIND. The format of the command is:

```
PUSH "NEWPARMS: string"
```

where *string* contains *keyword(value)* pairs or one or more *keyword* names. There must be a blank after NEWPARMS:. In the preceding example, program A's code would look something like this:

```
PUSH "NEWPARMS: APARAM(A)"
```

Program B's code would look something like this:

```
tval = "B1"  
PUSH "NEWPARMS: BPARAM1("||tval||") BPARAM2"
```

Again, keep in mind that this facility is used by all methods that NMFIND invokes (even NMVOICE and NMPAGE) and that your code may not be the only method that is invoked by NMFIND. Thus, your code must be able to discard any parameters it finds in the parameter string that it does not understand. Also, you must be careful to choose names that will not conflict with the names chosen by the programmers of other methods. All methods that CA supplies use parameters whose names start with "_NM". Do *not* use this prefix on any runtime parameters that you pass back to NMFIND. For examples of how to handle this type of parameter, see the NMVOICE program.

Emergency Mode Processing

NMFIND provides you, the user of the NMFIND script, with emergency mode processing for a simple reason. As the coder of NMFIND in your automation, you have no control over the methods that are used to notify the person you need to reach, and many methods can take a long time to complete.

For example, the LongVoice method, which invokes NMVOICE and tells it to try to call the person 20 times at intervals of 5 minutes and let the phone ring 10 times on each try, can take over 100 minutes to complete before returning control to NMFIND. Thus, even if there were other methods to be tried after the LongVoice method, they are not attempted for over 100 minutes. This is unacceptable when you need a response in a short amount of time.

Emergency mode processing circumvents this problem by providing you with a way to specify the amount of time each method can hold up the process of getting notification to someone who can handle the problem for which you invoked NMFIND. Emergency mode processing (using the EMERGENCYWAIT and FAILUREREXX operands) also gives you a way to specify the maximum amount of time you want to wait before attempting some other means (besides NMFIND) of obtaining a solution to a problem.

When you specify `EMERGENCYINTVL(n)`, NMFIND always starts new actions approximately *n* seconds apart. Even if all prior actions have completed so that none are running, NMFIND waits for the time interval to expire before starting a new action. NMFIND starts actions asynchronously on Windows by issuing the `START` command. Thus, each action runs as a separate process, using far more system resources than running all the actions synchronously inside a single process. Therefore, use this feature sparingly.

Before starting a new action, NMFIND checks all prior actions to see whether the request has been satisfied. If the contact had the "perform all active methods" flag set to `OFF`, the success of a single action satisfies the request. If the contact had the "perform all active methods" flag set to `ON`, all the actions that belong to the contact have to succeed before the request is satisfied. After the request is satisfied, NMFIND does not submit any new actions.

Because NMFIND submits new actions without regard to the completion of prior actions, NMFIND can end up with several actions running simultaneously. Therefore, NMFIND's normal behavior of quitting after the first successful action completes is somewhat modified. After an action completes successfully, new actions are not submitted, but actions that are already running are allowed to complete normally.

If NMFIND is only telling asynchronously (no `ASK` operand was specified), the first time NMFIND succeeds in telling the message, it considers the NMFIND request to be complete. When NMFIND considers the request complete, it stops starting new actions, but actions that it has already started continue to run and may eventually succeed in telling the message to the person who was supposed to be notified. Thus, more than one person can receive the message. In the synchronous case, only one person receives the message.

If a question is being asked asynchronously, the first answer received from someone is considered to be the answer to the question (even if that person was not the first person that NMFIND attempted to notify). After the question is answered, no new actions are started, but actions that were previously started continue to run. If another action subsequently succeeds in reaching someone, then that person hears the `TELL` message, the `ASK` question and answers, and the answer that was chosen, but NMFIND does not allow the person to answer the question.

Return Codes from Your Code

CA has reserved return codes in the range from 6200 to 6300 for your use. Your code should not return any other return codes.

To indicate failure conditions in your code to Notification Manager, you should exit with a return code and PUSH an entry onto the REXX XDQ. The format of the command is:

```
PUSH "ERROR: string"
```

string

Contains a return code that is optionally followed by explanatory text.

There must be a blank after the "ERROR:".

For example, if your code determines that it must terminate abnormally because of a shortage in virtual memory and you have defined return code 6208 to mean "insufficient virtual memory," your code could issue either of the following code sequences to exit properly according to the Notification Manager API:

Sequence 1:

```
rc = 6208  
PUSH "ERROR:" rc  
EXIT rc
```

Sequence 2:

```
rc = 6208  
PUSH "ERROR:" rc "Insufficient Virtual Memory"  
EXIT rc
```

Order of Parameters Passed to Your Code

When Notification Manager invokes a method, it passes all parameter types (architected, method-specific, and runtime) and levels (method, personal, and time block) of method-specific parameters to the method. The Notification Manager API defines the order in which your program receives the parameters:

- Architected parameters are passed first
- Method-specific parameters are passed next
- Runtime parameters are passed last

The order within the method-specific parameters is as follows:

- The parameters from the time block level are passed first
- The parameters from the personal level are passed next
- The parameters from the method level are passed last

The order in which parameters are passed is of particular interest in the case of method-specific parameters because the same parameter may have a value in the Notification Manager database at all three levels and, if that is the case, your method receives three different instances of that parameter. It is up to the method to decide how to handle this situation. The NMVOICE and NMPAGE methods both handle this by using the value found in the first instance of each parameter. (This means that they use the most specific value stored in the Notification Manager database.) Methods that you write can handle individual parameters differently.

For example:

- The method can take the most specific value for a particular parameter by using the first value it finds.
- The method can take the most general value for a particular parameter by using the last value it finds.
- The method can concatenate all values it finds.

How You Populate the Notification Manager Database

To establish your site's notification policies, you must perform the following administrative tasks in the order shown.

1. Set up methods.

In this step, you either modify supplied methods to better meet your site's needs or create new methods that refer to your own user-written REXX.

2. Set up people.

In this step, you define each person in the Notification Manager database, identify the methods by which a person can be notified, and set up the schedule for each person.

3. Set up groups.

In this step, you create a group name, create time blocks for the group, and then associate members of the group with each time block.

How You Use the Notification Website

The Notification Website is used to manage (create, edit, view, and delete) the methods, contacts, and login names used with Notification Manager. You must define a CA Automation Point DBMS before you can use the Notification Website.

Connecting to the Notification Website

Use your web browser to connect to the Notification Website. The format of the URL is:

```
http://webservicehost:port/context/default.jsp
```

webservicehost

Specifies the host name of the machine running the Java Servlet environment used to host the Notification Website. This host name must be the fully qualified host name as returned by the DNS server at your site.

port

Specifies the TCP/IP port number used by the Java Servlet environment to accept page requests. If you installed and configured the Java Servlet environment shipped with CA Automation Point, 8080 is the default port number used for these requests.

context

Specifies the context name (or servlet context) where the Notification Website was installed in the Java Servlet environment. By default, the context name used by the Notification Website is caapnfy.

Example:

You plan to install the Java Servlet environment shipped with CA Automation Point on the same machine as Notification Manager; the fully qualified host name of this machine is appgh1.ca.com; the context name is not changed from the default value; and the Java Servlet environment is not modified to change the default TCP/IP port number.

Under these conditions, the following URL would be used to launch the Notification Website home page:

```
http://appgh1.ca.com:8080/caapnfy/default.jsp
```

Overview of the Notification Website

The Notification Website allows authorized users to both update notification policy and issue notification requests to individuals and groups stored in the Notification Manager database.

Website Areas

The Notification Website is divided into three major areas, each of which is represented by a button on the Notification Website home page. These buttons provide access to pages described in the following table:

Button	Initial Page	Purpose
Send a Notification	Send a Notification	Initiate a notification request to any individual or group in the Notification Manager database.
Adjust My Schedule	Modify Individual Contact	Modify a subset of the policy information stored in the Notification Manager database for the individual associated with your current login.
Update Notification Policies	Update Notification Policies	Allow authorized users to access and update any aspect of Notification Manager Policy.

Menu System

The menu system at the top of each main page allows you to navigate the Notification Website. You can use this menu system to access any page contained within the same area of the website. For example, if you select the Update Notification Policies area from the home page, the menu system on the pages throughout that area let you navigate to any other page contained in that area.

Logging In

Before you can perform any Notification Manager functions from this website, you must log in using an approved Notification Manager login. The authority granted to your login determines which Notification Manager functions you can perform. For information on Notification Manager logins and the privileges associated with them, see [Secure the Notification Website](#) (see page 309) .

When you request to perform a Notification Manager function, you are automatically directed to the Notification Website Login page.

Method Information

For complete details about the notification methods provided by CA Automation Point, see the *Command and Keyword Reference Guide*.

Updating Notification Policies

You can update Notification policies using the PolicyInfo web page. To access this page, choose Update Notification Policies from the Notification Website home page.

From this the Notification Website home page, you can access any area of notification policy, including methods, contacts, and logins.

For more information on updating Notification Policies, see the Notification Website Help.

Troubleshooting the Notification Website

This section contains hints and tips about using the Notification Website.

Notifications Not Updated

Symptom:

I made a schedule change for a contact (added, changed, or deleted a schedule entry). Why didn't Notification Manager initiate notifications according to the updated schedule?

Solution:

The modifications to the contact were not saved. If you clicked OK after you added, changed, or deleted a schedule entry, the local copy of the schedule was refreshed, and you can see the updated schedule. However, you must click Commit to save the changes to the Notification Manager database. The Notification Website notifies contacts according to schedules that have been "committed" or saved in the Notification Manager database.

Methods Not Updated

Symptom:

I modified the parameter definitions for a method (added, changed, or deleted a parameter). Why didn't Notification Manager issue notifications using the updated method parameter definitions?

Solution:

The method modifications were not saved. You must click Commit to save the changes to the Notification Manager database. See the answer to the previous question for an explanation.

Notification Failure

Symptom:

I sent a notification to a contact that the Notification Website indicated was available. Why did the notification fail with an RC=6178?

Solution:

The contact schedule displayed is the active schedule as of the time you logged on to the Notification Website. If subsequent changes are made to the schedule, and a send notification is initiated, the contact is notified according to the currently active schedule. If no methods are defined for the current time, then the notification will fail with an RC=6178. To view the current contact schedule before sending a notification, log off and then log back on to the Notification Website. Jump to the Notify page and reselect the contact.

REXX Program Not Included in Invocation List

Symptom:

I created a REXX program in the site\myfiles\rexx directory to be invoked by a Notification Manager method. When I create or modify a method from the Notification Website, why doesn't the Notification Website include the new REXX program in the list of possible invocation programs?

Solution:

The list of invocation programs displayed from the Notification Website is the list of programs available as of the time you logged on to the Notification Website. If the REXX program was created after you logged on to the Notification Website, it will not be listed. To view the current list of invocation programs, log off and then log back on to the Notification Website. Jump to either Create Methods or Modify Methods and re-select the invocation program.

Login Message

Symptom:

I remained logged in to the Notification Website with no activity and was prompted to log back in again. What happened?

Solution:

You are logged out of the Notification Website after 30 minutes of inactivity.

RC=6040 Error

Symptom:

I attempted to update a contact schedule, but after clicking Commit, I received an RC=6040 error stating that the Times key specified does not exist.

Solution:

This situation can occur when multiple users are updating the same contact at the same time. Both users get the same data from the database, but once one user makes changes and commits them to the database, the data on the second user screen is out of date. If the second user commits changes, the updates may be to time blocks that no longer exist. To correct this error, return to the Modify Contact page and reselect the contact you are attempting to update. This obtains the latest information from the database. If there is still a problem at this point, contact your notification administrator.

Secure the Notification Website

The Notification Website allows you to take full advantage of the rich set of notification management features provided by CA Automation Point. You can make the website accessible within the company intranet and enable it as an open platform for authorized usage from anywhere, anytime.

The security measures that you take for the Notification Website can ensure many aspects of security and protection, while at the same time provide the widest user base with maximum functionality. To this end, the Notification Website provides these kinds of security:

- **Physical Security** – The CA Automation Point server and the web server are expected to run behind the corporate firewall within the company intranet on the company property. Your company controls any direct access to the physical site, the computers, and the software on them.
- **Data Encryption** –The data transmitted to and from the Notification Website is encrypted and protected. Your site's webmasters control the means for establishing secure data transmission mechanisms, such as SSL (Secure Sockets Layer), between the web server and the client browsers that are connected. The data transmitted between the web server and the AP server is encrypted using proprietary algorithms.
- **Authentication** – The Notification Website is protected from illicit users. The identity of a user is verified and confirmed before he or she is allowed to use the Notification Website.
- **Authorization** – The Notification Website prevents authenticated users from using the website in unauthorized ways. User privileges and permissions are checked before a legitimate user is allowed to access particular information or to use particular methods.

User Authentication

When a user attempts to enter the Notification Website, he or she is asked to log in by providing a user name and a password. Confirming the user's identity, or authenticating it, is the first security measure that is employed. This ensures that a user is who he or she claims to be. The user name and password are verified through an authentication service provider.

Notification Manager Website supports two forms of authentication:

- Windows authentication through a Windows domain server
- LDAP authentication through an LDAP-compliant directory server

You can forgo authentication altogether, but this option should be reserved only for abnormal or temporary situations, and only after you have thoroughly evaluated any potential security risks your site may be exposed to as a result.

Windows Authentication

Under Windows authentication mode, the login name and password supplied on a login dialog from the web page is passed back to the AP server, which then confirms the user's identity with a Windows domain controller. Sites that are maintaining their user logins and passwords for their Windows domains may prefer this option. Your site can choose whether to maintain the user account data on the local AP server or within a domain controller, and you can use standard operating system facilities for maintaining login names, passwords, and expiration policies. In either case, the login names and passwords that are authenticated under this mode are valid Windows user accounts.

LDAP Authentication

Under LDAP authentication mode, the login name and password supplied by the website user is confirmed with an LDAP-compliant directory server. Two types of sites may prefer this mode:

- Sites that are maintaining their user logins and passwords in their corporate directories
- Sites that are maintaining the user accounts data on mainframes and have LDAP access to these data centers

In either case, the login names and passwords that are authenticated are valid LDAP directory user accounts.

Note: The login names you create to be users of the Notification Website must directly correspond to either the Windows user accounts (if you use Windows authentication mode), or the LDAP directory user accounts (if you use LDAP Authentication mode). Otherwise, the authentication, if enabled, will fail.

How You Use NM Built-in Names

Notification Manager provides two built-in login names that you can use initially to set up your site, and use as a basis for subsequent transitions. These are NmAdmin and NmGuest. These names can be used both for authentication and authorization.

Note: For more information about authorization, see the section [User Authorization](#) (see page 312) in this chapter for details on authorization.

The built-in login names are described in the following table:

Login Name	Description	Privileges
NmAdmin	Administrative login name. Regardless of the authentication mode, the Notification Website checks the internally stored password to confirm this user.	When confirmed, this user can perform all operations available within the website.
NmGuest	Guest login name. Regardless of authentication mode, the Notification Website checks the internally stored password to confirm this user.	When confirmed, this user can perform authorized operations within the Notification Website. NMGuest has a set of privileges that are assigned by default. You can modify this login name through the Notification Website.

Important! We strongly recommend that you change the passwords of the built-in login names before you publish the website, and allow only limited access to these accounts. The built-in logins are especially necessary during startup, but if used improperly, they could endanger your security measures.

User Authorization

After a user is successfully authenticated, he or she can use the Notification Website to perform various tasks. The kinds of tasks and the degree to which the user can perform them depends on whether the user is properly authorized.

The fundamental question involved in user authorization is "who can do what to whom." In the case of the Notification Website user authorization, the *who* corresponds to the Notification Manager login object; the *whom* corresponds to the Notification Manager contact object; and the *what* corresponds to the operations available within the Notification Website. Such operations include the following:

- Notify someone by using the available Notification Manager notification methods
- Read schedule or contact information data
- Update schedule or contact information data
- Manage logins (create and delete login users within the Notification Website)
- Manage methods (create, modify and delete Notification Manager methods)

Authorization Modes

The Notification Website supports two modes of user authorization: a basic mode and an advanced mode.

Basic Privileges Mode—This mode is supported intrinsically by the Notification Website; that is, it can be managed using the Notification Website, and its data is stored internally by Notification Manager itself. Use this mode to manage broad privileges of a specific login that pertain to all contacts, such as "Joe can notify everybody."

Advanced Permissions Mode—This mode is supported by the Notification Website through the use of an external LDAP-compliant directory server. The Advanced Permissions mode allows you to specify detailed permissions of a specific login to a specific contact, such as "Joe can notify Jane." Your site can manage the advanced permissions data using any third-party, LDAP-compliant directory server product. The Notification Website queries this directory for the permissions data to achieve a fine degree of authorization control.

When and How Are These Modes Enabled?

The basic mode of user authorization is always enabled. You can enable or disable the advanced mode at your site. The Notification Website always checks the basic privileges data, and only checks the advanced permissions data if the Advanced LDAP Permissions mode is enabled.

The two modes of user authorization are evaluated according to a *combinatory-granting* rule; that is, authorizations that are granted to subjects from both authorization modes are combined. This means that an authorization is considered granted if it is granted in either the basic mode or the advanced mode, or in both; an authorization is considered denied if it is not granted at the basic mode level and not granted at the advanced mode (if that mode is enabled).

Amount of Control Exerted

The Basic Privilege mode gives you broad control over all contacts. For this reason, it is easier and quicker to set up. However, the Advanced Permissions mode gives you detailed and specific authorization capabilities by allowing access to external directory data. If your site wishes to curtail unauthorized access to the website as much as possible and is experienced with using directory structure, enabling these Advanced Permissions can provide you with a better and more scalable security solution.

If your site does not want to employ authorization controls to a fine degree, does not yet have a detailed security plan, or is relatively unfamiliar with directory use, you can choose to start up your site using just the Notification Manager's basic mode of authorization. Later, when you have formulated a security policy that meets your needs and you have gained knowledge and expertise with directories, you can choose to place the permissions data in the directory and enable the Advanced LDAP Permissions option with the Notification Website.

Basic Privileges Mode

The operations that can be authorized for a specific login user under the Basic Privileges Mode of authorization are described in the following table:

Operation	The specified login user can use the website to:
Notify All Contacts	Send a notification to any contact listed in the Notification Manager database.
Update Personal Schedule	Update schedule and contact information for the contact that is associated with the current login account.

Operation	The specified login user can use the website to:
Update Personal Groups	Update the group schedule and group membership information for groups of which the contact is a member. For this privilege, a user is considered a member of the group when their associated contact is defined in at least one group schedule.
View All Contacts	View schedule and contact information for any contact defined in the Notification Manager database
View All Notifications	View the status of all issued notification requests.
Answer All Notifications	Provide an answer for any pending notification request for which the ASK parameter was specified.
Manage All Contacts	Modify schedule and contact information of any contact listed in the Notification Manager database or create or delete any contact in the Notification Manager database.
Manage All Methods	Create, modify, or delete any methods in the Notification Manager database.
Manage All Logins	Create, modify, or delete any logins in the Notification Manager database, including changing the basic privileges of these logins.

In accordance with the overall authorization evaluation process, the basic privileges are evaluated according to the *combinatory-granting* rule. Authorizations that are granted to subjects from all the basic privileges are combined. As soon as the requested authorization is determined to be granted, the evaluation stops. For example, in the case of updating a personal schedule, an authorization is considered granted if it is granted either through the Update Personal Schedule privilege or the Manage All Contacts privilege.

Manage Basic Privileges

The Notification Manager intrinsically implements the Basic Privileges authorization mode, and stores the basic privileges data internally in the NM database. You can use the web pages on the Notification Website to manage the Basic Privileges user authorization.

From the Notification Website home page, you can follow the Update Notification Policies link to the Update Notification Policies web page where links are provided to let you create, modify, or delete a login name. For details on completing these tasks, see the online help.

Notes:

- Before you can manage these basic privileges, you must have the Manage All Logins privilege. For this reason, you may want to use the NM Built-in NmAdmin administrator login account during the early stages of the website setup.
- You must ensure that whatever login names and passwords you create correspond to the accounts that are to be authenticated with the external authentication services, either the Windows user account or the LDAP directory account.

By default, a newly created login has the same sets of privileges that the NM Built-in NmGuest guest login user has. The authorized administrator can use this login as a basis and adjust the privileges as appropriate for a particular login user. The authorized administrator can also adjust the privileges assigned to the "NmGuest" account to change the defaults for all future new logins.

Advanced Permissions Mode

The operations that can be authorized for a specific login user for a specific contact under the Advanced Permissions Mode of authorization are described in the following table:

Permission	The specified login user can use the website to...
Notify Contact	Send a notification to a specific contact
Read Contact	Read the schedule and contact information of a specific contact
Update Contact	Update or read the schedule and contact information of a specific contact

In accordance with the overall authorization evaluation process, the advanced permissions are evaluated following the *combinatory-granting* rule; that is, authorizations granted to subjects from all the advanced permissions, as well as from all the basic privileges, are combined, and as soon as the requested authorization is determined to be granted, the evaluation stops. For example, in the case of reading schedule information of a contact, an authorization is considered granted if it is granted either through the Update Personal Schedule basic privilege, by the Manage All Contacts basic privilege, by the Update Contact advanced permission, or by the Read Contact advanced permission.

Manage Advanced Permissions

The Notification Website uses an external LDAP-compliant directory service to support the Advanced Permissions authorization mode. The permissions data is stored externally in the directory. The external directory server product provides you with tools to administer and manage the directory and the data.

The Notification Website uses the LDAP protocol to communicate with the directory for the permissions data. This open standard allows CA Automation Point to provide you with the fine granularity of authorization control, without restricting your directory choice to any particular product or vendor. Please refer to the documentation supplied with your particular directory product for directory related technical guidance and specific assistance.

More information about the Lightweight Directory Access Protocol (LDAP) follows:

- A directory is a specialized database supporting hierarchical tree-like data structures.
- LDAP Runs over TCP/IP.
- Directories are typically optimized for large volume reading and searching.
- LDAP is an open standard that has wide industry support.
- Many vendors currently offer LDAP-compliant directory products and services, including CA, which offers CA Directory, and Microsoft Corporation, which offers Active Directory.

The Notification Website uses the LDAP protocol to communicate with the directory for the permissions data. This open standard allows CA Automation Point to provide you with the fine granularity of authorization control, without restricting your directory choice to any particular product or vendor. Please refer to the documentation supplied with your particular directory product for directory related technical guidance and specific assistance.

The following is a list of general actions you can take to manage the Advanced Permissions in your directory, assuming that you have successfully configured basic privileges for the login users and have successfully installed the directory on a server that the AP server has TCP/IP access to:

- Survey your website users to determine their responsibility hierarchy and decide who should have what access to whom. Use the survey to decide upon the hierarchical branching layout if one is necessary. Design permissions policies with a focus on long term success.
- Extend the directory with the CA Automation Point Notification Manager Advanced Permissions schema, using the tools provided by the directory product.

- Put the Advanced Permission data into the directory using the tools provided by the directory product.
- Enable the Advanced LDAP Permission in CA Automation Point.
- Gradually disable the basic privileges that were previously granted and can be replaced by the advanced permissions.

The following sections provide you with detailed information about the object definitions used to implement the Advanced Permissions for the website, the recommended branching layout in the directory information tree, and a few scenarios to illustrate how to set up your site.

NM LDAP Object Definitions

Entries, objectClasses, and Attributes

A directory contains data called *entries*, which contain information about particular objects. Each entry represents one or more *objectClasses* that defines the type of entry. An objectClass defines the *attributes* that are associated with it. Therefore, each entry is a collection of attributes with their values, and attributes that belong to the entry depend on the objectClass the entry is based on.

Schema

A directory uses a *schema* for the definitions of objectClasses and attributes. Directories are commonly shipped with a standard set of schema definitions such as country, organization, organizational unit, person, and so on.

OIDs

You can extend a directory's schema to suit your site's needs by allowing it to recognize and accept the objectClasses and attributes that are used to implement the Advanced Permissions authorization mode for the Notification Website. To allow for customization of directory schemas on-site, it is necessary to avoid any naming conflict among the objects defined by different entities. This is done through the use of guaranteed unique numbers called Object Identifiers (OIDs), each of which unambiguously identifies an objectClass or attribute. These Object Identifiers are guaranteed to be unique across all networks worldwide.

The OIDs form a hierarchy, as shown in the following example:

1	ISO- The root authority
1.3	ISO Identified Organization
1.3.6	US Department of Defense
1.3.6.1	Internet OID assignments
1.3.6.1.4	Internet Private

1.3.6.1.4.1 IANA – Registered Private Enterprises
 1.3.6.1.4.1.791 CA, Inc.

The Notification Manager facility is assigned OID 1.3.6.1.4.1.791.2.10.5.3. All OIDs used in our attributes and objectClasses stem from this root OID. This grants unique identification, regardless of what directory product you are using.

NM Attributes Definitions

The following table shows the attribute definitions used for the Advanced Permissions.

Attribute Name and OID	Description	Syntax
nmLoginName apnmOID.1.1 (1.3.6.1.4.1.791.2.10.5.3.1.1)	NM login name for whom the permissions are being specified This is also used to construct the NM login team branch in your directory	Case-insensitive string or caseIgnoreString string
nmLoginTeam apnmOID.1.2 (1.3.6.1.4.1.791.2.10.5.3.1.2)	A directory in the hierarchy entry containing all login names for whom the contact privileges are being specified	DistinguishedName
nmNotifyContacts apnmOID.1.3 (1.3.6.1.4.1.791.2.10.5.3.1.3)	NM contact name who may be notified	Case-insensitive string or a caseIgnoreString string
nmReadContacts apnmOID.1.4 (1.3.6.1.4.1.791.2.10.5.3.1.4)	NM contact name whose schedule and contact information may be read	Case-insensitive string or a caseIgnoreString string
nmUpdateContacts apnmOID.1.5 (1.3.6.1.4.1.791.2.10.5.3.1.5)	NM contact name whose schedule and contact information may be updated	Case-insensitive string or a caseIgnoreString string
nmNotifyContactTeams apnmOID.1.6 (1.3.6.1.4.1.791.2.10.5.3.1.6)	A directory entry in the hierarchy containing all contact names that may be notified	DistinguishedName
nmReadContactTeams apnmOID.1.7 (1.3.6.1.4.1.791.2.10.5.3.1.7)	A directory entry in the hierarchy containing all contact names whose schedule or contact information may be read	DistinguishedName

Attribute Name and OID	Description	Syntax
nmUpdateContactTeams apnmOID.1.8 (1.3.6.1.4.1.791.2.10.5.3.1.8)	A directory entry in the hierarchy containing all contact names whose schedule or contact information may be updated	DistinguishedName
nmContactName apnmOID.1.9 (1.3.6.1.4.1.791.2.10.5.3.1.9)	Notification Manager contact name. This is used to construct the Notification Manager contact team branch in your directory.	Case-insensitive string or a caseIgnoreString

NM Object Class Definitions

The following table shows the objectClass definitions used for the Advanced Permissions.

ObjectClass and OID	Description and Attributes	Structure
apnmLoginRights apnmoid.2.1 (1.3.6.1.4.1.791.2.10.5.3.2.1)	<p>Defines entries representing the NM login user. Contains attributes that specify the NM authorization permissions</p> <p>Mandatory attributes: commonName</p> <p>Optional attributes: nmLoginName nmLoginTeam nmNotifyContacts nmReadContacts nmUpdateContacts nmNotifyContactTeams nmReadContactTeams nmUpdateContactTeams</p> <p>Multiple values may be set for any of these attributes.</p>	<p>This is a structural object subclassed from the top. An apnmLoginRights entry may be placed under an organizationalUnit entry in the directory.</p> <p>For more information on the directory hierarchy, see the section How You Extend the Schema (see page 321).</p>

ObjectClass and OID	Description and Attributes	Structure
apnmLoginPerson apnmoid.2.2 (1.3.6.1.4.1.791.2.10.5.3.2 .2)	<p>Defines entries representing NM login user. Use to construct the NM login team branch in your directory.</p> <p>Mandatory attributes: commonName nmLoginName</p> <p>Multiple values may be set for either of these attributes.</p>	<p>This is a structural object subclassed from top.</p> <p>An apnmLoginPerson entry may be placed under an organizationalUnit entry in the directory.</p> <p>For more information on the directory hierarchy, see the section How You Extend the Schema (see page 321).</p>
apnmContactPerson apnmoid.2.3 (1.3.6.1.4.1.791.2.10.5.3.2 .3)	<p>Defines entries representing the NM contact user. Use to construct the NM contact team branch in your DIT.</p> <p>Mandatory attributes: commonName nmContactName</p> <p>Multiple values may be set for either of these attributes.</p>	<p>This is a structural object subclassed from top.</p> <p>An apnmContactPerson entry may be placed under an organizationalUnit entry in the directory.</p> <p>For more information on the directory hierarchy, see the section How You Extend the Schema (see page 321).</p>

Note: A special "everyone" contact name can be used for attributes that specify permissions for an NM contact user. These attributes include nmNotifyContacts, nmReadContacts, and nmUpdateContacts. This special value is recognized internally by CA Automation Point, and can be used to indicate the broad permissions a login user may have.

How You Extend the Schema

Important! Extending the directory schema is an advanced and complex operation with far-reaching implications. Different directory products have different syntax and methods for modifying and extending the schema. Before you do this, be sure to consult your directory's documentation.

CA Automation Point provides three sample files to help you extend the directory schema. They are:

- `nm_ldap_schema_for_eTrust.txt`

If you are using the CA Directory product, you can use this file as a sample eTrust .dxc configuration script file.

- `nm_ldap_schema_for_MSAD.txt`

If you are using the Microsoft Corporation Active Directory product, you can use this file as a sample LDAP Data Interchange Format (LDIF) script file. Alternatively, you can use the Active Directory Schema MMC Snap-in to extend the schema.

- `nm_ldap_schema_for_apacheds.txt`

If you are using the Apache Directory server product, you can use this file as a sample LDAP Data Interchange Format (LDIF) script file.

You can find these samples in the directory `installDir\SAMPLE\NM\WebSecurity`.

The NM LDAP Directory Hierarchy

Tree Structure

Directory data is arranged in a hierarchical tree-like structure. This data tree is called a directory information tree (or DIT). A node of this tree is an entry. Each entry is identified by a Distinguished Name (DN).

For more information on LDAP terminology, see the section [NM LDAP Object Definitions](#) (see page 317) in this chapter.

The NM Base DN Entry

In the tree structure, a single base organizational unit directory entry called the NM Base DN entry, identifies the root of the NM LDAP tree within the larger directory. All NM-related advanced user permissions data resides under this base DN. You must establish this base entry first by creating an organizational unit entry, and use Configuration Manager to provide your settings to CA Automation Point. All NM searches begin from this root entry in your directory.

Setting Up Your Site

Under the base directory, `apnmLoginRights` entries are stored. These entries contain advanced user permissions data. For each NM login user, you can create an `apnmLoginRights` entry and specify the NM contacts that the login can notify, read, or update. You do this by specifying the attributes `nmNotifyContacts`, `nmReadContacts`, and `nmUpdateContacts` with the appropriate NM contact names.

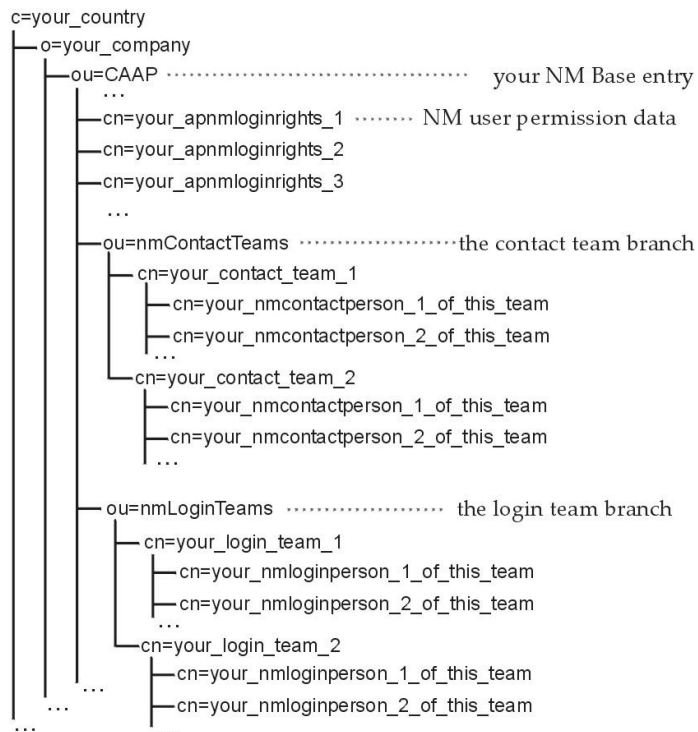
Also under the base directory, you can optionally establish two subdirectories, and use them to assign permissions to many logins, and/or for many contacts. For example, you can create an organizational unit entry named `nmContactTeams` and put all `apnmContactPerson` entries together into teams of contacts under this entry. You can also create an organizational unit entry named `nmLoginTeams` and put all `apnmLoginPerson` entries together into teams of logins under this entry.

Constructing contact teams allows you to assign user permissions to a specific login for all the contacts under the particular team DN hierarchy for whom the login can notify, read, or update. You do this by filling the attributes of `nmNotifyContactTeams`, `nmReadContactTeams`, and `nmUpdateContactTeams` with the corresponding contact team DN value.

Constructing login teams allows you to assign user permissions to a team of logins. You do this by filling the attribute `nmLoginTeam` with the corresponding login team DN values. All logins specified under the particular team DN hierarchy are given the permissions at once.

Using this directory structure, your site can explicitly set the permissions that each login name has for each contact. Your site can also organize the contacts and logins into teams and assign the permissions to these teams at the same time. The number of data entry tasks varies depending on the approach you take, but you are able to maintain precise control, regardless of approach.

The following is a partial sample directory structure:



NM LDAP Scenarios

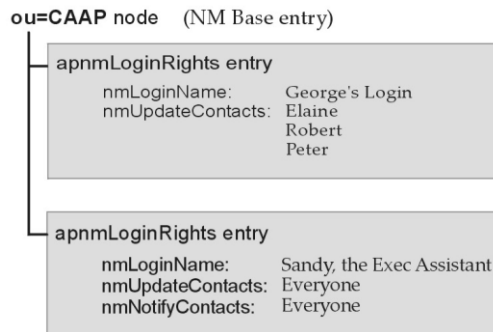
This section provides some examples of possible ways to arrange advanced permission data in the LDAP tree based on the nature of your site and the scale of your Notification Website. In the sample organization used throughout these examples, the following names are used:

- Brian – Head of the division in the company; oversees Marketing and Sales
- Sandy – Brian's assistant
- George – Directory of Marketing
- Elaine, Robert, and Peter – Marketing staff
- Julie – Manager of Sales
- Don and Sam – Sales staff
- Mike and Jake – Technicians in the IS department; IS is responsible for maintaining the computer resources throughout the company.

Scenario 1

Assign permissions for individual contacts to individual logins.

A site can easily assign permissions for individual contacts to individual logins by creating new entries of `apnmLoginRights` objectClass in the LDAP tree. In the following scenario, George, the marketing director can read and update contact information for all the marketing staff, including Elaine, Robert and Peter. Sandy, Brian's assistant, can notify and help maintain contact information for every NM contact, even though she may not have a corresponding contact in the NM database.

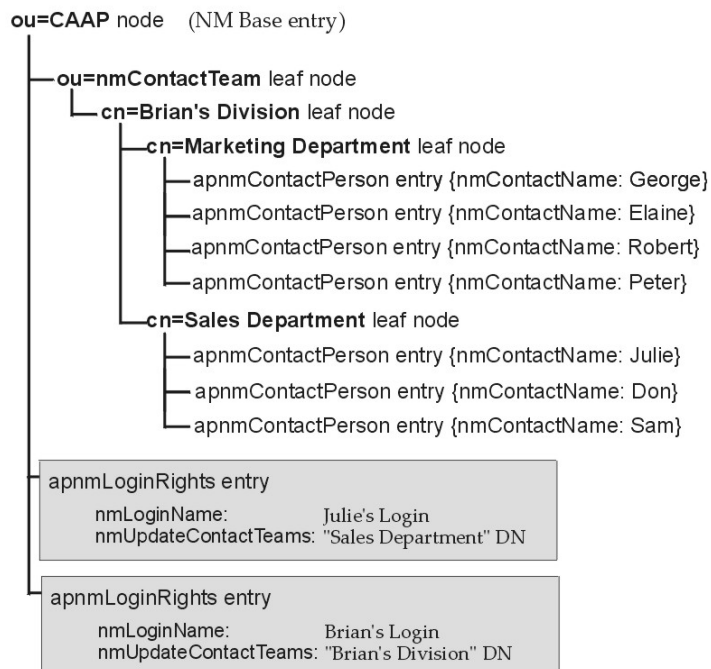


Scenario 2

Assign permissions for many contacts.

By using the LDAP hierarchical tree-like structure to organize the contacts into teams, and assigning permissions for these contact teams as a whole, a site can easily manage permissions of many contacts.

In the following scenario, Julie, the sales manager can access contact information for everyone in the sales department. Brian, the head of the division, can read and update contact information for everyone in marketing and sales departments. Now, the simple act of adding Sam, another support engineer, to the sales department automatically gives both Julie and Brian the contact permissions for him.

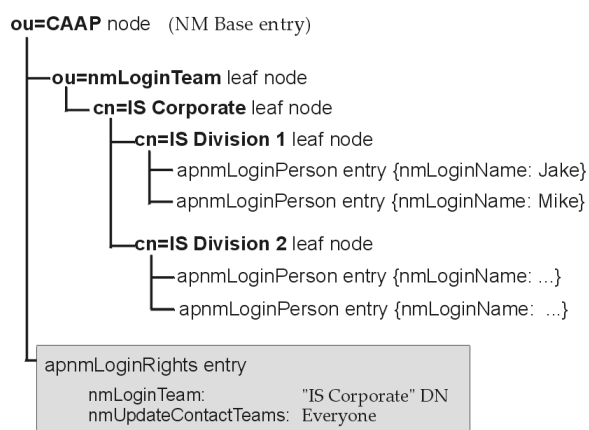


Scenario 3

Assign privileges to many logins.

Use the LDAP tree structure to organize logins into teams and assign permissions to these login teams as a whole. This allows you to easily manage permissions of many logins.

In the following scenario, all IS people, including Jake, Mike, and the technicians in "Division 2," can maintain contact information for every NM contact. If Jake hires Fred tomorrow and adds Fred to the IS Division 1 branch in the LDAP tree, Fred automatically gets all the permissions his group has.



LDAP Resources

The following LDAP Request for Comments (RFCs) are listed for your reference.

- RFC-2251 Lightweight Directory Access Protocol (v3) (December 1997)
- RFC-2252 Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions (December 1997)
- RFC-2253 Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names (December 1997)

The following websites provide more information on LDAP standards and Directory management:

- The International Standard Organization website at <http://www.iso.ch/>
- The Internet Engineering Task Force website at <http://www.ietf.org/>
- The CA Directory page at the [CA Support Online](#) website
- The Active Directory Programmer's Guide at the Microsoft website at <http://www.microsoft.com/>
- The OpenLDAP project, an "Open Source" implementation of LDAP, has its website at <http://www.openldap.org/>

Understanding Notification Manager Log Files

During the operation of Notification Manager, several log files are used to record progress and status information. These log files reside in the %AP_DATA%\Logs directory and are described in detail in the following sections.

VOXNM.LOG

REXX programs that comprise Notification Manager use this log file to record progress information on notification requests. This progress information includes the parameters used to launch the various REXX programs and the return codes from each REXX program. This log file has a maximum size of 4MB, and the logging system retains 10 previous copies of this log file. Each log message recorded in this file begins with a common prefix, which is formatted into the following fields:

```
YY/MM/DD HH:MM:SS SYSNAME YY/MM/DD HH:MM:SS.mmm DOMAIN\USER PID/TID  
PROGRAM:UNIQUEID
```

YY/MM/DD

Specifies the year, month, and day as recorded by the Notification Server.

HH:MM:SS

Specifies the time (in hours, minutes, and seconds) as recorded by the Notification Server.

SYSNAME

Specifies the system name (PPQ name) of the VOX client from which the request was initiated.

YY/MM/DD

Specifies the year, month, and day as recorded by the VOX client from which the request was initiated.

HH:MM:SS.mmm

Specifies the time (in hours, minutes, seconds, and milliseconds) as recorded by the VOX client from which the request was initiated.

DOMAIN\USER

Specifies the user account (in Windows domain format) used to initiate the request on the VOX client.

PID/TID

Specifies the process ID (PID) and thread ID (TID) from which the initial request was generated on the VOX client.

PROGNAME

Specifies the name of the REXX program executing on the VOX client that initiated the log request

UNIQUEID

Specifies the unique identifier used to represent the current instance of the running REXX program. This unique ID is used to differentiate between different running instances of the same REXX program executing on the same VOX client machine

WEBNM.LOG

This log file is used by the Notification Manager Web Gateway Service (NM Gateway Server) to record activity initiated from the Notification Website. This logged activity includes the following:

- Changes made to notification policy data (using the Notification Website)
- Starting and stopping of the NM Gateway Server service
- An audit trail of notification requests issued from the Notification Website

For this log file to be used, you must enable notification policy change logging on the NM Website dialog contained within the Configuration Manager application. After you have enabled this change-logging feature, you can adjust the logging parameters (log file maximum size and number of log files to retain) using the same Configuration Manager dialog. Each log message recorded in this file begins with a common prefix, which is formatted into the following fields:

YY/MM/DD HH:MM:SS (user:function)

YY/MM/DD

Specifies the year, month, and day the logged action was recorded by the NM Gateway Server.

HH:MM:SS

Specifies the time (in hours, minutes, and seconds) the logged action was recorded by the NM Gateway Server.

user:function

Specifies the NM login account used to initiate the logged action, and the name of the function currently being performed. If the logged action was the result of an internal system request instead of a login user request, the user name "SYSTEM" is displayed.

CAAPNFY.LOG

This log file is used by the Notification Manager web application that executes inside of the JSP/Servlet environment selected to host the Notification Website. The main purpose of this log file is to provide diagnostic information on the execution of the server-side Java programs used to dynamically create the web pages contained within the Notification Website. If the redistributed Apache Tomcat JSP/Servlet environment is used to host the Notification Website on the local CA Automation Point server machine, this log file will reside in the %AP_DATA%\Logs directory. The %AP_SITE%\Config directory contains a configuration file called caapnfy.lcf that controls logging parameters, such as those that control the amount of information written to the caapnfy.log file, the maximum size of this log file, and the number of previous log files to retain. Do not modify these settings in the caapnfy.lcf file unless you are instructed to do so by CA Technical Support. Each log message recorded in this file begins with a common prefix, which is formatted into the following fields:

YYYY-MM-DD HH:MM:SS,mmm LEVEL (MODULE:FUNCTION[:HOST])

YYYY-MM-DD

Specifies the year, month, and day that the logged action was recorded by the Notification Manager web application.

HH:MM:SS,mmm

Specifies the time (expressed in hours, minutes, seconds, and milliseconds) the logged action was recorded by the Notification Manager web application.

LEVEL

Specifies the logging level assigned to the logged action (ERROR, WARN, INFO, DEBUG).

MODULE

Specifies the name of the source code module

FUNCTION

Specifies the function name from which the log message was generated

HOST

Specifies the name of the machine viewing the Notification Website (using an approved web browser). This information is recorded only if this log message was written as the result of an action taken by a user on the Notification Website.

JolBeep Panel Emulation

CA Automation Point interfaces with CA FAQs ASO for z/VSE to monitor the JolBeep panel and to send notifications based on conditions that appear there.

To use the CA FAQs ASO for z/VSE interface to monitor the JolBeep panel

1. Install the complete CA Automation Point product.
2. Define to Notification Manager all the people and groups that you want Jolbeep to be able to contact. You do this using the Contacts pages on the Notification Website. (See the section [How You Use the Notification Website](#) (see page 304) in this chapter.) The names you define must match the names that appear in the PC Call List column of the PCS J panel. (The comparison is *not* case-sensitive.)
3. Define to Notification Manager a person named "JolBeep Mainframe Down."

Note: The person you define will be notified when the mainframe goes down or when the emulation is not working. When this person is notified, he or she must restore the CA Automation Point JOLBEEP session back to a state where the PCS J panel is working again. At that point, the emulation will start running again.

4. Create a session for the JOLBEEP mainframe session. In Configuration Manager, choose Expert Interface, Automation, Session Definition Sets. Right click on Sessions, and choose Add.
5. Define the session using the Automation Point Session Definition dialog.
 - a. Click Customize Session Settings.
 - b. Click Local Session Settings. In the dialog that displays, specify @E in the Restart String field. Click OK.
 - c. Click OK on the Automation Point Session Definition dialog.
6. Write a script that logs on to the JolBeep PCS J panel.
7. Name this script JOLBEEP.SCR and place it in the %AP_SITE%\MyFiles directory.
8. In Configuration Manager, choose Expert Interface, Automation, REXX, REXX Settings. On the REXX Settings dialog, browse to and select the JOLBEEP.REX program and add it as an "Initialization REXX Program" in your session definition set.

You have completed the configuration.

Chapter 12: Managing CMOS Processors

CA Automation Point's Hardware Access Facility (HAF) and the APCMOSI REXX program are remote HMC console applications that interact with the Hardware Management Console (HMC) or Service Element (SE) from the Automation Point Desktop environment. Use HAF and the APCMOSI REXX program to manage CMOS processors from the Automation Point Desktop environment. This chapter discusses the following:

- Establishing connectivity between CA Automation Point server and the HMC/SE.
- Establishing the HMC Console Application Programming Interfaces (API) runtime environment on the CA Automation Point server.
- Issuing HMC console commands using the APCMOSI REXX program.
- Monitoring HMC status messages using HAF

CMOS Processors

With the introduction of CMOS processors by IBM, the technology used for the hardware processor console changed from the legacy 3270-type console that was used with bipolar processors to a workstation application called the HMC. IBM System zSeries mainframe systems are examples of CMOS processors.

Hardware Management Console (HMC)

The Hardware Management Console (HMC) is a Linux workstation application featuring a graphical user interface (GUI) for operator access and console application programming interface (API) support for local and remote console applications. The console API allows programmatic access for performing system maintenance for HMC-managed mainframe systems automatically.

The HMC controls and manages one or more mainframe systems through a local area network connection (LAN) to the associated Service Element (SE) for each mainframe system.

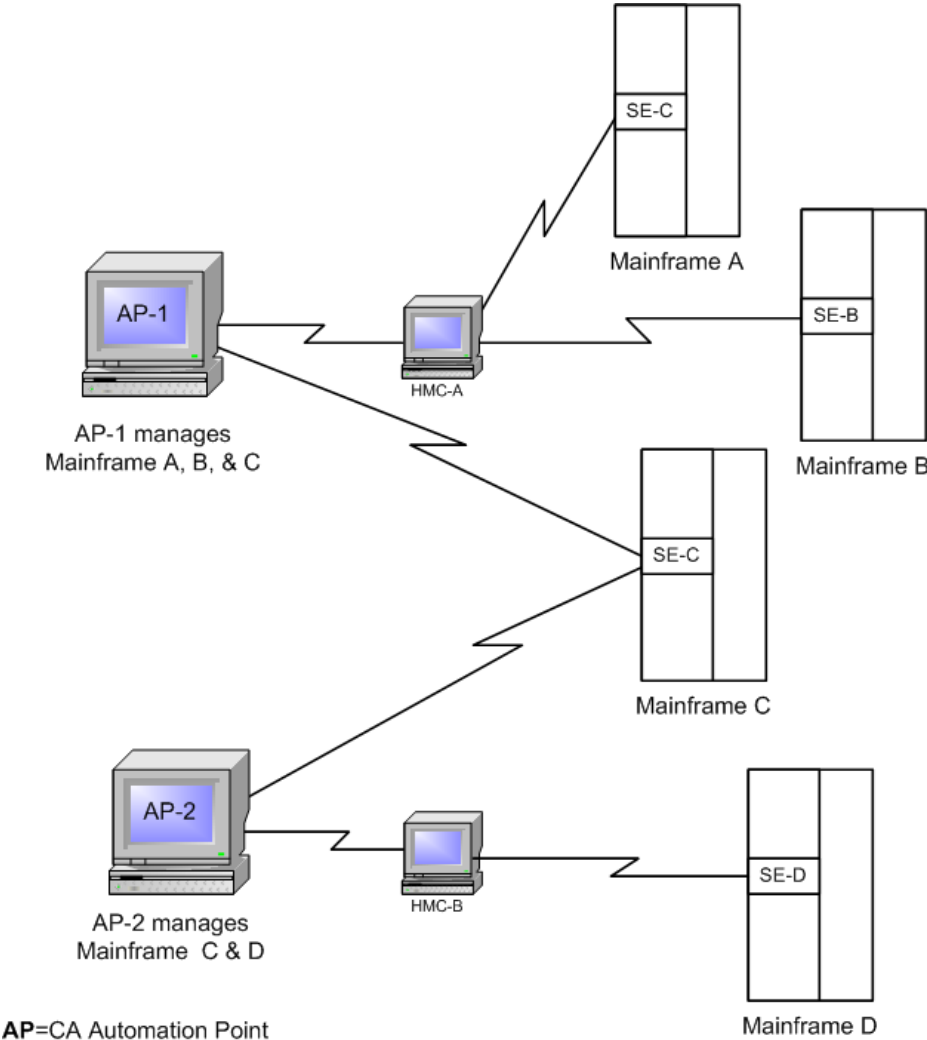
Service Elements (SE)

The Service Element is the hardware component attached to a single mainframe system. It is used to control and manage the associated mainframe system, as well as serve as the interface between the local area network (LAN) and the mainframe system. The SE provides GUI dialogs for operator access that performs the same functions as those of an HMC. The same console API that is supported on the HMC is also supported on the SE. Therefore console applications that function with the HMC function with the SE as well.

Communicating with CMOS Processors

All remote HMC console applications such as CA Automation Point must communicate with CMOS processors through the associated HMC or SE. After communications between CA Automation Point and the HMC or SE are established, you can direct IPL type commands either manually or automatically from the Automation Point Desktop environment to any of the mainframe systems that the HMC or SE manages. You can also receive HMC/SE status messages for monitoring by CA Automation Point. If you are operating with more than one HMC, you can configure CA Automation Point to communicate with multiple HMC workstations.

The following illustration show how CA Automation Point connects to HMC and SE to manage mainframe machines.



Establishing Connectivity with the HMC/SE

Accomplishing all the following steps to achieve HMC connectivity may require that you contact and work with your Network Administrator or Security Administrator.

For more information about configuring SNMP for use between CA Automation Point and your HMC, see the IBM manual SB10-7030: System z Application Programming Interfaces. Specifically, Chapter 6, "Configuring for the data exchange APIs," gives detailed a description of SNMP configuration.

The HMC/SE communicates using the SNMP network protocol. You need to configure the CA Automation Point server and the HMC/SE to send and receive SNMP traps between each other.

Note: Accomplishing all of the following steps to achieve HMC connectivity may require that you contact and work with your Network Administrator or Security Administrator.

For more information about configuring SNMP for use between CA Automation Point and your HMC, see the IBM manual "SB10-7030: System z Application Programming Interfaces". Chapter 6 "Configuring for the data exchange APIs" gives detailed a description of SNMP configuration.

To establish connectivity with the HMC/SE

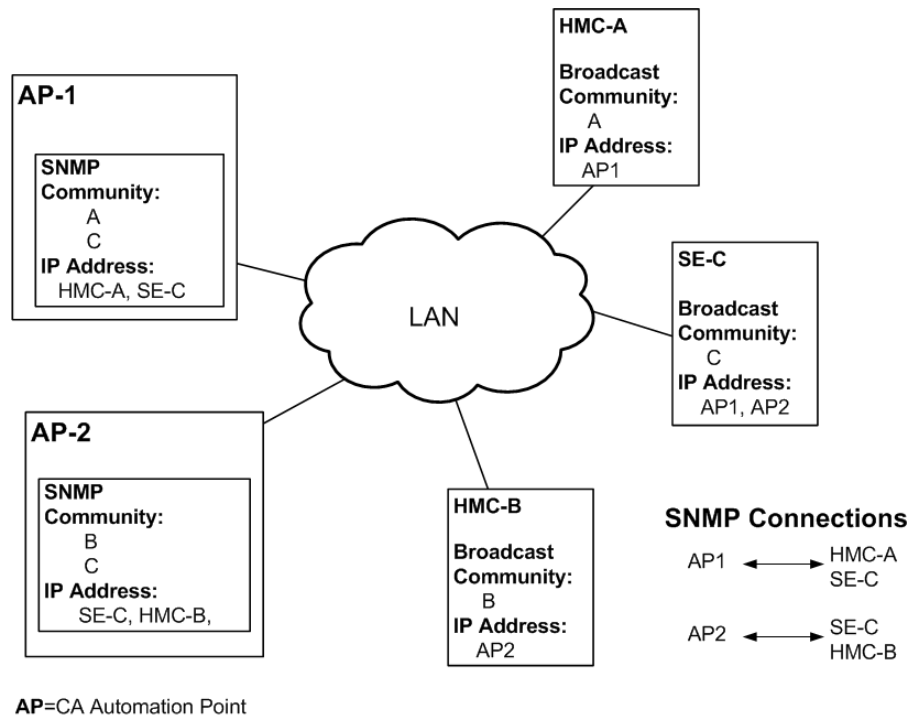
On the local CA Automation Point server:

1. Install Windows SNMP Service.
2. Determine the community name and TCP/IP address for every HMC/SE that CA Automation Point must communicate with. Note the community name and TCP/IP address for each HMC/SE.
 - a. The community name and TCP/IP address for the target HMC/SE are required parameters for each APCMOSI Rexx program call.
 - b. The community name and TCP/IP address for each monitored HMC/SE is required for configuring HAF under Configuration Manager.
3. Configure the Windows SNMP Service through the additional tabs on the properties for the SNMP service. From the Traps tab, configure SNMP service to receive and initiate traps from the list of HMC/SE community names and TCP/IP addresses with which CA Automation Point must communicate.

Note: Community names are case sensitive.

On each HMC/SE workstation:

1. Determine the community name and TCP/IP address for each CA Automation Point server with which the HMC/SE must communicate.
2. Add the CA Automation Point servers to the broadcast list for HMC/SE using the community name and TCP/IP address for each CA Automation Point server.



Establishing Runtime Environment for HMC Console API

CA Automation Point's Hardware Access Facility (HAF) and the APCMOSI Rexx program are remote console applications that interact with the HMC/SE from the Automation Point Desktop environment. As a remote HMC console application, CA Automation Point must have runtime access to the file named, hwmcawin.dll, an IBM provided Dynamic Link Library (DLL) that provides underlying support for the HMC console API. hwmcawin.dll is a 32-bit Windows dynamic link library containing the C language Data Exchange APIs and Commands API.

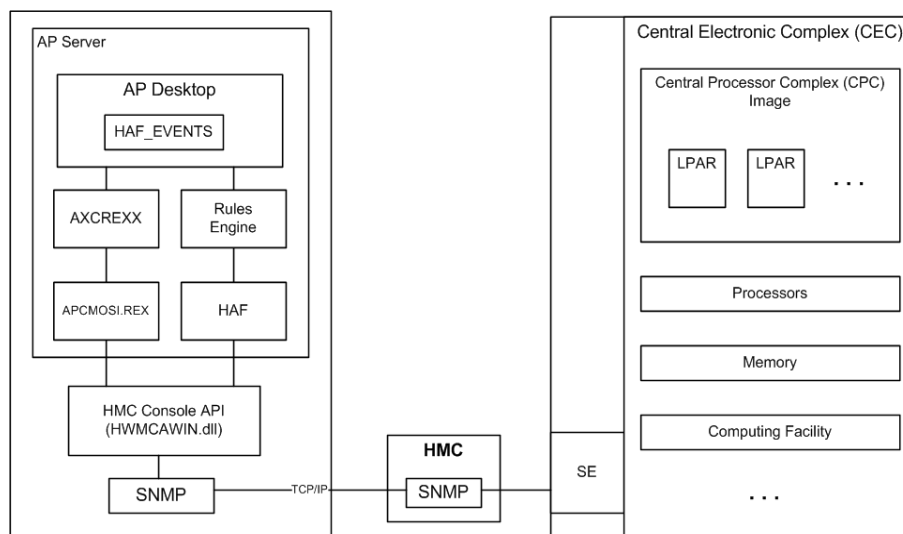
To prepare the runtime environment on the CA Automation Point server

1. Download the most current copy of the DLL file, hwmcawin.dll, from the IBM Resource website, www.ibm.com/servers/resourcelink onto the CA Automation Point server. The HMC/SE is also expected to run with a compatible version of the DLL files.
2. The DLL file must be accessible by the APCMOSI Rexx program at runtime. Copy the hwmcawin.dll file into a directory that is part of the search path. Note the directory where the hwmcawin.dll file is copied. The directory and file name are required for configuring HAF using Configuring Manager.

- Whenever you upgrade the code level of your HMC/SE, also upgrade the copy of the DLL file residing on the CA Automation Point server.

For information on how to configure the HMC to use the console API, see the *IBM System z Application Programming Interfaces* manual (SB10-7030-11).

The following illustration shows how CA Automation Point supports the HMC Console API.



AP=CA Automation Point

Using CA Automation Point to Manage CMOS Processors

CA Automation Point's remote HMC console applications, HAF and APCMOSI REXX program, can be configured to interact directly with the HMC/SE using the HMC console API over an SNMP connection. CA Automation Point can be used to interface with an HMC in the following ways:

- Through a HAF-configured session, named HAF_EVENTS, you can monitor all messages or state changes that occur on any number of HMC/SE workstations or the mainframe system objects that they control.
- Through the APCMOSI REXX program, you can query or set the state of any number of HMC/SE workstations and the mainframe system objects that they control.
- Through CA Automation Point message ID rules, you can invoke APCMOSI REXX program to automate state changes in the HMC/SE workstations or the mainframe system objects that they control.
- Through CA Automation Point time rules, you can make scheduled state changes to the HMC/SE workstation or the mainframe system objects that they control.

APCMOSI REXX Program

APCMOSI.REX is a CA Automation Point-provided sample program that can be initiated from the Automation Point Desktop environment to programmatically perform all actions on mainframe systems that can be manually initiated by the operator through the controlling HMC/SE. Submitting HMC console commands using the APCMOSI.REX program provides you the ability to remotely automate console commands using standard CA Automation Point facilities. When initiating HMC console commands from CA Automation Point, you'll be able to integrate or coordinate HMC console automation with other monitored session activities.

Note: APCMOSI.REX is a working sample that can also be customized as needed. Before APCMOSI.REX can be started from within the Automation Point Desktop environment, you must copy the program, APCMOSI.REX from *installDir*\sample\APCMOS to %AP_SITE%\site\myfiles\REXX directory. For syntax and detailed description of the sample APCMOSI.REX program, reference embedded comments in the program source. Reference the *apcmosi.txt* file found in the *installDir*\sample\APCMOS directory for most current details about the APCMOSI.REX program.

Submitting HMC commands with APCMOSI REXX Program

As a REXX program, with APCMOSI.REX, you'll be able to initiate the HMC commands targeted for the mainframe system in the same ways any REXX program can be started within the Automation Point Desktop environment.

- Manually initiated by operator
 - By selecting a user-customized menu option to start the APCMOSI.REX program
 - By selecting Start REXX menu option and entering APCMOSI program call with operator specified parameters into the host command area
- Automatically initiated by CA Automation Point
 - A rule triggered to start the APCMOSI.REX program.
 - A running REXX program, programmed to start the APCMOSI.REX program.

APCMOSI Command Syntax

Given the HMC console command and associated parameters, APCMOSI.REX directs the request and data as specified to a designated HMC/SE for execution using the SNMP protocol.

The command has the following format:

```
APCMOSI {ACTION(hmc_command) [{additional_argument(setting) ...}] |  
ACTION(GetAttribute) ATTRIBUTE(attribute_name) |  
ACTION(SetAttribute) ATTRIBUTE(attribute_name) VALUE(attribute_value) |  
ACTION(GetStatus) }  
CPC(cpc_name)  
COMMUNITY(hmc_community)  
[LPAR(lpar_name)]  
TARGET(hmc_tcpip_address)
```

ACTION

Specifies the HMC command to perform.

GetAttribute is an APCMOSI supported ACTION that is not a direct console command or action specified in the HMC Console API. ACTION(GetAttribute) returns the value of a specified attribute for a specified CPC or LPAR. The value of the returned attribute appears in a text message that is written to the default output stream.

SetAttribute is an APCMOSI supported ACTION that is not a direct console command or action specified in the HMC Console API. ACTION(SetAttribute) sets the value for a specified attribute for a specified CPC or LPAR.

Note: Some attributes are read-only and therefore cannot be set.

GetStatus is an APCMOSI supported ACTION that is not a direct console command or action specified in the HMC Console API. ACTION(GetStatus) returns a status indicator for a specified CPC or LPAR. The status appears in a text message that is written to the default output stream. For descriptions of returned status, see the *IBM System z Application Programming Interfaces* manual (SB10-7030-11) .

For a full list of possible ACTION parameter values that are supported by the APCMOSI command, see the *apcmosi.txt* file found in *installDir\sample\APCMOS* directory. For a detailed description of the HMC command, see the *IBM System z Application Programming Interfaces* manual (SB10-7030-11).

Note: To simplify the APCMOSI program call, APCMOSI accepts the console command without the standard prefix or suffix. Therefore, the HMC commands as described in the *IBM System z Application Programming Interfaces* manual (SB10-7030-11), are listed under the full name, which includes the prefix and suffix.

For example,

ACTION(RESETNORMAL) is described under HWMCA_RESETNORMAL_COMMAND

Note: APCMOSI.REX may be customized to add support for HCM commands that are not already supported by APCMOSI or listed in the apcmosi.txt file.

[{additional_argument(setting) ...}]

Specifies additional command arguments and their settings, if required, associated with the command specified by the ACTION parameter. For a list of possible additional argument values that are supported by the APCMOSI command for ACTION specified, reference the apcmosi.txt file found in *installDir\sample\APCMOS* directory. For a detailed description of command arguments applicable to the specified action, see the *IBM System z Application Programming Interfaces* manual (SB10-7030-11).

ATTRIBUTE

Specifies the name of the attribute that is subject to either the SetAttribute or the GetAttribute action. For the names of attributes that can be retrieved or set, see the *IBM System z Application Programming Interfaces* manual (SB10-7030-11).

Note: To simplify the APCMOSI program call, APCMOSI accepts attribute names with or without the prefix and suffix.

For example,

ATTRIBUTE(weight) is equivalent to ATTRIBUTE(hwmca_weight_suffix)

COMMUNITY

Specifies the community name of the HMC/SE workstation to which the HMC command specified in ACTION parameter is directed.

Note: This value is case sensitive.

CPC

Specifies the name of the Central Processing Complex (CPC) to which the HMC command specified in ACTION parameter is directed.

LPAR

Specifies the name of the Logical Partition (LPAR) within a given CPC where each partition supports one copy of the z/OS or other operating system to which the HMC command specified in ACTION is directed.

TARGET

Specifies the TCP/IP address of the HMC/SE workstation to which the HMC command specified in ACTION parameter is directed.

VALUE

Specifies the value the SetAttribute action sets for a specified attribute.

Follow these rules when invoking the APCMOSI command:

- Keywords and values are not case-sensitive.
- Parameters shown in brackets ([]) are optional.
- Parameter names can be abbreviated to the minimum number of characters that will uniquely identify the parameter. For example, if LPAR and CPC are the only required parameters, you can specify L(lpar1) and C(cpc1) in place of LPAR (lpar1) and CPC(cpc1). However if Action and ActivationProfile are allowable parameters, you must specify at least ACTION(pswrestart) and ACTIV(activation-profile-name).
- There are no default values for the parameters. The APCMOSI.REX program can be customized to set default values for parameters. If you set default values within the APCMOSI.REX program, you do not have to specify the parameter when invoking the APCMOSI command. However, you can still override the default values by specifying the parameter with overriding value. For example, if you have only one HMC/SE workstation or CPC, setting default values for Community, CPC, and Target parameters in APCMOSI.REX will simplify required parameter settings for APCMOSI program call to just the ACTION parameter.

APCMOSI Return Information

Upon successful completion of the HMC console command API, APCMOSI sets the REXX variable RC to 0 (zero). Non-zero return codes incurred while executing the HMC console command API are returned by APCMOSI in the REXX variable RC. For a detailed description of each return code and directions for resolving error, see the *IBM System z Application Programming Interfaces* manual (SB10-7030-11).

Examples:

Given the community name and TCP/IP address for the HMC/SE controlling the CPC named cpc1, issue the APCMOSI command to get the value of the processor weight attribute for LPAR named lpar1. For the name of attributes that can be retrieved, see the *IBM System z Application Programming Interfaces* manual (SB10-7030-11).

```
APCMOSI COMMUNITY(public) TARGET(101.102.103.104) CPC(cpc1) LPAR(lpar1)
ACTION(GetAttribute) ATTRIBUTE(weight)
```

Issue the APCMOSI command to set the weight attribute to 10.

```
APCMOSI COMMUNITY(public) TARGET(101.102.103.104) CPC(cpc1) LPAR(lpar1)
ACTION(SetAttribute) ATTRIBUTE(weight) VALUE(10)
```

Given the community name and TCP/IP address for the HMC/SE controlling the CPC named `cpc1`, issue the APCMOSI command to activate the LPAR named `lpar1` using the default activation profile. For acceptable syntax for specifying additional arguments required by the activate action, see the `apcmosi.txt` file found in `installDir\sample\APCMOS` directory

```
APCMOSI COMMUNITY(public) TARGET(101.102.103.104) CPC(cpc1) LPAR(lpar1)
ACTION(Activate) ACTIVATIONPROFILE()
```

Issue the APCMOSI command to deactivate LPAR named `lpar1`. For acceptable syntax for specifying the ACTION parameter, see the `apcmosi.txt` file in the `installDir\sample\APCMOS` directory.

```
APCMOSI COMMUNITY(public) TARGET(101.102.103.104) CPC(cpc1) LPAR(lpar1)
ACTION(Deactivate)
```

Hardware Access Facility (HAF)

The Hardware Access Facility (HAF) is a remote console application that uses the HMC console API provided by IBM to monitor the status of HMC/SE workstations and mainframe system objects or devices under their control.

Configuring HAF

To configure HAF

1. In Configuration Manager, navigate to Expert Interface, Automation, Events Interface, HAF Interface.
2. Check mark the Enable Monitoring box.
3. To the HMC list, add the community name and TCP/IP address for each HMC/SE workstation you wish HAF to monitor.
4. Specify the path and filename of the HMC API DLL file (`hwmwinca.dll`), copied to the local CA Automation Point server.
5. Click OK.
Your settings are saved.
6. Restart CA Automation Point to activate HAF.

Monitoring HMC Messages with HAF

HAF automatically polls the HMC/SE workstations that you have selected to detect whether a target HMC/SE can be reached and whether it will respond to a simple request for information. HAF reports when an HMC/SE is inaccessible or hung.

Using the same HMC console API, HAF is designed to be used in conjunction with the APCMOSI REXX program. Although HAF can only passively receive broadcasted messages from selected HMC/SE workstations, you must use the APCMOSI REXX program to issue commands to targeted HMC workstations.

When HAF and the APCMOSI REXX program operate under the control of the Automation Point Desktop, HMC automation can be written to correlate or interact with other defined sessions on the Automation Point Desktop. Messages received by HAF are displayed on the Automation Point Desktop in a session window named HAF_Events and automatically sent through CA Automation Point rules engine for processing. Therefore, you can write CA Automation Point rules to detect a HMC generated status message or event, and react by invoking APCMOSI to issue a predetermined HMC console command.

HAF Message Format

The messages sent from the HMCs through the API are in MIB format; that is, they consist mostly of sectioned numbers (for example, 1.2.3.4.5...) called object IDs or OIDs.

Wherever possible, these object IDs are translated into the English text that they represent (for example, the name of an LPAR or command).

Whether the HAF translates the object IDs or not, it does not reorder the pieces of information provided by the API.

Note: A multi-line message is an exception.

Because the HAF does not reorder pieces of information in messages, you can use the IBM API documentation to determine what data displays in the message and the order in which it displays. For example, if the HMC Access Facility receives a message containing 1.9243.7 (new status), 121.3.459.87.33874295.0 (the fictional LPAR ID for MVS001), and 11.259.44.783.91 (the event), it translates this to "Powered Off MVS001 Status Change." You might think that rearranging the elements of the message so that it says "MVS001 Status change: Powered Off" would make the message easier to understand, but such re-ordering of data chunks would make it impossible to use the manufacturer's documentation to determine what the format of each message should be.

HAF messages have the following format:

```
HAFImnny name type event keyword1 = value1 keyword2 = value2 ...hmc_name MMM NNN
```

mm

Specifies a number that indicates the type of message. To determine an *mm* value for a given event type, see the section [Determining HAF Message Numbers](#) (see page 348) later in this chapter. This number corresponds to the event portion of the message.

Note: Each time you change the code level of your HMC, look at the messages that it generates to ensure that the *mm* section of the message numbers has not changed for any of the messages that you are automating.

nn

Specifies a number that indicates the version of the message. Each time IBM changes the format of a particular message type, this number changes as well. This allows you to write an appropriate rule for the new data layout.

Note: Each time you change the code level of your HMC, look at the messages that it generates to ensure that the *nn* section of the message numbers has not changed for any of the messages that you are automating.

y

When a single HMC message is split up into multiple HAF messages, this single letter indicates which portion of the split HMC message this HAF message belongs to. (For more information about split messages, see the following descriptions of MMM and NNN.) Values are:

S (Single)

Indicates that HAF was able to present this HMC message as a single HAF message.

F (First)

Indicates that this HAF message is the first line of a split HMC message.

L (Last)

Indicates that this HAF message is the last line of a split HMC message.

M (Middle)

Indicates that this line is between the first and last lines of a split message.

name

Specifies the name of the HMC, SE, PEP, CPC, or LPAR that this message is about. In rare cases, the HAF cannot determine the name of the object from the object ID that displays in the raw message. If this is the case, the object IDs are shown.

type

Specifies the type of object (CPC_IMAGE_GROUP, CPC_Image, and so on) that this message is for.

event

Specifies the type of event that this message describes (STATUS_CHANGE, OBJECT_CREATED, and so on).

keywordN = valueN

Specifies a keyword/value pair that is part of a series that makes up the HMC message. The keyword portion is a MIB that describes what kind of data the value portion of the pair represents. The value portion contains the actual data that the pair is meant to convey. For example, two pairs could have a value portion of 2, but the first pair could have the MIB for RETURN_CODE as the keyword portion, and the second pair could have the MIB for MACHINE_STATUS as the keyword portion.

Each keyword/value pair in the HMC message is presented in the form *keyword = value*:

keyword

Specifies the ASCII text name for the type of object ID

value

Specifies the numeric value or the name of the specific instance of the type of object ID

Examples

- Suppose that in the IBM keyword/value pair, 1.3.6.1.4.1.2.6.42.0 and 1.3.6.1.4.1.2.6.42.0.3329843724 are the object ID for a CONSOLE_ID and the object ID for console MVS45C01, respectively. This object ID type/value pair would be displayed in the CA Automation Point message as CONSOLE_ID = MVS45C01.
- Suppose that in the IBM keyword/value pair, 1.3.6.1.4.1.2.6.42.0.20.0 and 1234567890 are the object ID for a CPC_SERIAL and the numeric value of the CPC serial number. This pair would be displayed as CPC_SERIAL = 1234567890.

Notes:

- Spaces are present on either side of the equal sign (=) in the *keywordN = valueN* pairs. This allows you to more easily use the word functions to parse the object names out of the message when you are using CA Automation Point rules.
- A hexadecimal value that has been translated to a name is also translated into a decimal number and included in parentheses after the name. For example, the condition code X'14' is presented as `COMMAND_CONDITION_CODE = REQUEST_RECV_ERROR (20)`, rather than `COMMAND_CONDITION_CODE = REQUEST_RECV_ERROR`.
- Within a single code level of a manufacturer's API for a given event type, the position and number of *keywordN = valueN* pairs will not change. However, although highly unlikely, the manufacturer may change the order of pairs or add or remove pairs when it changes the code level of the HMC. See the previous description of the *nn* portion of the message number for details.

hmc_name

The name of the HMC that generated this message. Use this along with MMM and NNN to relate the pieces of an HMC message to one another in the rules that you write.

MMM

A six-digit number, called the message number, that is incremented each time a message is received by HAF from an HMC. Each HMC has its own counter; that is, you will see message number 000001, message number 000002, and so on from each HMC. This value is useful in cases when HMC messages have been split into multiple HAF messages. You can use it to relate the pieces of a message to one another in the rules that you write. (See the NNN description for more information on split messages.)

NNN

A six digit number that starts with a value of 000001 for each new HMC message. This number is incremented for each new message that HAF creates when presenting the HMC message. As with MMM, you can use NNN to relate the pieces of an HMC message to one another in rules. HMC messages are split for one of two reasons; either the HMC message itself contains formatting characters that request that the HMC message be split into multiple HAF messages, or the HMC message contains more data than can be passed into rules in a single HAF message. Each HAF message in a multi-line HMC message has the same message ID (*HAFxmmmnn*) and the same message number (*MMM*), but *NNN* will be incremented by one for each HAF message that is generated from the HMC message.

Notes:

- *NNN* (with a value of 1) is placed on the end of the first message for a raw message even if the data from the raw message requires only one message; that is, the data will not be split into multiple messages.
- As described previously, the keywordN/valueN pairs of a message are only reordered in special circumstances. Some messages, such as hardware event messages, contain preformatted multi-line text as part of their data. When a message contains such known multi-line components, those keyword/value pairs are moved to the end of the data. This makes it easier to write rules against the message because *NNN* will vary for any piece of data that comes after any multi-line component of a message.
- ASCII text for a *keywordN* or *valueN* never splits across lines unless it contains formatting characters or is so long that a single value exceeds the maximum size of the rules buffer.
- In rare instances, a long message may split in a different place each time the message is passed through rules. If this happens, contact CA Technical Support.

As with z/OS or other operating system messages, you must study the format of the HAF messages that are received by CA Automation Point and write rules accordingly. The IBM HMC API manuals contain detailed information about the content of each of the HMC messages that HAF translates and presents.

Determining HAF Message Numbers

To determine a HAF message number (*mm*)

1. Look up the hexadecimal value of the event type in the document *IBM System z Application Programming Interfaces* manual (SB10-7030-11).
The number will have either no bits or a single bit on. Use the following table to determine the HAF message number.
2. Find the position of the bit that is on by numbering the bit positions from right to left with the rightmost position being 1. Consider no bits on to be position 0. Add one to the position to obtain the HAF message number.
3. Use the following table to find the *mm* value that corresponds to the hexadecimal event value.

Hexadecimal Event Value	<i>mm</i> Value in HAF <i>xmmnn</i>	Hexadecimal Event Value	<i>mm</i> Value in HAF <i>xmmnn</i>
0x00000000	01	0x00010000	18
0x00000001	02	0x00020000	19
0x00000002	03	0x00040000	20
0x00000004	04	0x00080000	21

Hexadecimal Event Value	<i>mm</i> Value in HAF <i>xmmnn</i>	Hexadecimal Event Value	<i>mm</i> Value in HAF <i>xmmnn</i>
0x00000008	05	0x00100000	22
0x00000010	06	0x00200000	23
0x00000020	07	0x00400000	24
0x00000040	08	0x00800000	25
0x00000080	09	0x01000000	26
0x00000100	10	0x02000000	27
0x00000200	11	0x04000000	28
0x00000400	12	0x08000000	29
0x00000800	13	0x10000000	30
0x00001000	14	0x20000000	31
0x00002000	15	0x40000000	32
0x00004000	16	0x80000000	33
0x00008000	17		

Samples for Determining HAF Message Numbers

As described earlier, the *mm* portion of message number HAF*mmnn* is associated with HMC event message types. The following samples show how to determine HAF message numbers for an IBM HMC.

This sample shows 14 different event types that are listed in IBM manual SC28-8141-09. The following names and hexadecimal values of the event types are listed in numeric order with the associated HAF message number included at the end in parenthesis:

- HWMCA_EVENT_COMMAND_RESPONSE 0x00000000 (01)
- HWMCA_EVENT_MESSAGE 0x00000001 (02)
- HWMCA_EVENT_STATUS_CHANGE 0x00000002 (03)
- HWMCA_EVENT_NAME_CHANGE 0x00000004 (04)
- HWMCA_EVENT_ACTIVATE_PROF_CHANGE 0x00000008 (05)
- HWMCA_EVENT_CREATED 0x00000010 (06)
- HWMCA_EVENT_DESTROYED 0x00000020 (07)
- HWMCA_EVENT_EXCEPTION_STATE 0x00000040 (08)
- HWMCA_EVENT_ENDED 0x00000080 (09)
- HWMCA_EVENT_HARDWARE_MESSAGE 0x00000100 (10)

- HWMCA_EVENT_OPSYS_MESSAGE 0x00000200 (11)
- HWMCA_EVENT_NO_REFRESH_MESSAGE 0x00000400 (12)
- HWMCA_EVENT_STARTED 0x00000800 (13)
- HWMCA_EVENT_HARDWARE_MESSAGE_DELETE 0x00001000 (14)

Other code levels for an IBM HMC could number these event types differently or could have more or fewer event types. In those cases, the HAF message numbers would change accordingly, but you would use the same process to determine what they should be.

Keep in mind that the *event* portion of a HAF message contains the event type for the message. For example, the third word of a HAFI05nn message for the previously mentioned level of IBM HMC would be ACTIVATE_PROF_CHANGE.

HAF Error Messages

You should always write a rule to detect and handle the following messages:

HAFI9999

This message informs you of problems with the format of the messages that your HMC is producing, as well as messages that are being lost because of performance parameters that are incorrectly set or because of program bugs.

HAFX9999

These messages report problems with HAF that are not related to any particular HMC.

Note: The X in HAFX9999 is the literal letter 'x.'

Chapter 13: Interacting with External Event Systems

This section contains the following topics:

[Deploy and Configure CA Automation Point Web Services](#) (see page 351)

[Using the CA NSM Interface](#) (see page 359)

[Using CA Automation Point to Monitor Windows Event Logs](#) (see page 388)

Deploy and Configure CA Automation Point Web Services

External applications can interact with CA Automation Point using Web Services. This section describes how to deploy and configure Web Service components. For an overview of Web Services capabilities, refer to *AP Web Services* section of *CA Automation Point Product Guide*.

Configure Web Services

Configure web service capabilities on your CA Automation Point server using Configuration Manager. Configuration Manager displays a web services dialog after you navigate to the Expert Interface, Infrastructure, Web Services entry in the configuration tree.

CA Automation Point web services are accessed through a Java servlet. Therefore, install a Java Runtime Environment (JRE) and a Java servlet application server on the AP server computer. The Configuration Manager dialog automatically detects when a JRE is installed and if Apache Tomcat is installed. If not, the dialog guides you through the process of installing those necessary components.

The web services provided by CA Automation Point are categorized into two areas of functionality *automation* and *notification*. You can enable just automation services, just notification services, neither, or both.

For automation web service requests to be processed, configure and run both the CA Automation Point Desktop application and the Remote Manager service.

Note: The CA Automation Point Desktop can be started manually from the Windows Start menu, or started automatically by the CA Automation Point Autostart Manager service.

For notification web service requests to be processed, configure and run the Notification Server service.

Deploy the RequestService Client

You are not required to run any CA software on your remote client computer. You can program HTTP requests directly from many different programming languages. However, when utilizing the CA Automation Point RequestService Java API or command-line application, deploy CA software on your remote client computer. The RequestService client can be used on Windows or Linux computers. For more information, see the *CA Automation Point Installation Guide* for supported operating systems.

Install a Java Runtime Environment (JRE) from <http://www.java.com> on your remote client computer before you run the RequestService client. JRE version requirements are described in Software Requirements section of *CA Automation Point Installation Guide*.

Follow these steps:

To deploy the RequestService client to your remote client computer:

1. Copy the following file from the CA Automation Point server to your remote client computer.

```
%AP_HOME%\Sample\websvc\RequestServiceClientDist.zip
```

2. Unzip the file into a directory of your choice.
3. To obtain documentation about the options for the Request Service command-line application, run the program
YourClientUnzipDirectory\RequestService\bin\RequestService.

Client documentation and samples are also included in the client distribution.

Define the RequestService Java API rootdir Property

When writing a client Java program that calls the RequestService Java API, the root directory of the RequestService client distribution must be specified to the Java environment. You supply this information using *the* `-D` flag to the java command when launching your own Java application. Through the `-D` flag, you specify a property named `com.ca.distauto.ap.websvc.rootdir`. The value for this property is the root directory of the RequestService directory tree that you unzipped onto your remote client computer. Specify a directory path syntax that is appropriate for your Windows or Linux operating system. For example, on Windows the command would be similar to this example:

```
java
-Dcom.ca.distauto.ap.websvc.rootdir="C:\YourClientUnzipDirectory\RequestService"
com.YourCompany.YourPackage.YourClass YourParameters
```

Both of the CA command-line scripts RequestService and RequestService.bat utilize this `-D` flag to point to the java API root directory. Both scripts are located in `YourClientUnzipDirectory\RequestService\bin`. The Java API uses the rootdir property to determine the directory into which it will write log files. The rootdir property is also used to find the RequestService.properties file, which controls configuration options for the Java API. This file controls security certificates and logging options. Directions for setting properties for security certificates are documented in [Configure a remote web service client for TLS](#) (see page 358). CA support staff can provide directions for changing logging properties when needed for diagnostic purposes.

Configure Web Service Security

All web service requests that CA Automation Point processes, are restricted to user IDs that are authorized to perform the desired operation. Do not create new user IDs for web services. Instead, utilize user IDs that are already in use by the underlying AP component. Operations against the web service sessions and intsessions objects are authorized by the user IDs configured in the following configuration item and authenticated by Remote Manager.

Configuration Manager

Expert Interface

Automation

Remote Viewing

Remote User Login Security

Similarly, operations against the web service *notifications* objects are authorized by the user IDs configured as Notification Manager login names. The particular privileges that a user must possess for a given operation are in “Web Service API” in the *CA Automation Point Command and Keyword Reference Guide*.

To secure the transmission of the user ID and password from the remote client to the CA Automation Point server, configure your communications to use TLS. TLS stands for Transport Layer Security, also known as SSL – Secure Sockets Layer. When sending HTTP requests to a TLS-secured web server, the HTTPS network scheme is required in your URI. This scheme secures your communications from the remote client to the Apache Tomcat web server. Both sides of this communication channel must be configured to use TLS.

Create TLS certificates

To configure TLS communications, create TLS certificates and create data stores to hold those certificates. A server certificate represents the CA Automation Point/Tomcat server. Depending on your security needs you can also choose to create a client certificate.

The Apache Tomcat server and the optional CA RequestService client both understand Java keystore files, therefore this document only describes the use of Java keystore files and certificates. When not using the RequestService client, Tomcat PKCS11 or PKCS12 format keystores can be configured. These two additional options are beyond the scope of this manual.

A server in a TLS conversation (in this case Tomcat) has a keystore in which it stores the certificates that authenticate the server. The server sends its certificate to a client during TLS negotiation. A client has a truststore, in which it stores certificates that it is willing to trust. The client confirms the certificate that is sent by the server during TLS negotiations against the client's truststore.

Certificate Authorities, like VeriSign, Entrust, and DigiCert can be used to sign a given server's certificate. With a signed certificate, a client does not need a certificate from every server in the world stored in its truststore. If the client trusts the Certificate Authority, it knows that it can trust a signed certificate that it receives from an otherwise unknown server. Public web server applications often use this type of approach, because of the sheer volume of clients and servers. Obtaining a certificate that is signed by a Certificate Authority is beyond the scope of this document.

A CA Automation Point server is typically deployed into a more confined corporate intranet. The time and cost of obtaining a certificate that is signed by a Certificate Authority may not be warranted for this more restricted deployment. Create a self-signed certificate for your server, place it into the server's keystore and into the client's truststore. This procedure configures a secure encrypted communication channel. Such a deployment remains practical as long as the number of AP clients and servers is no more than a few dozen computers.

When using the CA RequestService client, you can configure RequestService to ignore server certificates. By so doing, the client loses its ability to authenticate the server, however all data is securely encrypted. When you are confident that no rogue server exists behind your corporate firewall that can impersonate an AP server for destructive purposes, this configuration has the advantage that you only configure a server certificate into Tomcat's keystore. You do not need to deploy any server certificates into any client truststores. This is the minimal configuration that can be used to create a TLS connection between a remote client and the CA Automation Point/Tomcat server.

Each client web service request contains a user ID and password and can be sufficient client authentication for your site. However, you can also optionally configure the additional TLS security of client-side certificates. Client certificates are signed by a Certificate Authority or the client certificates are self-signed just like server certificates. In this scenario, set up Tomcat's TLS configuration to demand a client certificate. Create a client certificate and store it into the client's keystore and into the Tomcat truststore. This technique is essentially the reverse of the procedure that is used to deploy a server certificate. Do this procedure for every remote client in your environment. During TLS negotiation, Tomcat asks the client for its certificate. The client retrieves its certificate from its keystore and sends it to Tomcat. Tomcat verifies the certificate against the Tomcat truststore. Through this procedure, Tomcat verifies that the client application can be trusted.

Note: CA Technologies web service will still perform its user ID and password verification on the request.

You can create keystores and truststores (both of which are formatted as Java keystore files) with a program named `keytool` from your Java Runtime environment. That program can be found in `YourJavaInstallationDirectory\bin\keytool`. `Keytool` can also create a self-signed certificate, place a certificate into a keystore, import a Certificate Authority signed certificate, export a certificate from a keystore, and import a certificate into a truststore. Using these capabilities of `keytool`, you can create the various scenarios described in this section. Read the documentation for `keytool` to understand the details that are required to accomplish those steps.

Configure Tomcat for TLS

Apache Tomcat is a third-party product. The most accurate source of documentation is from Apache itself. The definitive reference for configuring SSL for Tomcat is the “Apache Tomcat SSL Configuration HOW-TO”. You can use the “Apache Tomcat HTTP Connector Reference” to look up the definitions of the configuration properties utilized in the HOW-TO document. Other web tutorials exist which describe the configuration in a step-by-step fashion. Find and follow documentation for the version of Tomcat that is used by the release of CA Automation Point that is installed.

In general, the TLS configuration tasks that you must accomplish for Tomcat are:

1. Use the `keytool` program to create the keystores, truststores, and certificates to achieve your desired security configuration.
2. Within Tomcat’s `server.xml` configuration file, modify the `Connector` element which has `port="8443"`. This port is the TLS connector. Specify a keystore file, the keystore type, and the keystore password. When using client certificate authentication, enable that option and specify the truststore file, type, and password.
3. You now have access to your URIs with both TLS and with an unencrypted connection. Test your TLS connection by specifying URIs with the HTTPS network scheme and the 8443 port. This test assumes that you retained the default TLS port number of 8443. Specify your URIs in the following format:
`https://localhost:8443/apwebsvc/YourDesiredApResource`.
4. Once you have successfully tested your TLS connection, you can disable (comment out) the unencrypted `Connector` element, which has port 8080. This procedure prevents Tomcat from serving any requests over an unencrypted connection.

Note: This procedure affects every application that is hosted under this particular Tomcat server.

Configure a remote web service client for TLS

Once your server requires a TLS connection, configure your remote clients to use TLS connections. If you are not using the CA RequestService client, follow the TLS techniques for your chosen programming language. This document describes how to configure TLS for the RequestService client.

Since you specify the URI to RequestService, you control the use of TLS, because you specify *https* and the 8443 port number on your URI. To control the use of a keystore and/or truststore you must modify a Java properties file that is named *RequestService.properties*.

To modify that file in a client distribution, first copy the file from the directory:

```
YourClientUnzipDirectory\RequestService\distrib
```

To the directory:

```
YourClientUnzipDirectory\RequestService\Site\Config
```

If you are using the RequestService client on the AP server, copy the file from this directory:

```
%AP_HOME%\distrib
```

To the directory:

```
%AP_DATA%\Site\Config
```

Within the RequestService.properties file, you control the use of keystores and truststores by using the following property names:

```
com.ca.distauto.ap.websvc.client.keyStoreFile  
com.ca.distauto.ap.websvc.client.keyStorePass  
com.ca.distauto.ap.websvc.client.trustStoreFile  
com.ca.distauto.ap.websvc.client.trustStorePass
```

With these properties, you can:

1. Validate server certificates against a client truststore that you have created.
2. Respond to a server request for a client certificate. A certificate is retrieved from a client keystore that you created, and the certificate is returned to the server.

You can specify that one of these two capabilities, both of them, or neither of them are enabled. A property is enabled if it is uncommented within the file and the property is assigned a value (Property=SomeValue). None of these properties are set by default. These default settings lead to the following behavior.

1. Server certificates are accepted but are not validated. No client truststore is needed.

2. The client does not return a client certificate to a server. No client keystore is needed.

Comments within the `RequestService.properties` file describe the detailed behavior of each property and the meaning of the value to which the property can be set.

Using the CA NSM Interface

CA Automation Point can interface with CA NSM in the following ways:

- Through the CA Automation Point ADDRESS TNG environment, objects that represent the systems and resources that are monitored by CA Automation Point can be stored in the CA NSM Repository and displayed on the CA NSM WorldView Map.
- Through the CA Automation Point Event Traffic Controller component and ADDRESS TNG environment, messages, events, commands, and requests for actions can be communicated between CA Automation Point and the CA NSM Event Manager.
- Through the CA Automation Point Remote Viewer application, sessions that are monitored by CA Automation Point can be viewed from a remote Windows workstation. The Remote Viewer can be launched from an object that represents a system that is managed by CA Automation Point on the WorldView Map. (This topic is discussed in detail in the chapter "Viewing Remote Sessions.")

Understanding the ADDRESS TNG Environment

This section contains information that will help you to better understand the ADDRESS TNG environment. There are separate sections for the WorldView Map and the Event Traffic Controller, as the information that pertains to each differs.

Using the WorldView Map

The ADDRESS TNG environment provides a REXX-based programming interface to the objects contained in the WorldView Repository. With this interface, you can

- Create new objects
- Read and write values into properties of existing objects
- List all objects that belong to a specific class
- Delete objects from the WorldView Repository

Using the ADDRESS TNG environment, you can graphically depict the activity that CA Automation Point is conducting on the WorldView Map.

For a simple example, you can create an object to represent each system that is being monitored by CA Automation Point and, using a combination of rules and REXX, turn an icon different colors based on various changes in the state of the managed system. For example, a communication failure could turn the icon red.

You can also use the WorldView Map interface to launch the CA Automation Point Remote Viewer from the menu associated with a session object, obtaining direct access to consoles monitored by CA Automation Point.

Before you can use the ADDRESS TNG environment to interface with the WorldView Map, you must have installed CA NSM in your computing environment. This may be the same workstation on which you are running CA Automation Point. The workstations running CA Automation Point only require the following:

- Network connectivity to reach the server holding the WorldView Repository
- Client-side database software to support the native database
- The CA NSM WorldView Map Interface component.

Currently, the CA-AP NSM Gateway service only supports concurrent access to one repository. You can change the repository name through the CA NSM WorldView Repository dialog of Configuration Manager. All CA Automation Point access to the WorldView Map will be to the new repository that you specify. The CA-AP NSM Gateway service runs as a Windows service.

The file `\SAMPLE\APTNG\aptng.txt` in the CA Automation Point directory contains information about importing class definitions for objects managed by CA Automation Point, for example:

- Sessions in the WorldView Repository
- A list of provided sample programs that illustrate the use of interfacing ADDRESS TNG with the WorldView Map.

Note: For more information about linking session objects to a Business Process View under the WorldView Map, see the CA NSM documentation.

Using the Event Traffic Controller

The ADDRESS TNG environment enables REXX programs to send events to the CA NSM Event Manager. The interface allows you to:

- Send a message to CA NSM Event Manager (and, optionally, wait for an operator's reply)
- Send a command to be executed by CA NSM Event Manager
- Send an SNMP trap to any SNMP-aware system manager (for example, CA NSM Event Manager).

For example, a REXX application that is performing an automation task in response to a z/OS event could issue a message to the CA NSM operator console, wait for the operator's reply, and then use the reply to proceed down one of several possible paths to a solution for the problem.

You could also use SNMP traps to communicate to any SNMP-knowledgeable system manager. Through this technique, you could convey events gathered from a legacy system (through a direct-connect data feed between the legacy system and CA Automation Point) to an SNMP manager that would otherwise have no mechanism to connect to the legacy system.

Before you can use the ADDRESS TNG environment to communicate with CA NSM Event Manager, you must have installed the CA NSM Event Manager Interface component on the machine that is running CA Automation Point.

The CA-AP NSM Gateway Service

For the ADDRESS TNG environment to operate between different machines, you must establish network connectivity to the machines that hold the CA NSM Repository or run Event Manager.

When you are ready to access the ADDRESS TNG environment from CA Automation Point, you must have an operating CA-AP NSM Gateway service on the machines running CA Automation Point.

To install the CA-AP NSM Gateway service

1. Start Configuration Manager.
2. Select Expert Interface, Automation, Events Interface, CA NSM WorldView Repository.
3. Place a check mark in the box Enable the ADDRESS TNG Environment.
4. Enter the information requested for the WorldView Repository, User Name, and Password fields.
5. Click OK.

TNG Command Summary

After the CA-AP NSM Gateway service is running, you can access the WorldView Repository from the ADDRESS TNG REXX environment. This environment supports these major commands (see the *Command and Keyword Reference Guide* for command syntax):

CREATE

Creates a new object

DELETE

Deletes a specified object

GET

Reads one or more PROPERTY/VALUE pairs from a specified object

LIST

Lists all objects from a specified class

SET

Writes one or more PROPERTY/VALUE pairs or the contents of a stem variable construct to a specified object

SNMPTRAP

Sends an SNMP trap to the specified host

UNICMD

Tells the CA NSM Event Manager component, which resides on the specified host, to execute the supplied command

UNIWTO

Sends the supplied message to the CA NSM Event Manager component on the specified host

UNIWTOR

Sends the supplied message to the CA NSM Event Manager component on the specified host and receives a reply

VER

Returns information about the version of the ADDRESS TNG environment that is running

All of these commands are accessed using the standard CA Automation Point style for using REXX command environments. For example:

```
ADDRESS addressenvironment 'majorcommand commandmodifiers'
```

By using these commands and checking return codes, you can use REXX to create visual effects of automation on the WorldView Map. For example, the following command creates a new object in the WorldView Repository:

```
ADDRESS TNG 'CREATE OBJECT(Hub.MyHub)'
```

In this example, the object is a wiring hub. The object class type is Hub and object is MyHub. The ADDRESS TNG commands that interface with the WorldView component require the OBJECT parameter. The argument to the OBJECT parameter consists of two parts: the name of the class to which the target object belongs (defined in the CA NSM environment) and the name of the object itself. These two parts *must* be separated by a period.

By creating the object MyHub in the WorldView Repository, you automatically make it available to the CA NSM graphical interfaces. You can then set the value of the properties of MyHub (for example, the status of port 1) by entering the following command:

```
ADDRESS TNG 'SET OBJECT(Hub.MyHub)
PROPERTY("port1_status") VALUE("In Service")'
```

Similarly, you can query the object by entering this command:

```
ADDRESS TNG 'GET OBJECT(Hub.MyHub)
PROPERTY("port1_status")'
```

Using the CA NSM command processors in REXX programs triggered by either CA Automation Point rules or manual procedures, you can effectively maintain a graphic status of the devices and processes that are managed by CA Automation Point on the WorldView Map.

See the *Command and Keyword Reference Guide* for information on specific ADDRESS TNG commands and ADDRESS TNG return information.

Configuring CA Automation Point CA NSM Objects Using Configuration Manager

In addition to using REXX scripting to manage CA NSM objects, you can also use Configuration Manager to create and delete objects representing CA Automation Point sessions. The CA NSM WorldView Repository dialog in Configuration Manager provides two options for configuring CA Automation Point CA NSM objects.

- **Express configuration.** Use this option to specify that all CA Automation Point sessions defined in the active session definition set are to be represented. You automatically create a CA Automation Point server object, and within this object, a CA NSM object for each CA Automation Point session defined in the active session definition set.
- **Advanced Configuration.** Use this option to add or delete CA Automation Point CA NSM objects individually. The CA Automation Point server object is queried using the active session definition set to determine which sessions have CA NSM representation. An Advanced dialog shows a list of CA Automation Point sessions and designates which ones are represented. You can add or delete CA NSM objects for the sessions.

After you have used Configuration Manager to configure CA Automation Point CA NSM objects, the CA NSM WorldView map reflects your actions by accessing an object within Managed Objects entitled APServerGroup. An object named after the CA Automation Point server displays within this group. When you drill down into this server, you will see representations of the CA Automation Point sessions you have configured.

To configure CA Automation Point CA NSM objects using Configuration Manager

1. In Configuration Manager, choose Expert Interface, Automation, Events Interface, CA NSM WorldView Repository.

The CA NSM WorldView Repository dialog displays.

2. In the Configure AP CA NSM Objects area of the CA NSM WorldView Repository dialog, click Express to perform express configuration or Advanced to perform advanced configuration. For details on these options, refer to the dialog help.
3. If you select Express configuration, and an AP server object already exists, you are prompted to delete this object to ensure that only active session definitions are represented. Follow this instruction. At the end of the configuration, a summary dialog displays telling you the number of objects created.

If you select Advanced configuration, the Configure AP CA NSM Objects – Advanced dialog displays. Use the Create and Delete buttons on this dialog to add or delete CA NSM objects. Use the dialog help to assist you. Close the dialog when you are finished.

4. Click OK to save your changes.

Understanding the Event Traffic Controller

Through Configuration Manager, you can configure the Event Traffic Controller. The Event Traffic Controller provides an interface between CA Automation Point and the CA NSM Event Manager, allowing you to control the flow of event traffic between the two products.

Here is a list of some functions you can perform with the Event Traffic Controller:

- Specify the CA NSM or Windows hosts you want to process
- Enable or disable a given CA NSM or Windows event type
- Turn logging on or off for a given event type
- Forward all CA Automation Point messages to CA NSM by default
- Control performance metrics, including the frequency of event polling
- Modify session and system names

For useful information about how you can use the Event Traffic Controller, see the section [Troubleshooting the Event Traffic Controller](#) (see page 380) in this chapter.

Communication Between Event Traffic Controller and Event Manager

The Event Traffic Controller provides an interface between CA Automation Point and the CA NSM Event Manager component, allowing you to control the flow of event traffic between the two products.

Sending Events from CA Automation Point to Event Manager

The Event Traffic Controller provides the capability to send events from CA Automation Point to CA NSM Event Manager, including the following:

- Forwarding messages from CA Automation Point to CA NSM Event Manager
- Sending arbitrary messages with the UNIWTO command
- Sending SNMP traps from CA Automation Point
- Enabling CA Automation Point to execute remote commands

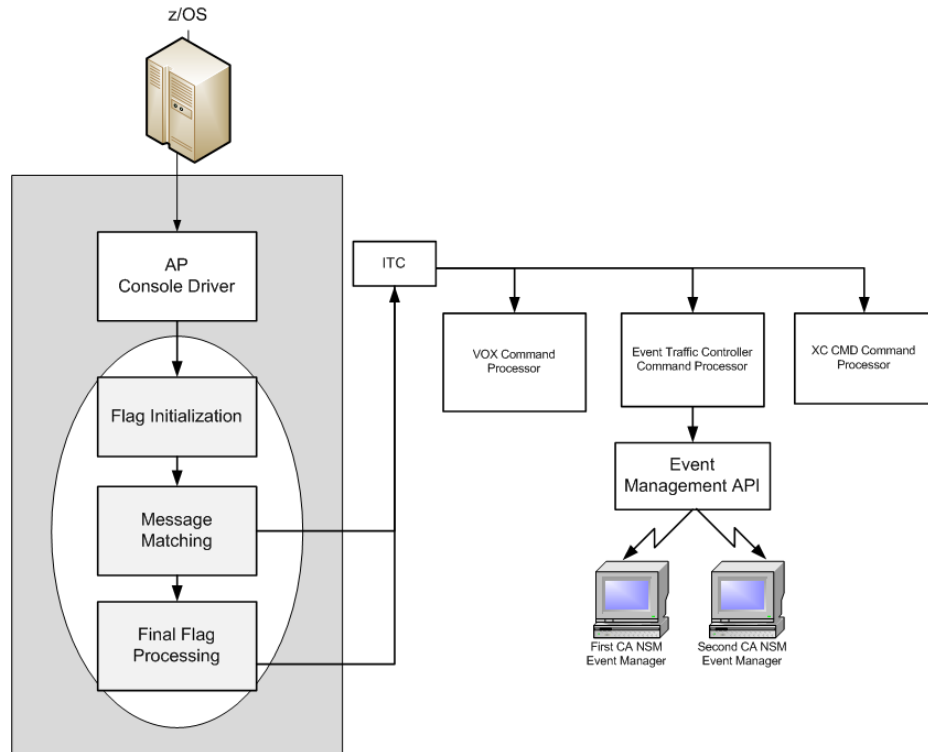
Forwarding Messages

The CA NSM Event API library is used to forward messages from CA Automation Point to CA NSM Event Manager. The messages are forwarded to all specified CA NSM hosts, and their message text comes directly from CA Automation Point.

A rules flag, UNIFWD/NOUNIFWD, enables you to easily send all messages or no messages to CA NSM Event Manager without adding message matching to the CA Automation Point rule set. The UNIFWD flag is initialized before rules processing begins for an event. This flag can be reset numerous times as various rules are matched. At the completion of rules processing, the value of the flag is used to take an appropriate action.

When the UNIFWD flag or other Event Traffic Controller commands are processed, the CA Automation Point Inter Task Communication (ITC) tools queue the message item to the Event Traffic Controller command processor, ensuring that the execution of Event Traffic Controller commands and the continued processing of rules occur asynchronously. Three types of items are then read from the CA NSM Event Manager queue: messages, commands, and SNMP traps.

The following diagram illustrates this flow of events:



The CA Automation Point Configuration Manager utility allows you to specify the CA NSM hosts to which you want to send events from CA Automation Point to CA NSM Event Manager. The UNIFWD flag forwards each selected message to every host you specified.

For detailed information about UNIFWD, see the *Command and Keyword Reference Guide*.

When thinking about message forwarding, you should consider the type and volume of messages you want forwarded to the Event Management console. For example, you may want to forward only those messages that pertain to the items that have made their way to CA Automation Point for which you have not written automation or those messages that notify you that a particular problem has been corrected.

One possible reason to forward CA Automation Point messages to CA NSM Event Manager is to use the CA NSM Event Manager interface to the CA Service Desk for opening and managing issues.

Note: For information about the CA Service Desk interface with CA NSM Event Manager, see the CA Service Desk documentation.

To forward CA Automation Point messages to CA NSM Event Manager

1. From the Configuration Manager main window, choose Expert Interface, Automation, Events Interface, CA NSM Event Manager.
2. Select the Forward Messages to CA NSM event.
3. Add the CA NSM host to which you want to forward messages.
4. Select the host by using the Include button to move it from the Available Hosts list to the Selected Hosts list.
5. Save your changes.
6. Add the UNIFWD keyword to the MSGID rule of each message you want to forward to the selected CA NSM host.
7. Restart CA Automation Point.

Sending Arbitrary Messages

The UNIWTO rules keyword enables you to send messages to only those CA NSM hosts that you specify with the message text that you specify.

Sending SNMP Traps

The process of sending an SNMP trap from CA Automation Point is similar to the process of sending a message. However, the SNMPTRAP rules keyword creates an SNMP item on the Event Traffic Controller queue. The Event Traffic Controller command processor then sends an SNMP trap using the CA NSM Event API library. The SNMPTRAP command supports all the operands that are accepted as Open System standards for an SNMP trap command. The trap can be received by any SNMP-aware network manager (for example, CA NSM Event Manager).

Executing Remote Commands

The process of sending remote commands from CA Automation Point to CA NSM Event Manager is similar to the process of sending an SNMP trap. The UNICMD rules keyword creates a command item on the Event Traffic Controller queue. The Event Traffic Controller command processor then sends the command to CA NSM Event Manager for execution using the CA NSM Event API library. Because CA NSM Event Manager can issue any native operating system command, this feature enables CA Automation Point to issue commands to a remote operating system that has CA NSM Event Manager running, without the need to maintain a full-time session to that operating system.

Information on Events Sent from CA Automation Point to CA NSM

When an event is sent from CA Automation Point to CA NSM, information about the event is available in the following CA NSM fields:

Message

Displays the message text associated with the event that was captured by CA Automation Point.

Node

Displays the name of the CA Automation Point host that sent the event to CA NSM.

Station

Displays the name of the CA Automation Point session in which the event occurred. You can override this value using the ORIGHOST operand on the UNIWTO or UNIWTOR command.

Facility

Displays a value of *SNMP* for SNMP traps; the value is *Automation Point* for all other events.

Category

Displays the CA Automation Point component (UNICMD, UNIFWD, UNIWTO, or UNIWTOR) that sent the event.

Severity

Displays a value of E (error) when the event is a CA Automation Point action message; the value is I (information) for all other events. You can override this value using the SEVERITY operand on the UNIWTO or UNIWTOR command.

Time

Displays the time at which the event was generated.

Receiving Events From Event Manager

The CA NSM Event API library is the central component involved in enabling CA Automation Point to receive messages from CA NSM Event Manager. The API library function enables a program to read the CA NSM Event Manager log, and it can be used whether the CA NSM Event Manager resides on the same machine or a different machine on the network. Each entry in the CA NSM Event Manager log is a potential message that can be used in CA Automation Point.

Types of Monitored Events

The entries that are to be read from the CA NSM Event Manager log can be filtered so that only certain messages are retrieved. CA Automation Point monitors the CA NSM Event Manager log for the following types of events. CA Automation Point monitors each type of event in a separate, automatically defined special session.

The following are names of the sessions for each event type:

- **Any entry that matches user-supplied filtering parameters**—These entries are read and passed to CA Automation Point rules for processing within the UNI_MSGS session. When no filtering is specified, all CA NSM Event Manager messages are forwarded to CA Automation Point rules for processing.
- **Notification entries**—Entries that begin with the NMFIND keyword are read and passed to CA Automation Point rules for processing within the NM_REQUESTS session. This session is also used for NMFIND requests that originate from CA OPS/MVS.
- **SNMP traps**—SNMP traps that CA NSM Event Manager captures and places in its log are read and passed to CA Automation Point rules for processing within the SNMP_TRAPS session.
- **Host (session) commands**—Entries that begin with the SESSCMD keyword are read and passed to the HOST_COMMANDS session.
- **Windows event log messages**—Messages from a Windows system's event log are read and passed to CA Automation Point rules for processing within the WIN_EVENTS session. This event type is read directly from the Windows operating system.

Types of Sessions

During the configuration of the Event Traffic Controller, rules are automatically generated for the sessions that are associated with the selected event types. The session definitions for the special Event Traffic Controller sessions are automatically defined in the session definition set. Each of the message streams described in the previous section are funneled into one of the following sessions:

- **UNI_MSGS**—User-filtered messages are sent to this session. The only generated rule associated with this session logs the message. You can add rules that match messages in this session, enabling CA Automation Point to take action upon any event that is captured by CA NSM Event Manager.
- **NM_REQUESTS**—Notification messages are sent to this session. The generated rules associated with this session log the message and launch an appropriate NMFIND command based on the fields that are contained in the message. In this manner, CA NSM Event Manager can indirectly issue pages, voice, and e-mail notifications. You can use this method to issue notifications on behalf of other CA NSM options that have the ability to write messages to CA NSM Event Manager, such as CA Service Desk.

Note: For information about the CA Service Desk interface with CA NSM Event Manager, see the CA Service Desk documentation.

This session is also used for NMFIND requests that originate from CA OPS/MVS.

- **SNMP_TRAPS**—SNMP messages are sent to this session. The only generated rule associated with this session logs the message. You can add rules that match the messages in this session, enabling CA Automation Point to take action upon any SNMP event that is captured by CA NSM Event Manager.
- **HOST_COMMANDS**—SESSCMD messages are sent to this session. The generated rules associated with this session log the message and execute the supplied command in a specified CA Automation Point session. In this manner, CA NSM Event Manager can indirectly issue z/OS commands.
- **WIN_EVENTS**—Messages from a Windows event log are sent to this session. The only generated rule associated with this session logs the message. You can add rules that match the messages in this session, enabling CA Automation Point to take action upon any Windows event.

Note: These session names are the default names, and should normally *not* be changed.

Obtaining Monitored Event Information

When an event is extracted from CA NSM, information about the event is available in the following CA Automation Point variables:

&HOSTTIME

Specifies the time at which the event was generated.

&JOBID

Specifies the TCP/IP address of the device that sent the SNMP trap.

&JOBNAME

Specifies the value from the CA NSM Process field; this field normally contains the value *ProcessID,ExecutableName*.

&MONNAME

Specifies the name of the host that generated the event.

&MONNUM

Specifies the name of the CA NSM Event Manager host from whose log the event was extracted.

&MONPRTY

Specifies Error, Warning, Info, Success, or Failure.

&MONTYPE

Twenty bytes from the CA NSM Facility field are combined with twenty bytes from the Category field, resulting in a value of *Facility,Category*.

&MSG, &WORD1, &WORD2, ...

Specifies the entire message text associated with the event and the individual words of the message.

To monitor and process CA NSM Event Manager messages with CA Automation Point

1. From the Configuration Manager main window, choose Expert Interface, Automation, Events Interface, CA NSM Event Manager Events.
The CA NSM Event Manager Events dialog displays.
2. Select the Monitor arbitrary messages from CA NSM event from the Type of Event list.
3. Check the Enable Monitoring box.

4. Add the CA NSM host that you want CA Automation Point to monitor and receive messages from.

Select the host by using the Include button to move it from the Available Hosts list to the Selected Hosts list.

Note: You can restrict the CA NSM messages that will be received and processed by highlighting the selected host and clicking the Filtering Criteria button.

5. Click the Host Access Security button, and then specify a valid login user ID and password for the remote CA NSM host.
6. Save your changes.
7. Add a CA Automation Point rule similar to the following rule to take the desired action:

```
MSGID(text that matches a CA NSM message)
COLOR(RED)
```

8. Restart CA Automation Point.

Monitoring Multiple Copies of Event Manager

Because multiple copies of Event Manager can run throughout an enterprise, CA Automation Point enables you to specify multiple Event Manager host names for it to monitor, enabling CA Automation Point to receive messages from each occurrence of Event Manager that is running.

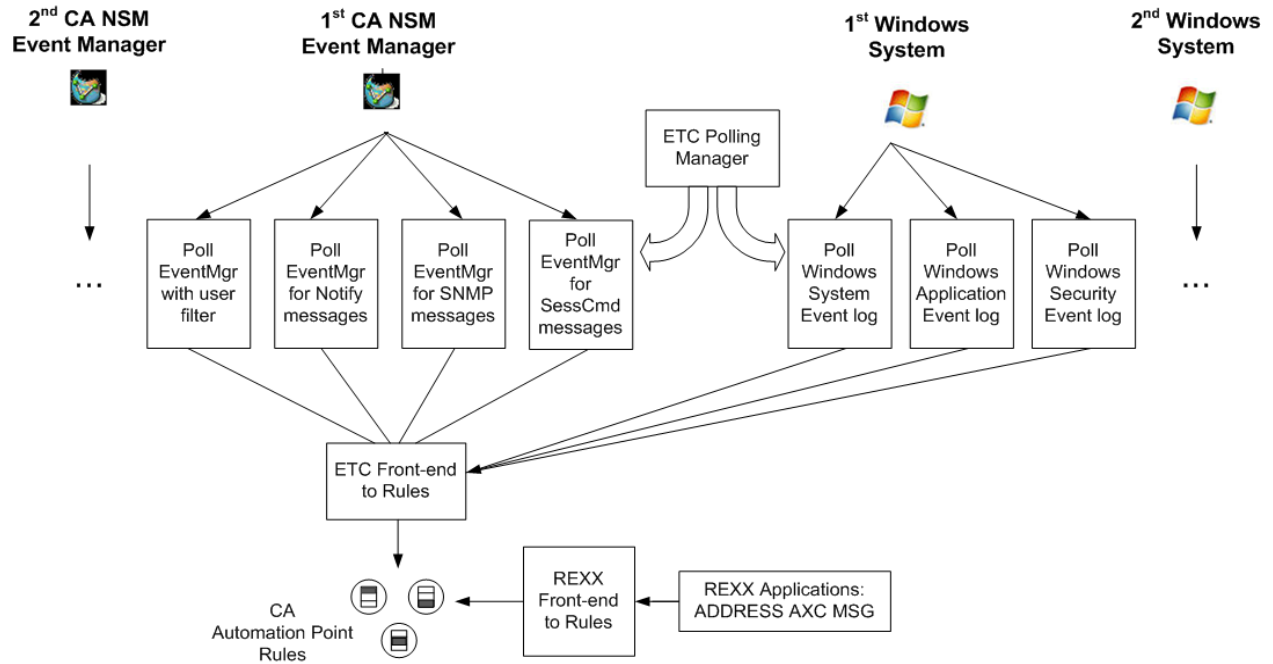
Event Polling

The Event Traffic Controller polling manager is automatically started during CA Automation Point startup if you have configured any events to be monitored. The polling manager controls the polling of external sources of events. The polling manager launches up to four Event Manager polling processes for each Event Manager host that is to be monitored and one Windows event log polling process for each Windows event log to be monitored.

Each Event Manager polling process detects whether Event Manager can be reached and whether CA Automation Point is running. When both components can be reached, an Event Manager polling process connects to the Event Manager log and reads a message stream. Each Windows event log polling process operates in a similar fashion.

Each polling process submits messages to the rules engine for processing. This action is similar to performing an ADDRESS AXC MSG operation from a REXX program.

The following diagram illustrates this process:

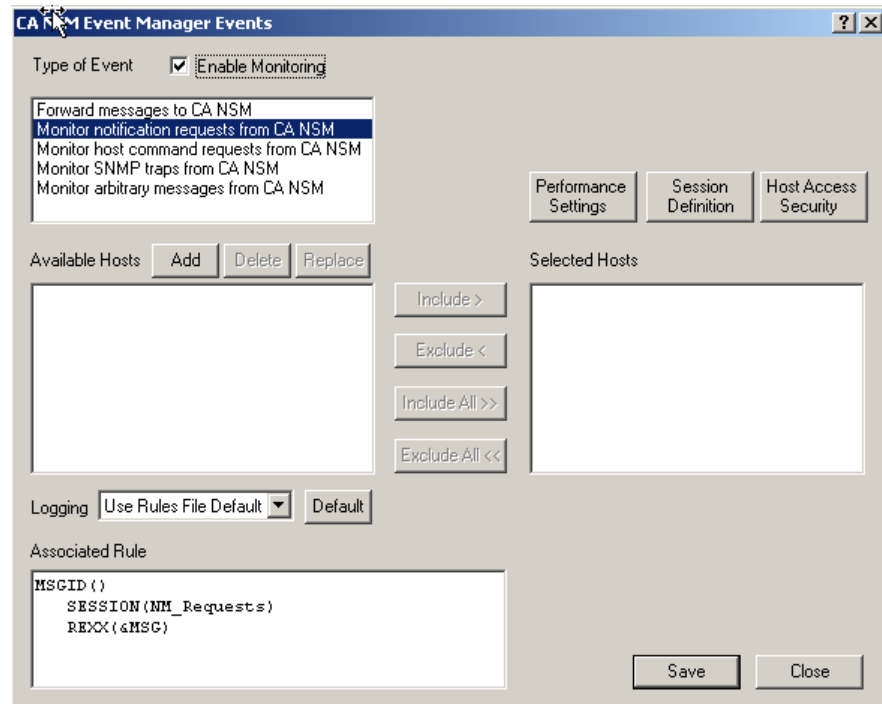


Minimizing the Rules and REXX You Need to Write

The Event Traffic Controller lets you send messages between CA Automation Point and Event Manager without writing any REXX code. However, to match the messages that are being sent from Event Manager, sessions need to be defined and CA Automation Point rules need to be written. This is accomplished through Configuration Manager. Using the data you supply, Configuration Manager generates the appropriate sessions and rules.

Configuring the Event Traffic Controller

To configure the Event Traffic Controller, select Expert Interface, Automation, Events Interface, CA NSM Event Manager Events from the Configuration Manager main window. The CA NSM Event Manager Events dialog displays:



This dialog allows you to configure the event types listed in the Type of Event field. The options displayed on this dialog depend on the type of event you select.

For detailed information for each field, see the dialog Help.

Important! *To ensure that the changes you make to the configuration of the Event Traffic Controller take effect, you must restart CA Automation Point.*

Configuring Security for the Event Traffic Controller

You can use the following remote viewing security measure with the Event Traffic Controller. If a list of trusted host names is specified, the Event Traffic Controller restricts SESSCMD requests to only those user names from the hosts specified in this file.

You must stop and restart CA Automation Point for Event Traffic Controller security to reflect any security-related changes that have been made to CA AP Remote Manager options.

See the chapter "[Viewing Remote Sessions](#) (see page 109)" for details about setting up security for remote viewing.

Accessing a Remote Host

Events that monitor remote hosts display a Host Access Security button. This button launches the Remote Host Access Security dialog, which allows you to specify a security context for the Windows and CA NSM hosts that are defined to the Event Traffic Controller.

Both the Event Manager log and the Windows event logs are treated as remote resources and are protected by Windows security; therefore, each user must have a security context to access these remote resources.

Note: Even if you have not entered a security context in the Remote Host Access Security dialog, one may already be set up for a variety of reasons. One example is that you are running CA Automation Point under a certain user ID and password and the same user ID and password also exist on a remote machine.

User Rights

There are certain user rights that any remote login that is used by the Event Traffic Controller must have.

Any user name entered on the Remote Host Access Security dialog must have the Access this Computer from Network user right. Typically, this user right is assigned by default when a new user is created. This right can also be assigned by accessing the User Manager, and then choosing Policies, User Rights, Right, Access this computer from network.

If the Event Traffic Controller is configured to monitor the Windows Security Event Log on a remote system, the user name that is used to log in to that remote system must have the Manage Auditing and Security Log user right. This right can be assigned by accessing the User Manager, and then choosing Policies, User Rights, Right, Manage auditing and security log.

Enabling REXX Applications to Send and Receive Event Manager Messages

The SNMPTRAP, UNICMD, UNIWTO, and UNIWTOR commands enable CA Automation Point to perform the following actions:

- Send messages to Event Manager
- Send messages to Event Manager and wait for an operator's reply
- Send SNMP traps
- Issue remote commands

These commands are available within a REXX script through the ADDRESS TNG environment.

The following example sends a message to CA NSM:

```
ADDRESS TNG 'UNIWTO HOST(unihost) MESSAGE(HELLO)'
```

Enabling CA NSM Event Manager to Generate Voice Notifications

The process of enabling Event Manager to generate voice notifications involves the following actions:

- Users define the Event Manager actions that issue a notification request. This is accomplished using standard Event Manager message records and actions.
- CA Automation Point monitors Event Manager's log for notification requests and launches an appropriate NMFIND command

The goal is for the Event Manager administrator to place a CA Automation Point NMFIND command into the Event Manager log. This is accomplished by matching a message and using the SENDOPER action and entering a complete NMFIND command in the Text field.

To send notification requests (NMFIND commands) from Event Manager to CA Automation Point

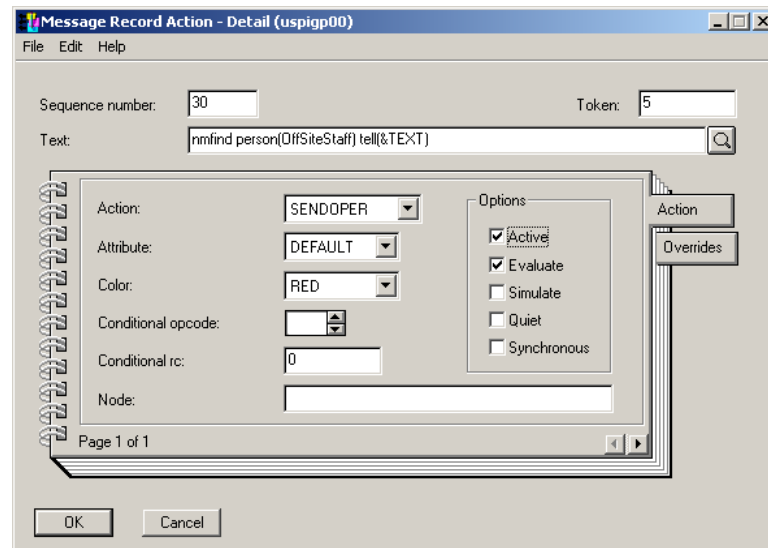
1. From the Configuration Manager main window, select Expert Interface, Automation, Events Interface, CA NSM Event Manager.
2. Select the Monitor notification requests from CA NSM event.
3. Check the Enable Monitoring box.
4. Add the CA NSM host for which you want CA Automation Point to monitor notification requests.
5. Select the host by using the Include button to move it from the Available Hosts list to the Selected Hosts list.
6. Click the Host Access Security button, and then specify a valid login user ID and password for the remote CA NSM host.

7. Save your changes.
8. Restart CA Automation Point.
9. On the CA NSM host, define an Event Manager message action to issue the notification request (NMFIND command) to CA Automation Point.

For details about the NMFIND command, see the *Command and Keyword Reference Guide*.

For information about defining a message action, see the CA NSM documentation.

The following illustration shows how to generate a notification using the CA NSM Message Record Action - Detail window (pay particular attention to the Text and Action fields):



The following example shows how to generate a notification by using an Event Manager rule:

Action:

SENDOPER

Text:

```
nmfind PERSON(BILL SMITH) TELL (IMPORTANT MESSAGE)
ASK (ARE YOU COMING TO THE DESIGN MEETING,
YES I AM ON MY WAY::YES.REX,
NO I HAVE ANOTHER OBLIGATION::NO.REX)
```

Notes:

- As in the preceding example, enter the NMFIND keyword in lowercase. This enables the request to be recognized by the Event Traffic Controller whether the CA NSM Event Manager pattern matching option has been set to case sensitive or insensitive.
- The ask clause and the answer scripts in the preceding example are optional. If you use answer scripts in your own code, the path names that you specify must be names of REXX programs that reside on the CA Automation Point machine.

Enabling CA NSM Event Manager to Issue Commands to CA Automation Point Sessions

The process of enabling Event Manager to issue commands to CA Automation Point sessions is similar to the process of enabling Event Manager to issue voice notifications.

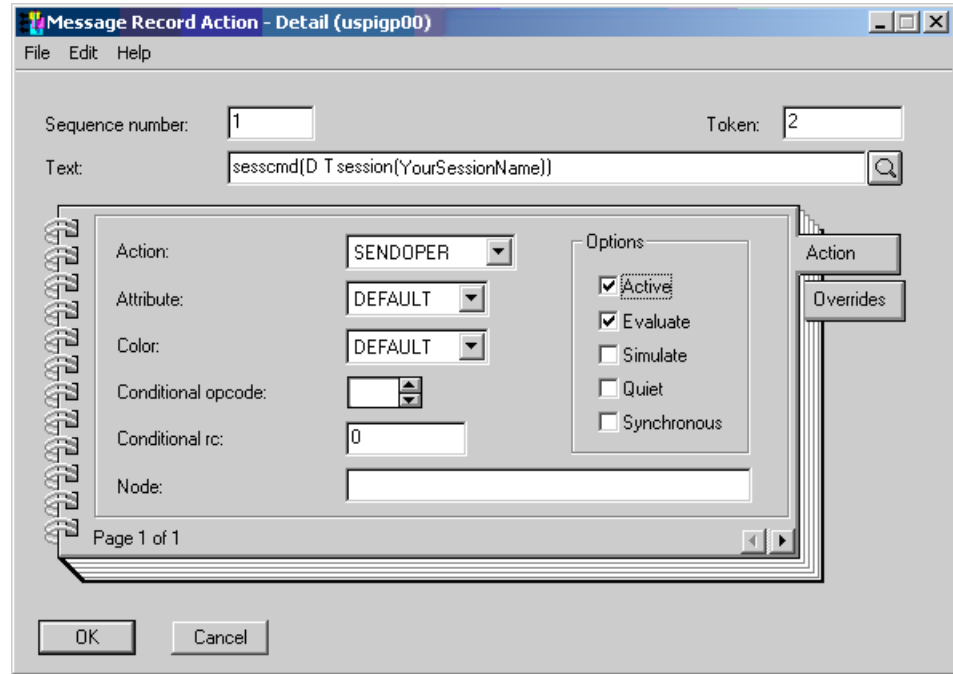
A CA Automation Point SESSCMD keyword is placed into the text field that is associated with the SENDOPER action. CA Automation Point remotely monitors the Event Manager log for session command requests and issues a SESSCMD when it encounters a request.

Note: Enter the SESSCMD keyword in lowercase. This enables the request to be recognized by the Event Traffic Controller whether the CA NSM Event Manager pattern matching option has been set to case sensitive or insensitive.

Because the SESSCMD keyword is the interface, you are not limited to using only z/OS commands. You can issue commands to any operating system for which an active session exists in CA Automation Point.

Note: For the proper syntax for the SESSCMD keyword, see the *Command and Keyword Reference Guide*.

The following illustration shows how to tell z/OS to display the time, using the CA NSM Message Record Action - Detail window (pay particular attention to the Text and Action fields):



Enabling CA NSM to Reply to CA Automation Point Messages

CA Automation Point places a value of AutomationPoint in the CA NSM Source field of any message it sends to CA NSM. To prevent CA Automation Point and CA NSM from entering an infinite loop, CA Automation Point does *not* read any messages from the CA NSM event log that have this value.

However, the CA NSM SENDOPER action retains the Source field from the original message. For example, when SENDOPER is used to place an NMFIND or SESSCMD request into the CA NSM event log (in response to a message originally sent from CA Automation Point) the Source field will contain the value AutomationPoint. Therefore, that value in the Source field causes CA Automation Point to ignore the NMFIND or SESSCMD request.

To avoid this situation, we recommend that when CA Automation Point captures an event that requires a notification or host command request, CA Automation Point should issue the request directly from CA Automation Point rules. CA Automation Point can then send a message to CA NSM acknowledging that it encountered the event and took corrective action.

However, if you decide that it is necessary to issue requests directly from CA Automation Point to CA NSM, follow following the procedure to ensure that CA Automation Point will accept the requests. In this procedure you configure the CA NSM message action record that specifies that SENDOPER is the action to be taken.

To configure the CA NSM message action record

1. Access the CA NSM Message Record Action – Detail dialog.
2. On the Action tab, specify your SENDOPER action.
3. Choose the Overrides tab. In the Source Field, specify any text other than Automation Point.

Enabling CA Automation Point to Receive SNMP Traps

CA Automation Point automatically receives all SNMP traps that are captured by Event Manager when the CATRAPD program is running on the remote Event Manager machine.

The CATRAPD program may not be automatically started when Event Manager is started. If the CATRAPD program is not running, you can enable it through the CA NSM Settings application (caogui.exe settings) by selecting the SNMP Trap Server Activated setting on the Component Activation Flags tab.

Troubleshooting the Event Traffic Controller

This section contains hints and tips, in question-answer format, about using the Event Traffic Controller.

Sending Commands from Event Manager

Symptom:

How can I send commands from Event Manager to hosts that are managed by CA Automation Point?

Solution:

1. From the Configuration Manager main window, choose Expert Interface, Automation, Events Interface, CA NSM Event Manager.
2. Select the Monitor host command requests from CA NSM event.
3. Check Enable Monitoring.
4. Add the CA NSM host for which you want CA Automation Point to monitor host session command requests.

5. Select the host by using the Include button to move it from the Available Hosts list to the Selected Hosts list.
6. Click Host Access Security and then specify a valid login user ID and password for the remote CA NSM host.
7. Save your changes.
8. Restart CA Automation Point.
9. On the CA NSM host, define an Event Manager message action to issue the command request to CA Automation Point.

Message Not Sent

Symptom:

How can I find out why a message was not sent from CA Automation Point to Event Manager or why CA Automation Point did not monitor events that it was configured to monitor?

Solution:

See the ASOTRACE log; all Event Traffic Controller components record their errors in this log.

CA Automation Point Not Producing Events as Configured

Symptom:

Why did all of the event types that were configured and enabled through Configuration Manager fail to produce events of any type in CA Automation Point?

Solution:

CA Automation Point must be recycled before such changes take effect. If the product has been recycled, ensure that you applied the configuration changes to the session definition set and rules files that it is using. Configuration Manager generates session definitions and rules that correspond to your configuration settings. If, upon exiting the CA NSM Event Traffic Configuration dialog, you do not apply the generated definitions and rules to the session definition set and rules files that are to be used by CA Automation Point, the event monitors may not send the events to existing sessions' names, and the events will be lost. If Event Manager has not been installed on the local CA Automation Point machine, CA Automation Point will not be able to dynamically load the required Event Management DLLs.

No Event Received by Event Manager

Symptom:

When I specify that a message, command, or trap is to be sent to Event Manager from within CA Automation Point rules, no event is received by Event Manger. Where could my problem lie?

Solution:

If the command contains a syntax error, the command will not be executed. If more than one instance of Event Manager is running at your site, the host name specified in the command may have sent the event to a different host from the one where you are watching the CA NSM Event Console Viewer. If you have not installed Event Manager on the local CA Automation Point machine, CA Automation Point is not able to dynamically load the required Event Management DLLs. For cases in which an actual error has occurred, more detailed information is placed in the ASOTRACE log.

User-filtered CA NSM Messages Not Received by CA Automation Point

Symptom:

None of the desired user-filtered CA NSM messages received by CA Automation Point

Solution:

The field values specified by the Filtering Criteria button within the CA NSM Event Traffic Configuration dialog are logically ANDed. Every value must match before the message is processed by CA Automation Point.

No SNMP Traps Received by CA Automation Point

Symptom:

CA Automation Point received no SNMP traps.

Solution:

CA Automation Point receives SNMP traps only from Event Manager. If Event Manager is not running, CA Automation Point will not receive SNMP traps. Even if Event Manager is running, it may not be configured to capture SNMP traps. Event Manager does not capture SNMP traps by default. To enable this feature, set the SNMP Trap Server Activated field to YES on the Component Activation Flags tab of the CA NSM Settings application (caogui.exe settings).

Suppressing Event Display Prevents CA Automation Point From Receiving Command

Symptom:

Why would suppressing the display of an event in the Event Manager Console Viewer prevent CA Automation Point from receiving a command request?

Solution:

If you are using the SUPPRESS keyword in Event Manager rules on a message that contains a command request for CA Automation Point, set Log Suppressed Messages to Y on the Event Management tab of the EM Settings application (caogui.exe settings). This allows the suppressed message to be recorded in the Event Manager log. CA Automation Point retrieves the command request from the log. The DISCARD keyword in Event Manager rules will not display or log the message, so CA Automation Point will never be able to retrieve the message.

Windows Event Log Not Appearing in CA Automation Point

Symptom:

Why would Windows event log messages not appear in CA Automation Point, even though new events are appearing in the Windows Event Viewer?

Solution:

Both Event Manager and CA Automation Point could be configured not to monitor Windows events. Enable this feature in one of the products by setting the Event Management LogRdr Agent Activated field to YES on the Component Activation Flags tab of the EM Settings (caogui.exe settings).

Two Windows Event Log Message for Each Windows Event

Symptom:

Why would two Windows event log messages be displayed by CA Automation Point for each Windows event that occurs?

Solution:

Both Event Manager and CA Automation Point can be configured to monitor the Windows event logs. If both products have been configured and CA Automation Point is also configured to monitor CA NSM messages, CA Automation Point retrieves and displays the Windows events that were also recorded by Event Manager. Reconfigure one of the products so that it will not monitor Windows events.

Too Many Messages When CA NSM Message Monitoring Enabled

Symptom:

A high volume of messages occurs within CA Automation Point when CA NSM message monitoring is enabled. How can this volume be reduced?

Solution:

The Filtering Criteria button within the CA NSM Event Traffic Configuration dialog lets you restrict the set of messages to be processed by CA Automation Point. Only those messages containing the values specified for the listed filtering fields will be processed by CA Automation Point.

Note: If no value is specified for any field, CA Automation Point will process every message.

Too Many Messages in Event Manager

Symptom:

A high volume of messages occurs within Event Manager when the "Automatically forward ALL messages" option is enabled during the configuration of the CA NSM host names to which CA Automation Point is to forward messages. How can this volume be reduced?

Solution:

If the level of message suppression on your monitored hosts is not sufficient to reduce the traffic flow from CA Automation Point to acceptable levels, do not enable Automatically forward ALL messages. If Automatically forward ALL messages is not enabled, no messages will be forwarded unless you explicitly ask CA Automation Point to do so. Add new MSGID rules to CA Automation Point that match the subset of messages that you want to forward to Event Manager. Specify the UNIFWD keyword on each applicable MSGID rule.

Message in Event Manager Log Not Displayed by CA Automation Point

Symptom:

When a message is sent from CA Automation Point to Event Manager, it is recorded in the Event Manager log. Because CA Automation Point is monitoring this log, why does it not retrieve and display this message?

Solution:

Because CA Automation Point sends messages to Event Manager at the same time it reads new messages from CA NSM, it could get into a feedback loop. CA Automation Point places the string AutomationPoint into the Source field when it sends messages to Event Manager. When CA Automation Point reads CA NSM messages, it discards any message whose Source field contains that value.

Second CA Automation Point Server Not Receiving NMFIND Requests

Symptom:

A CA Automation Point server is monitoring z/OS messages and forwarding them to the CA NSM Event Manager. In my Event Manager message actions, I use the SENDOPER command to issue an NMFIND request. This NMFIND request is then to be executed by a second CA Automation Point server, which is dedicated to sending notifications. Why does the second CA Automation Point server not receive any NMFIND requests, even though it has been configured to monitor my CA NSM Event Manager host?

Solution:

CA Automation Point places the string "AutomationPoint" into the Source field of every message it sends to CA NSM, and refuses to read back any message whose Source field contains that value. Your CA Automation Point z/OS message had a Source value of AutomationPoint. Therefore, the CA NSM Event Manager SENDOPER action places the NMFIND command into the CA NSM log. By default, SENDOPER leaves the original Source value in the new message that it places in the CA NSM log. Thus, this new NMFIND request has a source value of AutomationPoint, and your second CA Automation Point server will not read that message. On the CA NSM dialog where you specify the SENDOPER action, select the Overrides tab, and enter an overriding value (for example, from CA NSM) in the Source field.

Successful Return Code on Non-existent Host Name

Symptom:

Why would a successful return code result from UNIWTO when a non-existent host name is specified?

Solution:

If Event Manager is running on the same local host as CA Automation Point, Event Manager has store and forward enabled, and CA Automation Point sends a UNIWTO message to a remote node name that does not exist, the UNIWTO returns successfully under the assumption that the host is down. Event Manager expects to store the message and forward it when the remote host comes back up.

Finding Status of Notification Request on Remote Workstation

Symptom:

I can issue a notification request from Event Manager; however, this action is taken asynchronously on another workstation. How can I determine whether this remote operation succeeded or failed?

Solution:

The NMFIND command contains an optional parameter for this purpose. The ACKNOWLEDGE(UNI) parameter sends a message to Event Manager, indicating the success or failure of the notification request.

Using New Features of Event Traffic Controller With ADDRESS VOX

Symptom:

I want to use the new features of the Event Traffic Controller to enable Event Manager to request voice and paging functions from CA Automation Point. However, my site continues to use the low-level ADDRESS VOX facilities rather than using Notification Manager. Because the Event Traffic Controller uses the NMFIND command, can I take advantage of the new Event Traffic Controller facilities?

Solution:

Yes. In the CA Automation Point Distrib directory, rename the nmfind.rex program to nmfind.rex.orig. Create a new REXX program, named nmfind.rex, that contains your application logic and issues the desired ADDRESS VOX commands. Your NMFIND command will require some site-specific command line parameters. Use your site-specific parameters on the NMFIND command that you type into your Event Manager action definitions. Your NMFIND command will not acknowledge its success or failure to Event Manager unless you build that functionality into the application yourself. Otherwise, the behavior of the Event Traffic Controller will be the same as when it is used with Notification Manager. When you upgrade your version of CA Automation Point, do not forget that you must save your site-specific code stored in the file named nmfind.rex, because the installation of the new version will overwrite it.

Cannot Retrieve CA NSM Messages from Remote Host

Symptom:

I am unable to retrieve CA NSM messages from a remote host due to the following error:

```
apolluni->main-> Unable to open EventMgr log file for date = '19980709'  
on host = 'UniHost' for event type = 'UserFilter' due to error code = '-202'
```

Solution:

Ensure that the system date is set correctly on both machines. If the dates are correct, establish a login user ID and password on the remote host and specify that same user ID and password within the Remote Host Access Security dialog, which displays when you click Host Access Security on the CA NSM Event Traffic Configuration dialog. For a remote Windows CA NSM host, the user ID caunint is a good choice because it has already been automatically created during your installation of CA NSM.

Cannot Retrieve Windows Event Log Messages from Remote Host

Symptom:

I am unable to retrieve Windows event log messages from a remote host due to the following errors:

```
apollnt->main-> Function = 'OpenEventLog' failed with error code (5) = Access is denied.
```

```
apollnt->main-> Failed to open Windows Event Log = 'System' on host = 'Windowshost'.
```

Solution:

Establish a login user ID and password on the remote host and specify that same user ID and password within the Remote Host Access Security dialog, which displays when you click Host Access Security on the CA NSM Event Traffic Configuration dialog.

Using CA Automation Point to Monitor Windows Event Logs

CA Automation Point can monitor the system, application, and security Windows event logs. To configure CA Automation Point to monitor Windows event logs, open Configuration Manager. From the Configuration Manager main window, select Expert Interface, Automation, Events Interface, Windows Event Logs.

As described earlier in this chapter in the discussion of CA NSM Event Manager, messages from an event log of a Windows system are read and passed to CA Automation Point rules for processing within the WIN_EVENTS session. This log entry type is read directly from the Windows operating system. Because Event Manager also monitors these logs, this feature should be used primarily on stand-alone CA Automation Point systems. If you use this feature with Event Manager, ensure that Event Management LogRdr Agent Activated is set to No on the Component Activation Flags tab of the CA NSM Settings (caogui.exe settings) to avoid duplicating messages in CA Automation Point.

Note: Because CA Automation Point uses a common architecture to monitor both the event logs from CA NSM and the Windows operating system, you can find many details on monitoring Windows event logs in the CA NSM Interface portion of this chapter.

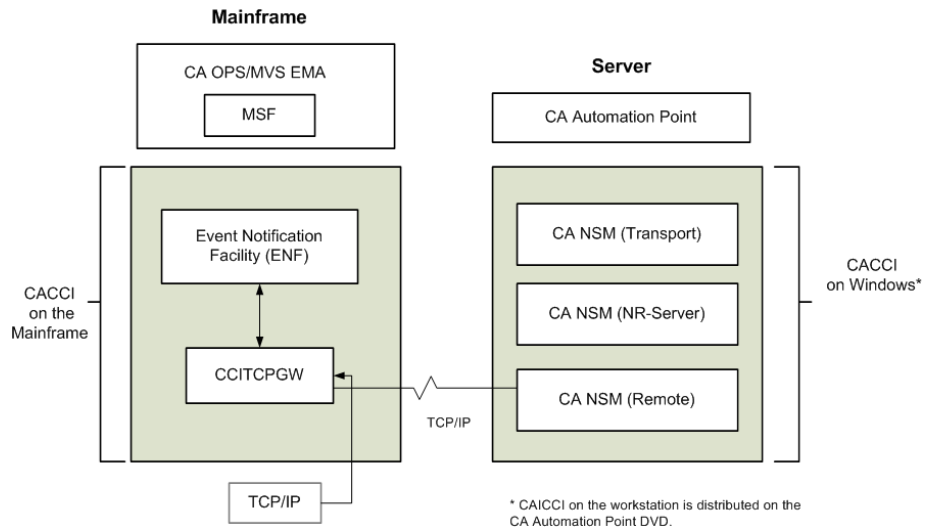
Chapter 14: Using the CA OPS/MVS Interface

A REXX API interface in CA Automation Point allows it to communicate with CA OPS/MVS. This general-purpose interface uses CAICCI as its communication protocol and allows CA Automation Point to act as a remote Multi-System Facility (MSF).

CAICCI (CAI Common Communications Interface) is a communications facility that allows CA solutions to communicate with one another. CAICCI is one member of the group of routines that comprises CA Common Services. CA Automation Point uses TCP/IP to communicate with CA OPS/MVS.

MSF is an optional feature of CA OPS/MVS that allows multiple copies of the product running on different z/OS images to communicate with each other using a variety of communication protocols. The MSF can also be used as an interface for communication between CA OPS/MVS and CA Automation Point. For details about the MSF, see the CA OPS/MVS documentation.

The following is an illustration of the components involved in the communication between CA OPS/MVS on the mainframe and CA Automation Point on Windows.



Sending Data from CA Automation Point to CA OPS/MVS

The CA Automation Point ADDRESS OPS host environment provides one-way communication from CA Automation Point to CA OPS/MVS. ADDRESS OPS provides the following functionality:

LIST

Returns a list of CA OPS/MVS hosts that are currently configured on the issuing CA Automation Point workstation

OPER

Sends a z/OS system command execution request to the CA OPS/MVS host

OSFTSO

Sends a TSO command execution request to the CA OPS/MVS Operator Server Facility (OSF) component

WTO

Sends a message to the CA OPS/MVS Multi-System Facility (MSF) component

See the *Command and Keyword Reference Guide* for further information about the ADDRESS OPS commands.

CA Automation Point sends data to CA OPS/MVS asynchronously, without acknowledging that the data was received by CA OPS/MVS. If an acknowledgement is desired, you can write CA OPS/MVS rules to trap the responses from the invoked REXX program and forward them to CA Automation Point. For details, see the CA OPS/MVS documentation.

Sending Data From CA OPS/MVS to CA Automation Point

ADDRESS AP Host Command Environment

From CA OPS/MVS, you can invoke the following through the ADDRESS AP host command environment:

- **REXX programs**—REXX command requests are queued for execution by AXCREXX.

Note: The length of the REXX argument list (REXX program name and its arguments) is limited to 505 characters. An incoming argument list longer than 505 characters is truncated.

- **NMFIND commands**—NMFIND requests appear in the Session NM_Requests window.

Notes:

- NMFIND requests are run through CA Automation Point Rules so the NMFIND command can be launched on the Automation Point desktop, not so you can write rules against them.
- CA Automation Point limits NMFIND command requests to 509 characters.

- **PPQ WRITE commands**—PPQ WRITE commands are submitted as soon as CA Automation Point receives them. If a PPQ WRITE command fails on the local CA Automation Point workstation, an error message displays in the AP Messages window.

Note: CA Automation Point limits PPQ WRITE requests to 30,000 characters.

ADDRESS WTO Host Command Environment

You can also issue a WTO (Write-to-Operator) message to CA Automation Point using the ADDRESS WTO host command environment. WTO messages are run through CA Automation Point rules and appear in the CA-OPS/MVS Messages window. You can also write rules to display WTO messages in the Merged Message window, color WTO messages, or highlight them.

See the CA OPS/MVS documentation for more information on these command environments.

Configuring the CA OPS/MVS Interface

There are three parts to configuring CA Automation Point for communications with CA OPS/MVS:

- Setting Up Communications to CA OPS/MVS
- Defining the CA OPS/MVS nodes with which you want to communicate
- Defining CA Automation Point systems to CA OPS/MVS

The following sections describe these processes.

Setting Up Communications to CA OPS/MVS

The first part of configuring CA Automation Point for communication with CA OPS/MVS is setting up CAICCI. CAICCI provides cross-platform communication.

For CA OPS/MVS on the mainframe and CA Automation Point on the workstation to communicate, you must install and configure CAICCI on each platform during setup. The CAICCI component is installed as part of the CA NSM and the CA Common Services installations. Before using the CA OPS/MVS interface, you must install either CA NSM or CA Common Services.

To set up CAICCI on the mainframe

1. Install and configure CAICCI on the mainframe. See the CA Common Services for z/OS documentation for further details.
2. Ensure that each copy of CA OPS/MVS has an active CAICCI connection on the Z/OS system. The CA-OPS/MVS INITCCI parameter must be set to YES.

To ensure that your TCP/IP connection is working

1. Ping the workstation. For instructions, see the information on how to verify that TCP/IP is working in the chapter "Viewing Remote Sessions."
2. Ping the mainframe system.

If you get no reply, modify the TCP/IP setup on the Windows workstation and the mainframe. Contact your network administrator for assistance.

To install CAICCI on the CA Automation Point workstation

1. Run setup.exe from the CA Automation Point installation CD.
2. From the setup dialog, choose Modify.
3. Navigate the Features tree to open the menu item Interfaces to Companion Products, and then select CA-OPS/MVS Interface.

To define a connection between the nodes on the mainframe and the CA Automation Point workstation

1. From Configuration Manager, select Expert Interface, Automation, Events Interface, CA-OPS/MVS Interface
2. Edit the CCI Remote configuration that appears. Add REMOTE statements for connection to each of the mainframes where your CA OPS/MVS product is installed.

The CAICCI configuration file should include both LOCAL and REMOTE statements. The format is the similar for the LOCAL and REMOTE statements.

The **LOCAL** statement has the following format:

```
LOCAL = TCP/IP_name CCI_name [buffersize] startup_options [alias_options]
port_options retry_interval
```

The **REMOTE** statement has the following format:

```
REMOTE = TCP/IP_name CCI_name [buffersize] startup_options [alias_options]
[heartbeat_options] port_options retry_interval
```

TCP/IP_name

Specifies either an IP address or a name that is used as input to a name service to retrieve an IP address. You may use the TCP/IP name with the PING command to determine whether a remote connection is live. The default is the TCP/IP host name. It does not require a logical connection to the CCI name.

To find the TCP/IP name for the REMOTE statement, enter the following:

```
HOMETEST
```

You will see something similar to the following:

```
TCP Host Name is: MVS01
```

CCI_name

Specifies the logical name CAICCI uses to identify this host. This is the system name, which may or may not be the same as the IP hostname. This name may be as long as 64 characters, but an alias must be used for names greater than eight characters.

For the REMOTE statement that defines the mainframe, the CAICCI name is the value specified by SYSID (xxxx) in the //ENFPARMS DD statement. This value is also on the PROTOCOL statement in //ENFPARMS.

buffersize

(Optional) Specifies the maximum buffer CAICCI will receive or send over the socket, a value used for segmenting the data transfer. Each side of the connection may have this set to a different value, between 1024 and 32768. The lesser of the two values will be used. It is generally not necessary to alter this field.

Important! Contact CA Technical Support before changing the buffer size.

startup_options

Tells ccirmtd (the CAICCI remote daemon) whether to initiate a connection--sometimes you may want only one side to initiate the connection. Not having the server start connections eliminates a succession of messages when CAICCI is recycled.

STARTUP

Tells CAICCI to attempt a remote connection when activated, whereas

NOSTART

Implies that the remote system will be initiating the connection to the node.

alias_options

(Optional) Specifies an alias name used to differentiate multiple remote computers having exactly the same first eight characters (when their host names exceed eight characters).It specifies that, for hosts with a host name greater than eight characters, this name is used for internal CAICCI structures. This alias need not appear in DNS or IP host file. It must be unique and consistent across nodes.

When an alias is used on the LOCAL line for a host with a name greater than eight characters, all hosts to which it will be connected must have a REMOTE line for this host with this same alias defined. The format is ALIAS=*aliasname*.

port_options

Allows you to specify an alternate port for this specific connection only. Sometimes you may have two groups of hosts. One group may still use the old 7000 port and the new ones may use the new 1721 port. Any host from the first group wishing to communicate with the second group must be made aware that the connection should be made to a different port. The format is PORT=1721. Do not change this value.

To find the port number for the REMOTE statement

1. From the ISPF Command Shell on the mainframe, enter NETSTAT.
2. Under the USERID, look for CCITCPGW with a state of LISTEN.
3. Under Local Socket, you will find a value similar to this one:

(0.0.0.0..7000)

This value is the port number.

heartbeat_options

(Optional) Specifies whether the CAICCI remote heartbeat feature is enabled. The heartbeat feature detects connection loss with the remote node if the remote node terminates the connection without notification. The format to enable the heartbeat feature is HEARTBEAT=YES. The format to disable the heartbeat feature is HEARTBEAT=NO.

You can also define the heartbeat functionality by using Windows environment variables as follows:

CAI_CCIRMT_HEARTBEAT

When this variable is set to YES, the heartbeat option is enabled to all the nodes listed in the ccirmt.rc file. However, explicitly setting the option in the ccirmt.rc files takes precedence over the global flag.

CCIRMPINGTIMEOUT

By default, the wait time for an acknowledgment to the ping from the remote node is 30 seconds. You can alter this value by setting the variable to a desired number of seconds. The minimum and maximum values allowed are 15 and 45 seconds. The value defaults to 30 seconds if any other value is set.

You can set these variables using the Configuration Settings GUI (Options, CCI Options tab) or from the command prompt by running the following commands:

```
cautenv setopt CAI_CCIRMT_HEARTBEAT value
```

```
cautenv setopt CCIRMPINGTIMEOUT value
```

retry_interval

Determines how ccirmt behaves if the connection is dropped and specifies the number of seconds between retry connect attempts. The format is as follows:

- If $x=0$, ccirmt will not retry the connection.
- If $x=-1$, ccirmt will start with a two second retry interval and double after each unsuccessful retry attempt.
- If $x > 0$, ccirmt will wait n seconds between retry attempts.

Note: Retry interval is mainly used in conjunction with the nostart option to allow the server to sit passively and wait for incoming connection requests. If a client host goes down, the server will not attempt to reconnect.

3. If necessary, override the default local CCI application name.

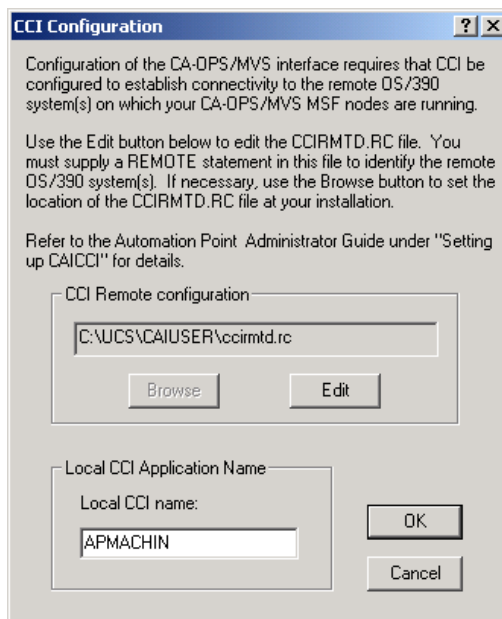
The local CCI name is the eight-character name of the workstation. This name *must* be unique within your network. If it is not, you should override it.

Important! Use caution when overriding this name because CA Automation Point uses it as a unique identifier for this system in its network communications.

To override the default local CCI name

- a. Start the Configuration Manager. Select Expert Interface, Automation, Events Interface, CA-OPS/MVS Interface, CCI Settings.

The CCI Configuration dialog displays.



- b. Type the name of the desired local CAICCI application in the Local CCI name field, and then click OK.
- c. Click the **?** for a description of each field on the CCI Configuration dialog.

To verify that CAICCI has established communications, issue the following command:

```
ccii
```

The following is an example of the output of this command:

```
12:21:11 c:\tngfw\bin>ccii
Oid(USABCP4F,OPS/MVS.USABCP4F ) Did( , ) type(L)
Oid(A44SENF,CA-TOP-SECRET ) Did( , ) type(R)
Oid(A44SENF,DRASQADRAS01 ) Did( , ) type(R)
Oid(A44SENF,CAS9VTAMCW410000 ) Did( , ) type(R)
Oid(A44SENF,CAS9VTAMCW410000 ) Did( , ) type(R)
Oid(A44SENF,CAS9VTAMCW410000 ) Did( , ) type(R)
Oid(A44SENF,W410_SPAWN_SERVER ) Did( , ) type(R)
Oid(A44SENF,W410_SPAWN_SERVER ) Did( , ) type(R)
Oid(A44SENF,W410_SPAWN_SERVER ) Did( , ) type(R)
Oid(A44SENF,MVS_START_SERVER ) Did( , ) type(R)
Oid(A44SENF,OPS/MVS.OPS44F ) Did( , ) type(R)
Oid(A44SENF,SYS1202 ) Did( , ) type(R)
Oid(A44SENF,SYST1400 ) Did( , ) type(R)
Oid(A44SENF,OPS/MVS.OPS44M ) Did( , ) type(R)
```

In this example, USABCP4F is the Windows workstation on which CA Automation Point is installed. The lines with the A44SENF entries indicate that the CAICCI connection has been made to a mainframe. The CCII command displays the CAICCI applications that are available.

Defining CA OPS/MVS Nodes

The next part of configuring CA Automation Point for communication with CA OPS/MVS is defining the CA OPS/MVS nodes with which CA Automation Point communicates.

To define CA OPS/MVS nodes

1. Start Configuration Manager.
2. Select Expert Interface, Automation, Events Interface, CA OPS/MVS Interface. The CA OPS/MVS Event Traffic Configuration dialog displays.
3. Move the desired CA OPS/MVS nodes from the Available MSF nodes list to the Selected MSF nodes list by clicking Include or Include All.
4. Click Save.
5. Restart the CA OPS/MVS Communications Interface for the changes to take effect.

Note: For information about the other fields in the Event Traffic Configuration dialog, click the help button.

Defining CA Automation Point Systems to CA OPS/MVS

The third part of configuring CA Automation Point for communication with CA OPS/MVS is defining CA Automation Point systems to CA OPS/MVS. To do so, issue these commands from an OPS/REXX program:

```
Address OPSCTL "MSF DEFINE MSFID (CCI name) APPLID (TCP/IP name) AP"
```

```
Address OPSCTL "MSF ACTIVATE MSFID (CCI name)"
```

Note: You can also define CA Automation Point systems to CA OPS/MVS by using the OPSVIEW MSF Control panel. See the CA OPS/MVS documentation for further details.

Recycling the CA OPS/MVS Communications Interface

You can stop and restart, or recycle, the CA OPS/MVS interface independently of the CA Automation Point desktop. Recycling the interface is useful in the following situations:

- You have added or removed OPS MSF nodes from the CA OPS/MVS Interface configuration (by choosing Expert Interface, Automation, Events Interface, CA-OPS/MVS Interface from the Configuration Manager main window)
- CAICCI communications have been disrupted
- One or more of your CA OPS/MVS hosts have been reconfigured to communicate with your CA Automation Point product.

You can recycle the CA-OPS/MVS Communications Interface manually, programmatically, or automatically (based on an event).

To manually recycle the CA OPS/MVS interface

1. From the CA Automation Point desktop, go to the CA-OPS/MVS Messages window.
2. Select Action, Recycle OPS Communications Interface.

To programmatically recycle the CA OPS/MVS interface

1. Run an AP REXX program to issue the following command:

```
address AXC 'SESSCMD /@"RECYCLE_OPS"/ SESSION(AXC)'
```

To automatically recycle the CA OPS/MVS interface based on a monitored event

1. Write a CA Automation Point rule to recycle the interface. For example:

```
MSGID(RECYCLEOPS) SESSCMD(("@RECYCLE_OPS") SESSION(AXC))
```

MSGID

Specifies the initial identifying character string of an event monitored by CA Automation Point reported as a message.

SESSCMD

Specifies a keystroke string that can contain text or keystrokes to be sent to the session that issued the current message.

SESSION

Specifies the session name or type.

For more information about these keywords, see the *Command and Keyword Reference Guide*.

Command-level Security

When the CA OPS/MVS OSFSECURITY parameter has a value of CHECKUSERID, the following rules apply:

- All messages and commands sent directly from CA Automation Point to CA OPS/MVS are associated with the z/OS user ID defined by the CA OPS/MVS APDEFAULTUSERID parameter.
- If you are using Remote Viewer to access CA Automation Point sessions, all messages and commands are associated with the Remote Viewer username. The Remote Viewer username overrides the CA OPS/MVS APDEFAULTUSERID parameter, and must conform to the restrictions of the z/OS security package.
- The z/OS user ID defined by the CA OPS/MVS APDEFAULTUSERID and the Remote Viewer username(s) should be granted the proper security access on z/OS.

If the CA OPS/MVS OSFSECURITY parameter has a value of NOSECURITY, commands sent from CA Automation Point to CA OPS/MVS execute with the security attributes of the OSF TSO servers.

For details about these parameters, see the CA OPS/MVS documentation.

Chapter 15: Using AS/400 Manager

This section contains the following topics:

- [What Is AS/400 Manager](#) (see page 401)
- [Installing and Configuring AS/400 Manager](#) (see page 403)
- [Managing Multiple AS/400 Systems](#) (see page 414)
- [Test Starting AS/400 Manager](#) (see page 418)
- [Designing Automation for AS/400 Manager](#) (see page 418)
- [Operating AS/400 Manager](#) (see page 425)
- [Technical Information](#) (see page 429)
- [AS/400 Default Logon Screen](#) (see page 440)

What Is AS/400 Manager

The CA Automation PointAS/400 Manager application allows you to use the power and flexibility of CA Automation Point to manage multiple AS/400 systems. In addition, the AS/400 Manager application:

- Is easy to understand and use.
- Supports multiple-system environments with several AS/400 menu languages. AS/400 Manager recognizes AS/400 dialogs in Danish, English, German, Norwegian, and Swedish. The application also has provisions for a user-specified language.
- Uses standard operator dialogs without any special software.
- Requires no special software on the AS/400 system.

AS/400 Manager Application Commands

The AS/400 Manager application provides a set of commands you can use to do the following:

- Display the status of the AS/400 systems that CA Automation Point manages. (The application's AS400SSD plot window also continuously displays status information graphically.)
- Display configuration data for the application
- Issue commands to an AS/400 system or a group of systems
- Suspend and resume the application's control of an AS/400 system

For more information about these commands, see the section [Summary of the REXX Programs](#) (see page 430) in this chapter.

AS/400 Manager Terminology

The section that follows includes terminology that you should become familiar with before using AS/400 Manager.

alert primary focal point

Identifies the AS/400 system services control point, from which the AS/400 Manager application reads alerts that other AS/400 systems have forwarded to the focal point.

alert session

Identifies a CA Automation Point session processing alerts from a local AS/400 system that is an alert primary focal point. Each alert session (you can have more than one) is signed on automatically to its AS/400 system via a REXX program called AS400CTL.

local AS/400 system

Identifies a directly connected AS/400 system. A local system can be connected using a TN3270 session. Physical location has no bearing on whether a system is considered remote or local.

remote AS/400 system

Identifies an AS/400 system that is accessed indirectly through another AS/400. Remote systems can be connected using either of the following:

- Display Station Passthru through a local AS/400 system
- TCP/IP Telnet

Physical location has no bearing on whether a system is considered remote or local.

Connecting CA Automation Point and AS/400

Use a TN3270 or TN 5250 connection to connect CA Automation Point to an AS/400 system. Because TN5250 is the native protocol for i5/OS, we recommend you use that connection protocol.

AS/400 System Requirements

To use the AS/400 Manager application, the AS/400 current system name and the AS/400 local control point name must be defined in OS/400 as the same name. Issue the OS/400 command DSPNETA to check whether the names match. A screen similar to the following displays:

```

Display Network Attributes

Current system name . . . . . System: S44A0060
Pending system name . . . . . : S44A0060
Local network ID . . . . . : APPN
Local control point name . . . . . : S44A0060
Default local location . . . . . : S44A0060
Default mode . . . . . : BLANK
APPN node type . . . . . : *NETNODE
Maximum number of intermediate sessions . . . . . : 200
Route addition resistance . . . . . : 128
Server network ID/control point name . . . . . : *LCLNETID *ANY

More...

Press Enter to continue.
F3=Exit F12=Cancel

```

If a change is required, it is safer to change the AS/400 system name. To do so, issue the CHGNETA command and re-IPL the relevant AS/400 system.

Installing and Configuring AS/400 Manager

This section discusses how to install and configure AS/400 Manager.

Setting Up AS/400 for Communication with AS/400 Manager

To set up the AS/400 system for communication with CA Automation Point and the AS/400 Manager application

1. Obtain either the host alias or IP address for the AS/400 system. Verify that TCP/IP is properly installed and configured on the AS/400 system to be managed.
2. Get your AS/400 system name. The AS/400 should display its system name in the upper-right corner of its logon screen on line 2.

If the AS/400 system name is not displayed on the right side of line 2, AS/400 Manager will not recognize the system. You must reconfigure the logon screen, placing the system name in the same location specified by the screen example shown in the section [AS/400 Default Logon Screen](#) (see page 440).

3. Create a user profile for CA Automation Point to use on the AS/400 system to be managed. The topic [Creating a User Profile for CA Automation Point](#) (see page 407) describes how to create an AS/400 user profile for CA Automation Point. AS/400 Manager uses this user profile when it logs in to an alert session and an automation session.

If you choose to have a manual session, manually log in using whatever user IDs you already use in your AS/400 operations.

4. Create the KBD3270 CL program on the AS/400 system according to the topic [Creating the 3270 Keyboard Map Routine in AS/400](#) (see page 409). This program is called from CA Automation Point scripts to enable the use of the PF keys.
5. Using the instructions in the section [Setting Up the Alert Function in AS/400](#) (see page 409), enable the OS/400 alert facility on the AS/400 system to be managed.

Setting Up Your Workstation for Communication with AS/400

To set up the CA Automation Point workstation for communication with the AS/400 system

1. Before you install AS/400 Manager, create backup copies of all the CA Automation Point files you have customized.
2. Verify that TCP/IP is properly installed and configured on the workstation and able to communicate with the AS/400 system. To do so, open an operating system command prompt window and type *one* of the following commands:

PING hostname

PING ip-address
3. Configure two TN5250/TN3270 sessions to the AS/400 system. Save the sessions with the names AS400GET and AS400AUT. Optionally, configure a manual session named AS400MAN.

To set up the workstation in a multiple AS/400 system configuration, see the section [Setting Up Your Workstation When Connecting to Multiple Stand-alone Systems](#) in this chapter.

Customizing CA Automation Point Files to Install AS/400 Manager

You need to insert configuration information for AS/400 Manager into your CA Automation Point files.

To insert configuration information for AS/400 Manager into your CA Automation Point files

1. Customize the CA Automation Point rules file.

Add the AS/400 Manager rules to your CA Automation Point rules file, AXCRULES.RUL. If you have rules you want to keep, use an editor to merge the AS400RUL.rul contents into your current AXCRULES.RUL file. Everywhere you see the system name of S44A0060, change it to your actual system name.

2. Update your active session definition set by following these steps:

- a. Create three non-automated TN5250/TN3270 sessions named "AS400GET" (alert session), "AS400AUT" (automation session), and "AS400MAN" (manual session).

For each session:

1. Set the system name to the actual name of your system.
2. Choose an appropriate terminal type. If you are unsure, choose 3278_2, which is the default terminal type.
3. Choose the "AS400" menu.
- b. Create an automated asynchronous session named "AS400RUL":
 4. Set the system name to "AS400RUL".
 5. Choose the "AS400" menu for this session.
 6. Choose "VT100" as the terminal for this session.
 7. In the Local Session Settings for this session, set the Session Timeout Interval to zero.
 8. In the Communication Settings for this session:
 - a. Set Communication Device to "MEMORY."
 - b. Set Parity to "Ignore."
 - c. Set CR/LF Interpretation to "Yes".
 - c. Create a function window named "AS400SSD" of type "PLOT" for AS/400 System Status.

3. Update the REXX settings by adding an initialization REXX program named "AS400SET". You do this from the REXX Settings dialog in Configuration Manager.

4. If you have not already done so, create a user profile for CA Automation Point to use. For instructions, see the section [Creating a User Profile for CA Automation Point](#) (see page 407) in this chapter.

Customize the CA Automation Point script, AS400GET.SCR, which is the sign-on/recovery script for the alert session.

Edit the AS400GET.SCR file. On the line "KEY=(userid@Tpassword@E)" substitute "userid" with the user ID you want to log on to. If your AS/400 system requires a password, substitute "password" with the correct password as well. If no password is required, delete the following text, which represents a TAB keystroke and the password:

```
@Tpassword
```

Do not delete the text "@E," which represents a carriage return.

If the AS/400 system is using a non-English language or customized log-on screen, you may need to customize the search text in this file.

5. Customize the AS/400 Manager configuration file, AS400CFG.COM, by running the AS4CFG program, as described in the section [Configuring AS/400 Manager](#) (see page 406) in this chapter.

To customize CA Automation Point files in a multiple AS/400 system configuration, see the section [Customizing CA Automation Point Files to Install AS/400 Manager When Connecting to Multiple Stand-alone Systems](#) (see page 415) in this chapter.

Configuring AS/400 Manager

To configure AS/400 Manager, you must run the configuration program AS4CFG. The program updates the values in the AS400CFG.cmd file. Initially, the configuration program supplies defaults that define a single local AS/400 system.

To run the AS4CFG program

1. From Configuration Manager, choose Expert Interface, Automation, AS/400 Manager.
2. In the AS/400 Manager Configuration window, select the AS400AUT session name, and then Click Change.
3. In the AS/400 Session Description window, select the system name S44A0060, and then Click Change.
4. In the AS/400 System Description (AS400AUT) window, under System Name, change the system name to the name of your system.

5. Select the tag name S44, Click Change, and then enter a tag (alias) name for your system.
6. Select the appropriate language code. If the title of the AS/400 WRKALR panel is displayed in English uppercase characters (for example, WORK WITH ALERTS), change the language code for the session from ENG to ENU.
7. Click OK until you return to the AS/400 Manager Configuration window.
8. Repeat Steps 1 through 6 for the remaining default session name, AS400GET.
Note: The default tag name is A44. Change it to a tag name for your system.
9. Click Save to save the settings to the AS400CFG.cmd file.

To configure AS/400 in a multiple AS/400 system configuration, see the section [Configuring AS/400 Manager When Connecting to Multiple Stand-alone Systems](#) (see page 415) in this chapter.

Creating a User Profile for CA Automation Point

As part of your setup for AS/400 Manager, you need to create a user profile for CA Automation Point to use on the AS/400 system to be managed.

To create a user profile for AS/400

1. Issue the following OS/400 command:

```
CRTUSRPRF
```

2. Press the F4 key to display the Create User Profile dialog.
3. Enter the following parameter values:

User profile

Specifies the profile name. This name can be up to eight characters, such as ASMANUSR

User class

SYSOPR (This is suitable for most environments.)

Text 'description'

Describes the profile, such as "User profile for CA Automation Point-AS/400 interface"

4. When you have finished typing in the required parameter values, press Enter to save those values.

User Profile Default Values

The following table lists the default parameter values for the Create User Profile dialog.

Note: You do *not* have to enter these values—they are automatically displayed.

Parameter	Default value
User password	*USRPRF
Set password to expired	*NO
Status	*ENABLED
Assistance level	*SYSVAL
Current library	*CRTDFT
Initial program to call	*NONE
Initial menu	MAIN
Library	*LIBL
Limit capabilities	*NO

Creating the 3270 Keyboard Map Routine in AS/400

Note: Creating a 3270 keyboard map is necessary only if you choose your AS/400 sessions to be type TN3270. If you choose TN5250, this step is not necessary.

To use the PF keys on AS/400, create a keyboard map routine for each AS/400 system that AS/400 Manager will control. The keyboard map routine is a CL program that must be compiled and must reside in your library list.

```
PGM CHGKBDMAP PF1(*F1) PF2(*F2) PF3(*F3) PF4(*F4) PF5(*F5) +
      MONMSG MSGID (CPF8701 CPF0000)
      PF6(*F6) PF7(*DOWN) PF8(*UP) PF9(*F9) +
      PF10(*F10) PF11(*F11) PF12(*F12) +
      PF13(*F13) PF14(*F14) PF15(*F15) +
      PF16(*F16) PF17(*F17) PF18(*F18) +
      PF19(*F19) PF20(*F20) PF21(*F21) +
      PF22(*F22) PF23(*F23) PF24(*F24) +
      PA1PF1(*HELP) PA1PF2(*HLP3270) +
      PA1PF12(*SYSREQ)
ENDPGM
```

Name the CL program KBD3270. The sample initialization scripts for AS/400 Manager call the program using the KBD3270 name when the application starts.

Note: The KBD3270 program must properly map the PF keys on the AS/400 system for the AS/400 Manager to function properly. Contact your AS/400 administrator if you have any problems with the KBD3270 program.

Setting Up the Alert Function in AS/400

Issue the OS/400 CHGNETA command to each of your managed AS/400 systems. The command text to use depends on the role your AS/400 plays in processing alert functions.

Local AS/400 System Non-Focal Point

To have a local AS/400 system that will not be a focal point generate alerts, issue this command:

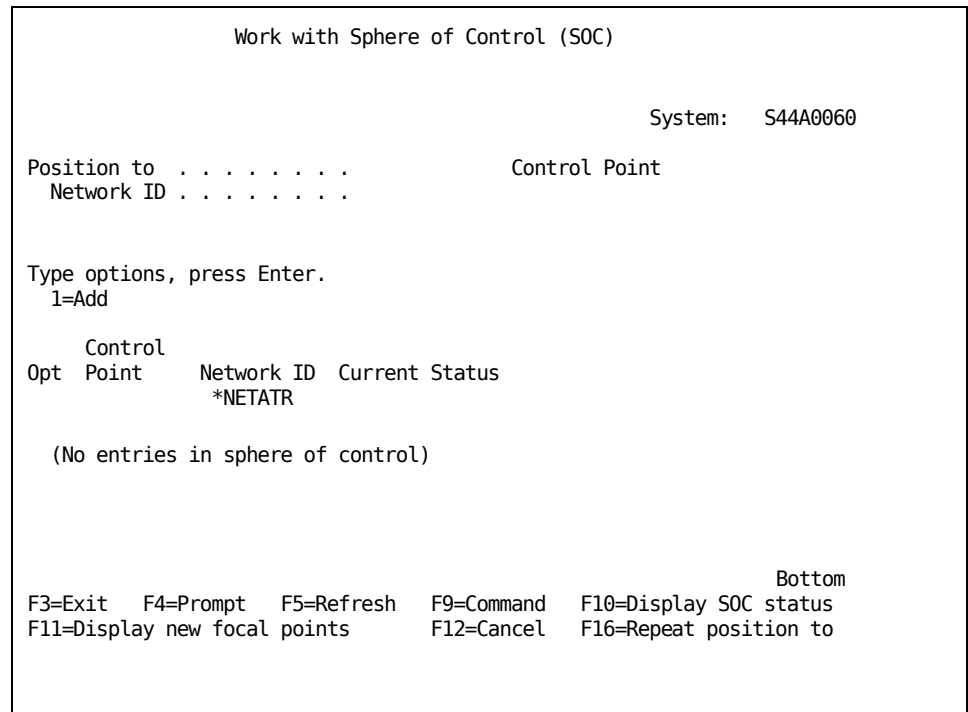
```
CHGNETA ALRSTS(*ON) ALRPRIIFP(*NO) ALRLOGSTS(*LOCAL)
```

Local AS/400 System as an Alert Primary Focal Point

To have your AS/400 system work as an alert primary focal point for remote AS/400 systems, issue this command:

```
CHGNETA ALRSTS(*ON) ALRPRIFFP(*YES) ALRLOGSTS(*ALL)
```

You need to explicitly define (in the local AS/400 system's sphere of control) the remote AS/400 systems that will send alerts. To do this, issue the WRKSOC command to add network node control point names. The Work with Sphere of Control screen displays; this screen shows the parameters for the WRKSOC command.



AS/400 Manager handles messages from a system that is set up as a primary alert focal point without any additional setup.

Remote AS/400 System

To have a remote AS/400 network node (defined in the primary focal point sphere of control) send alerts to its focal point, change the network attribute by issuing this command:

```
CHGNETA ALRSTS(*ON) ALRPRIFFP(*NO) ALRLOGSTS(*NONE)
```

Understanding Configuration Settings

To specify configuration settings for AS/400 Manager, run the AS4CFG program, which updates values in the AS400CFG.cmd file. This section is provided as a reference, to help you understand the settings in AS400CFG.cmd.

Note: Do not edit the AS400CFG.CMD file. Running the AS4CFG program sets all configuration values.

AS400CFG.cmd Settings

The statements in the AS400CFG.CMD file are described in the following table.

AS400_MAX_SESSIONS = *n*

Specifies the total number of alert and automation sessions used.

AS400_SESSION_NAME_ *n* = *name*

Specifies the *name* of an alert or an automation session defined to the AS/400 Manager interface and (together with the following two status variables) repeats *n* times where *n* is equal to the value specified in the variable AS400_MAX_SESSIONS.

AS400_SESSION_TYPE_ *n* = ALR | AUT

Specifies whether the session type is an alert or an automation session.

AS400_sessname_MAX_SYSTEMS = *n*

Specifies how many AS/400 systems are connected in the session name *sessname*.

AS400_sessname_NAME_ *n* = *sysname*

Specifies the AS/400 system name (*sysname*) for each of the AS/400 systems connected to the session *sessname*.

The following status variables apply to a single AS/400 system. Each line is repeated from 1 to 5 times, depending on the number of AS/400 systems that are connected to session *sessname*.

AS400_sessname_TAG_n_0 = m

Specifies the number of AS/400 tag names defined for AS/400 system number *n* that is connected to the session *sessname*.

AS400_sessname_TAG_n_m = tagname

Specifies an AS/400 tag (*tagname*) from 1 through *m* times for AS/400 system number *n* that is connected to the session *sessname*.

AS400_sessname_TERM_n = termname

Specifies a unique terminal ID or LU name (*termname*) to be used for identifying a terminal session before signon occurs to an AS/400 system.

AS400_sessname_PLIMIT_n = i

PLIMIT variables represent the maximum number of alerts (*i*) that can be processed from AS/400 system number *n* in a single pass, and are only defined for alert sessions. An initial default is set in the configuration, but this value can be changed dynamically in response to either the QTIME information or to some other user-determined criteria. The AS400CTL program responds to changes in the PLIMIT variables and adjusts its processing priorities accordingly.

AS400_sessname_TIMER_n = 0

The suspension timer value is set to 0 (zero) at CA Automation Point startup for each AS/400 system controlled by an alert or automation session.

AS400_sessname_RECINT_n = m

RECINT variables contain the time interval in minutes (*m*) between successive attempts to recover the associated system after a FAILED status occurs for that system.

AS400_sessname_RECMAx_n = m

RECMAx variables contain the number of recovery attempts (*m*) for the associated system after a FAILED status is recorded for that system.

AS400_sessname_RECcnt_n = m

RECCnt variables contain the current number of recovery attempts (*m*) that have been initiated for the associated system after a FAILED status has been recorded. A special rule sets the value to 0 (zero) after the system's status returns to ACTIVE.

AS400_sessname_QTIME_n = 0

The alert queue time value is set to 0 (zero) at startup and is only defined for alert sessions.

QTIME variables are maintained by the AS400CTL program and represent, for the latest alert from the relevant system, the difference (in minutes) between the workstation time when the alert is received and the recorded AS/400 time when the alert was generated.

AS400_sessname_LANG_n = yyy

Specifies the national language used by the current AS/400 system. Valid values for *yyy* are DAN, ENG, GER, SWE, USR, or ENU (English uppercase).

AS400_sessname_ST_n = RSTART | CLOSED

The initial status value is set for each AS/400 system connected to the session (*sessname*). RSTART indicates that an initialization script should be run for the relevant AS/400 system. CLOSED indicates that AS/400 Manager should ignore the system.

AS400_sessname_CST_n = STRTUP

The copy status value is set at initialization for each AS/400 system connected to the *sessname* session. The copy status variable is used by AS/400 Manager to record status changes for the CA Automation Point Plot feature.

AS400_sessname_SCRIPT_n = *scriptname*

Specifies the name of the initialization SCRIPT (*scriptname*) to be run against each AS/400 system in the session (*sessname*).

AS400_GLOBAL_GRP_0 = *i*

Specifies the number of global groups (*i*) that are defined.

AS400_GLOBAL_GRP_i_N = *groupname*

Specifies a global group (*groupname*) for a set of AS/400 systems. Variations of this status variable repeat *i* times.

AS400_GLOBAL_GRP_i_0 = *j*

Specifies the number of AS/400 systems (*j*) belonging to the global group *groupname* and is specified once for each global group (*i*).

AS400_GLOBAL_GRP_i_j = *sysname*

Specifies the name of each AS/400 system (*sysname*) belonging to the global group number *i* and is specified *j* times.

Managing Multiple AS/400 Systems

If you plan to connect to multiple AS/400 systems, you must do the following:

- Choose a method for connecting to the systems.
- Set up AS/400 to connect to the systems
- Access the systems.

Choosing a Method for Connecting to Multiple Systems

To connect to more than one AS/400 system, use one of the following methods:

- Set up an AS/400 system as an alert primary focal point to which multiple AS/400 systems can route their alert messages.

For instructions, see the section [Setting Up the Alert Function in AS/400](#) (see page 409) in this chapter.

Note: If you need to issue a system command to another AS/400 system, you must define an automated session for that system.

- Set up AS/400 Manager to process alert messages from multiple stand-alone AS/400 systems. In this case, you must set up AS/400 Manager to connect to each additional system. The procedures for this are described in the sections that follow.

Setting Up AS/400 to Connect to Multiple Stand-alone Systems

To set up AS/400 Manager to connect to multiple stand-alone systems, perform the following procedures.

- Set up AS/400 for communication with AS/400 Manager.
- Set up the workstation to communicate with the AS/400 system.
- Customize CA Automation Point files.
- Configure AS/400 Manager.

Note: The procedures you follow here parallel the ones you use to install AS/400 Manager using the sample configuration procedure.

Repeat the steps in these procedures for each system that you want to add, using new session names, file names, short names, and so on for each system.

Setting Up AS/400 to Communicate with AS/400 Manager

For instructions on preparing the AS/400 system to communicate with CA Automation Point and the AS/400 Manager application, see the section [Setting Up AS/400 for Communication with AS/400 Manager](#) (see page 403) in this chapter.

Setting Up Your Workstation When Connecting to Multiple Stand-alone Systems

To set up a workstation to connect to multiple standalone systems, follow this procedure:

1. Verify that TCP/IP is able to communicate with the AS/400 system. To do so, open an operating system command prompt window and type **one** of the following commands:

PING hostname

PING ip-address

2. Configure two sessions to the AS/400 system. Save the sessions with the names you prefer (for example, SYS2GET and SYS2AUT). One is the alerts session, the other the automation session. Optionally, you can configure a manual session also. You may choose to define these sessions as connection type TN5250 or TN3270.

Customizing CA Automation Point Files to Install AS/400 Manager When Connecting to Multiple Stand-alone Systems

To insert configuration information for AS/400 Manager into your CA Automation Point files

1. Create backup copies of all your CA Automation Point files.
2. Customize the CA Automation Point rules file by doing the following:
 - a. Edit the sample file AS400ADG.rul in the CA Automation Point Distrib directory. Change all occurrences of AS400GET to the alerts session name used for this system (for example, SYS2GET).
 - b. Change all occurrences of the suffix `_1` to `_2`.
 - c. Copy the new contents of AS400ADG.rul into your active rules file.
 - d. Edit the sample file AS400ADA.rul in the CA Automation Point Distrib directory. Change all occurrences of AS400AUT to the automation session name used for this system (for example, SYS2AUT).
 - e. Change all occurrences of the suffix `_1` to `_2`.
 - f. Copy the new contents of AS400ADA.rul into your active rules file.

3. Update your Active Session Definition Set by creating three non-automated TN5250/TN3270 sessions.
 - a. Name the alert session, the automation session, and the manual session to the corresponding alert, automation, and manual session names used for this system (for example, SYS2GET, SYS2AUT, and SYS2MAN respectively).
 - b. Set the system name to the actual name of your system for each session created previously.
 - c. Choose an appropriate terminal type for each session. If you are unsure, choose 5292 or 3278_2, which are the defaults for TN5250 and TN3270, respectively.
 - d. Choose the "AS400" menu for each session.
4. Create and customize a sign-on/recovery script for the system's alert session by doing the following:
 - a. Copy AS400GET.scr to a new file that will be used for the system's alert session (for example, SYS2GET.scr).
 - b. On the KEY statement that contains the sign-on user ID and password, specify the correct user ID and password for the AS/400 system.
 - c. Save this file in the CA Automation Point/Site/Myfiles folder.
5. Create and customize a sign-on/recovery script for the system's automation session by doing the following:
 - a. Copy AS400AUT.scr to a new file that will be used for the system's automation session (for example, SYS2AUT.scr).
 - b. On the KEY statement that contains the signon user ID and password, specify the correct user ID and password for the AS/400 system.
 - c. Save this file in the CA Automation Point/Site/Myfiles folder.

Configuring AS/400 Manager When Connecting to Multiple Stand-alone Systems

To configure AS/400 Manager, run the configuration program AS4CFG, which updates the values in the AS400CFG.cmd file. Perform the following steps:

1. From Configuration Manager, choose Expert Interface, Automation, AS/400 Manager.
2. In the AS/400 Manager Configuration window, Click Insert.
3. In the AS/400 Session Description window, supply values for the following fields:

Session Name

The system's alert session name (for example, SYS2GET)

Session Type

ALR

4. Click Insert.
5. In the AS/400 System Description window, specify the following:

System Name

The name of the AS/400 system

Terminal Name

The name of the session (for example, SYS2GET)

Language Code

The appropriate language code. For more information, see Step 6 in the section [Configuring AS/400 Manager](#) (see page 406) in this chapter.

Script File

The name of the script file you created

Startup Status

RSTART

Tag Names

An alias for the AS/400 system

6. Click OK until you return to the AS/400 Manager Configuration window.
7. Repeat steps 2 through 6 for the automation session.

Note: This time specify the automation session name (for example, SYS2AUT), choose a session type of AUT, and specify the name of the script file you created.

8. Click Save to save the settings to the AS400CFG.cmd file.

Accessing the New Systems

After you have completed all the procedures for connecting AS/400 Manager with multiple stand-alone systems, restart CA Automation Point. To access the new alerts session or automation session, choose Jump To Window from the Window menu. If you defined a manual session, it displays as an icon on the CA Automation Point desktop.

Test Starting AS/400 Manager

Start CA Automation Point

CA Automation Point creates an icon on the desktop for the manual session AS400MAN and for the system status window AS400SSD. AS400SET.cmd reads the configuration variables and starts AS400CTL for the alerts session. The script should log on to the alert session and begin processing alerts. To access the alerts session (AS400GET) or automation session (AS400AUT), choose Jump To Window from the Window menu.

The system status plot should, within a few minutes, show the alert session in ACTIVE state.

The automation session remains in STRTUP state until a REXX program requests that some general automation activity be performed. This request can come from a time rule (for example, a rule that causes AS400CKJ.rex to execute), from a message rule (triggering execution of AS400MRP, for example), or from the SYSCMND application command.

Customize Desktop

Customize your desktop to suit your personal taste. You can open (and minimize) windows to the alert, automation, and manual sessions. You can also open or minimize other windows, such as the AP Log window.

After you have customized your windows, use the CA Automation Point Window Save Desktop option to maintain your window positions. These positions are "remembered" until you use the Window Save Desktop option again or until you make changes to the session definition using Configuration Manager.

You are now ready to begin writing your own rules and REXX automation procedures to automate AS/400 management.

Designing Automation for AS/400 Manager

This section describes how to automate events on your AS/400 System.

Defining Messages as Alerts in AS/400

Every OS/400 or application message that CA Automation Point handles must be defined on each AS/400 that CA Automation Point controls as an alert. For example, the OS/400 message ID CPA5305 is defined as an alert with the following OS/400 command:

```
CHGMSGD MSGID(CPA5305) MSGF(QSYS/QCPFMSG) ALROPT(*IMMED)
```

Executing CHGMSGD Commands from a CL Program

Group all your CHGMSGD commands as a CL program in a file and use the editor to add, change, or delete these commands. When necessary (after an OS/400 operating system update, for example), compile the CL program and run it. Using this method ensures easier OS/400 release changes when a new message description file is provided.

Writing CA Automation Point Rules

When writing CA Automation Point rules for AS/400, consider the following:

- Environmental variables usage
- Rules keywords usage
- Monitoring AS/400 system status
- Examples of automation routines

The following sections cover each of these topics.

Environmental Variables for AS/400 Manager

AS/400 Manager recognizes these environmental variables:

Variable

Description

&DATE

The current date in the form *MM/DD/YY*.

&DAY

The day: MON, TUE, WED, THU, FRI, SAT, or SUN.

&HOSTTIME

The time of the alert message from the Display Alert Detail screen, in the form *HH:MM:SS*.

&JOBNAME

The name of the AS/400 system the current message came from.

&JULDATE

The Julian date in the form *YYNNN*.

&MSG

The first 129 characters of the message. The message is the alert message ID, alert type, and alert description if the alert message ID is UNKNOWN or the alert message ID and alert message text.

&SYSNAME

The name of the AS/400 system the current message came from.

&TIME

The workstation time in the form *HH:MM:SS*.

&WORD1 up to &WORD85

Words from the alert text or application command.

For focal point machines, you can use the &SYSNAME variable to differentiate between systems. For example, the following rule executes only when the CPA5305 message is received from system S44A0060:

```
MSGID(CPA5305) WHEN(&SYSNAME EQ S44A0060) ...
```

Note: The value of &SESSION is always AS400RUL.

Rules Keywords Used for AS/400

You can use all CA Automation Point rules keywords for AS/400 *with these exceptions:*

OSCMD

Using this keyword can cause unpredictable results. Instead of OSCMD, use the supplied REXX program AS400MRP to issue responses to messages, or write REXX programs to issue commands to the AS/400 MAIN menu command line.

REPLY

Using this keyword can cause unpredictable results. Use REXX programs as detailed for OSCMD.

SCRIPT

Instead of using SCRIPT, invoke scripts from REXX programs to ensure the integrity of communications with managed AS/400 systems.

SESSCMD

For integrity reasons, use this keyword only in REXX programs.

Monitoring AS/400 System Status

The AS/400 Manager application maintains availability status variables for each managed AS/400 system. The status variables can contain any one of these values:

ACTIVE

Sign on to the AS/400 system is successful.

SUSPND

A SUSPEND command or an AXC0000 alert was issued for this AS/400 system. The AS/400 Manager application takes no action for an AS/400 system in SUSPND status. The suspension can be for a specified duration or it can be indefinite.

FAILED

AS/400 Manager has detected a communication error with an AS/400 system. Recovery is attempted.

ALARMS

AS/400 Manager is unable to recover from a FAILED status (the recovery retry attempts limit was reached).

CLOSED

AS/400 Manager takes no action for an AS/400 system. You can set this value as an initial status value in your configuration (AS4CFG.exe).

RSTART

The initial status after a CA Automation Point startup. It is also the status for a system leaving SUSPND state (time expired or RESUME command issued). You can set this status to have AS/400 Manager make an attempt to sign on to an AS/400 system by issuing the RESUME application command. The first time that activity on the session is required, a sign on is attempted. After this attempt:

- The status is set to ACTIVE if the sign on is successful.
- The status is set to FAILED if the sign on is not successful.

Important! Check the system status regularly. Also, process the warning messages issued by the REXX programs when a status change takes place. Initiate appropriate recovery or communications action if an ALARMS status occurs.

Sample Status Checking Rule

Use this rule as an example for designing your own status variable checking:

```
TIME(00:00), EVERY(5 MINUTES), WHEN(&AS400_AS400GET_ST_1 EQ ALARMS),
SET(&AS400_AS400GET_ST_1 = CLOSED),
```

This rule can also be used as a notification rule. CA Automation Point provides a number of notification methods, including sounding an audible alarm, writing a message in the Action Message window, voice notification, and paging.

Automation Routines

CA Automation Point rules are commonly used to:

- Reply automatically to an inquiry message
- Enter a command in response to a message
- Check job status

The following sections cover each of these topics.

Reply to an AS/400 Inquiry Message

The REXX program AS400MRP.cmd facilitates automatic replies to inquiry messages on the AS/400 operator queue. AS400MRP.cmd compares 75 characters of the alert message text to the inquiries on the operator queue. If the comparison criteria are met, it gives the response you specified to the most current message.

Use the following format to call AS400MRP from a rule:

```
AS400MRP sysname reply &MSG
```

sysname

The AS/400 system name.

reply

The text of the reply to the message.

&MSG

Passes the message text to AS400MRP.

Example 1:

The following example shows how to call AS400MRP from a CA Automation Point rule:

```
MSGID(CPA5305), WHEN(&JOBNAME EQ S44A0060),  
  REXX(AS400MRP &JOBNAME I &MSG)
```

For message CPA5305, an automatic response of I (ignore) is issued to the inquiry message in the operator message queue on AS/400 system S44A0060.

Example 2:

The following example shows how to automatically reply to this message:

```
CPA3394 Load form type 'x' device LASP01 writer Y. (H C G I R)
```

To supply an automatic reply of an I (ignore) to message CPA3394, specify a rule similar to this example:

```
MSGID(CPA3394), WHEN(&JOBNAME EQ S44A0060 AND &WORD7 EQ LASP01),
  REXX(AS400MRP &JOBNAME I &MSG),
  WTO(AP HAS RESPONDED I TO MSG CPA3394 ON SYSTEM &JOBNAME)
```

Example 3:

The following rule starts the REXX program AS400VRY.cmd with the AS/400 system name and workstation name supplied as parameters. The workstation name is contained in the eighth word (&WORD8 in the alert text).

```
MSGID(CPF1397), WHEN(&JOBNAME EQ S44A0060 AND &WORD8 EQ PC202S1),
  REXX(AS400VRY &JOBNAME &WORD8)
```

Note: The REXX program AS400VRY.cmd is coded to be general for multiple AS/400 systems. It receives the AS/400 system name and workstation to be varied on as parameters.

Example 4: Check Job Status

The following example REXX program verifies (every weekday at 16:00) that the job BACKUP1 has been submitted to the batch input queue:

```
*
* At 16:00 every Monday to Friday, check if job
* BACKUP1 is submitted
* to the batch input queue
*
TIME(16:00),
WHEN(&DAY(1:1) NE 'S' AND (&AS400_AS400AUT_ST_1 EQ 'ACTIVE' OR &AS400_AS400AUT_ST_1
EQ 'RSTART')),
  REXX(AS400CKJ S44A0060)
```

This rule starts the REXX program AS400CKJ.cmd, with the AS/400 system name (S44A0060) supplied as an argument.

Note: The REXX program AS400CKJ is coded to be general for multiple AS/400 systems. It receives the AS/400 system name as a parameter.

Writing Automation REXX Programs

The sample automation REXX programs AS400VRY and AS400CKJ can be used as a guide to write your own REXX programs. The sample programs follow these REXX programming guidelines:

- Call the supplied REXX program AS400JMP first to:
 - Find the automation session name of the current AS/400 system
 - Bind the automation session (this prevents another REXX program from interfering with the session)
 - Check the status of the AS/400 system automation session
 - Perform error recovery (if necessary)
 - Report errors in the AP Msg window
- If the RESULT variable returned from AS400JMP contains the word ERROR, communication with the AS/400 system you requested was disabled for some reason. In this situation, AS400JMP calls the UNBIND function, if necessary. Therefore, you need to issue an appropriate error message or take some other action appropriate to the failure of your intended action, and then terminate your REXX program.
- If the RESULT variable does *not* contain the word ERROR, it contains the name of the CA Automation Point session through which communications with the AS/400 system name you specified is being managed.
- Issue the commands to the AS/400 system automation session using the SESSCMD command processor with the CA Automation Point session name returned by AS400JMP.
- If the SESSCMD command processor returns a nonzero RC (return code), call the REXX program AS400SLR (session lost routine) to perform error actions.
- The last actions performed by your REXX program should be to return the AS/400 screen to the MAIN menu ready for the next communication, and to call the UNBIND function supplied by CA Automation Point to release control of the automation session from your REXX program.

Calling AS400JMP

Call the AS400JMP program as follows:

```
CALL AS400JMP sysname
```

sysname

The AS/400 system name or tag name.

On return, the special REXX variable RESULT contains the automation session name for the current AS/400 system name specified, and the session is bound. When your procedure is finished, it must call UNBIND. If AS400JMP is unable to locate the specified system, it returns the name ERROR instead of a session name. You do *not* need to call UNBIND in this case.

Calling AS400SLR

Call the AS400SLR program as follows:

```
AS400SLR sessname sysname
```

sessname

The CA Automation Point session name.

sysname

The AS/400 system name or tag name.

Operating AS/400 Manager

AS/400 Manager provides several AS/400 Manager application commands for your use. The commands are:

SUSPEND

Suspends control of a system

RESUME

Resumes control of a system

DSPSTAT

Displays system status in the AP Msg window

DSPCNFG

Displays configuration information in the AP Msg window

SYSCMND

Sends an OS/400 command to a system

Note: These application commands require that automation be active on the rules session. AS/400 Manager application commands do not function when CA Automation Point is globally paused or when automation is paused in the rules session.

SUSPEND and RESUME

The SUSPEND and RESUME functions enable you to suspend and later resume control of an AS/400 system (or all managed AS/400 systems). You can invoke the functions from:

- An alert initiated by an AS/400 system with message ID AXC0000
- Commands in the CA Automation Point command area
- Rules that invoke the AS400SPN REXX program directly

Sending Alert AXC0000 to Suspend Control

At times, you may need to re-IPL an unattended AS/400 system. To prevent CA Automation Point from generating error messages during the IPL, the AS/400 system should send an alert to CA Automation Point before shutdown and IPL occur, as shown:

```
AXC0000 nn comments
```

nn

The number of minutes before AS/400 Manager communications are restarted.

comments

Any desired comments; ignored by AS/400 Manager.

A rule directs AS/400 Manager to suspend control for the specified AS/400 system during the time period specified (excluding time spent in the alert queue).

SUSPEND Command

To manually suspend control of an AS/400 system, issue this command from a red command area in CA Automation Point:

```
SUSPEND sysname [nn] [mm]
```

sysname

Specifies an AS/400 system, tag, or group name to be suspended.

nn

(Optional.) Specifies the number of minutes before control of the AS/400 systems should be resumed.

mm

(Optional.) Specifies the number of minutes that the invocation of this routine has been delayed. This is normally specified only if called from the message rule for AXC0000 (the voluntary suspension message), and represents the number of minutes the AXC0000 message has been waiting in the alert queue. The suspend time is adjusted to compensate for this delay.

RESUME Command

To manually resume control of an AS/400 system, issue this command in the command area:

```
RESUME sysname
```

sysname

Specifies an AS/400 system, tag, or group name to be resumed.

Note: After an AS/400 system fails, use the RESUME command to reestablish control of that AS/400 system. The actual effect of the RESUME command is to set the status of the relevant system (or systems) to RSTART. This causes screen-position checking (and, where necessary, error recovery) procedures to be invoked the next time communication to the system is needed.

Displaying System and Configuration Data

The AS400SSD window graphically displays the current status of all AS/400 sessions. A set of 15-second time rules controls the updating of the plot.

The actual interval in which the display is updated also depends on the Evaluation Frequency field value, set in the Rules Settings dialog for your session definition set. For information on this field, see the help for the Rules Settings dialog.

Displaying System Status

Issue the DSPSTAT command to display a list of the system status variables for a single session or a group of sessions:

```
DSPSTAT sysname
```

sysname

Specifies the AS/400 system, tag, or group name.

Displaying Configuration Data

The DSPCNFG command displays the configuration data of AS/400 Manager:

```
DSPCNFG
```

DSPCNFG does not use any parameters.

Note: The output from this command can be quite lengthy, depending on the configuration.

Sending OS/400 Operator Line Commands

Do not use the automation sessions directly to enter OS/400 operator commands; doing so can cause errors in the affected sessions. Instead, use the command SYSCMND to send OS/400 operator line commands and return the response from line 24 of the AS/400 screen. (If you need to perform full-screen functions, use the manual session or access the system from outside CA Automation Point.)

The SYSCMND function sends an OS/400 operator line command to a single AS/400 system or group of systems. Issue the command as follows:

```
SYSCMND sysname command
```

sysname

The AS/400 system, tag, or group name.

command

A single OS/400 operator line command.

The command and its response appear in the Merged Msg and the AP messages windows.

Unbinding Sessions

If an automation program contains a syntax or logic error, it could end or loop without releasing a bound session. In such a case, other automation REXX programs attempting to bind to that same session could hang or fail. To recover from this situation, issue the UNBIND command to force a release of the bind:

```
UNBIND sessname
```

sessname

Specifies any alert or automation session name.

Note: Tag and group names are not recognized by the UNBIND command.

AS/400 Manual Session

Use the manual session to manage AS/400 systems independently from the automation sessions. To avoid conflicting with REXX programs, enter OS/400 full-screen commands from the manual session only. Immediately after you log in, map the keyboard by entering the following command:

```
CALL KBD3270
```

A suggested session name for the manual AS/400 session is AS400MAN.

Your installation may have defined multiple manual sessions for multiple systems or none at all.

Accessing the Manual Session

You can access the manual session using the CA Automation Point Remote Operations feature.

Technical Information

This section provides summaries and important technical information.

Summary of the REXX Programs

The following table lists the REXX programs supplied with AS/400 Manager, what the application is called by, and what the program does after it is called.

This REXX program...	Called by...	Does this...
AS400CKJ.cmd	User-written automation program	Illustrates how to implement a program to perform an automatic action
AS400CMD.cmd	Rule that implements SYSCMD application	Issues OS/400 operator line mode commands
AS400CTL.cmd	The configuration program AS400SET at startup, or restarted by a rule if AS400CTL fails	Captures alerts from AS/400 systems in RSTART status, converts them to CA Automation Point messages, and sends them to rules
AS400INI.cmd	AS400JMP with the session and system names	Runs a session sign on program
AS400JMP.cmd	AS400MRP and from user-written automation REXX programs	Checks the system status, and then binds to the session
AS400MRP.cmd	User-written rules	Performs message reply to AS/400
AS400REC.cmd	Rules invoked when a system is in FAILED status	Performs recovery on failed communications with system
AS400SET.cmd	IREXX at CA Automation Point startup	Calls AS400CFG.cmd to initialize the AS/400 configuration; starts AS400CTL.CMD for each alert session defined
AS400SLR.cmd	User-written automation programs and from AS400MRP	Invokes a session error program
AS400SPN.cmd	Rules	Implements the SUSPEND RESUME application commands
AS400SSD.cmd	Time rules when the system status has changed	Plots the system status in the AS400SSD plot window for all defined sessions
AS400STA.cmd	Rule that implements the DSPSTAT application command	Displays AS/400 system status information in the AP Msg window

This REXX program...	Called by...	Does this...
AS400SUS.cmd	Time rules	Manages suspend time for systems in SUSPND status and decrements the suspension timer value. When the suspend interval has elapsed, it sets the system status to RSTART.
AS400SYS.cmd	Rule that implements the DSPCNFG application command	Displays a list of configuration definition status variables in the AP Msg window
AS400VRY.cmd	User-written automation program	Illustrates how to implement a program to perform an automatic action

REXX Program AS400CTL

AS400CTL.CMD is called by the configuration program AS400SET.cmd at CA Automation Point startup or restarted by a rule with the following parameters:

AS400CTL *sessname* WAIT *nn*

sessname

The session name.

nn

The number of seconds to wait before processing starts.

AS400CTL.cmd controls an alert session, parses the alert messages, and sends the messages to CA Automation Point rules for processing.

How AS400CTL.cmd Processes Alerts

The AS400CTL program has the following key functions:

1. If the relevant AS/400 system status variable is set to RSTART, this program calls the appropriate initialization script as defined by CA Automation Point rules to:
 - Sign on to the specified AS/400 systems
 - Configure the AS/400 keyboard for 3270 use on each system
 - Issue the OS/400 WRKALR command on each system
2. If the AS/400 system status is ACTIVE, AS400CTL checks for alerts at regular intervals by pressing the PF10 (refresh) key in the WRKALR screen.

3. Checks the response from all entered commands and function keys to see whether the session has failed. If it has failed, tries to recover the session by calling AS400REC and initialization scripts before setting the relevant AS/400 alert session system status variable to FAILED if the recovery was unsuccessful.
4. Handles each alert, working from the bottom of the WRKALR screen upward (that is, oldest alert first) by:
 - Typing 8 before the alert on the WRKALR screen to display the alert details screens
 - Extracting message ID, message description, severity, and system ID from the alert details, usually paging forward using the PF8 key at least once
 - Pressing Enter to return to the WRKALR main screen
 - Sending the previously mentioned information to the CA Automation Point rules processor using the MSG command processor
 - Typing 4 to request deletion of the alert
 - Pressing Enter to confirm deletion of the alert

Screen Examples

The following sample screens show a dialog initiated by AS400CTL:

```

Work with Alerts                                     S44A0060

Type options, press Enter.
 2=Change  4=Delete  5=Display recommended actions 6=Print details
 8=Display alert detail 9=Work with problem

Resource
Opt  Name      Type  Date   Time  Alert Description: Probable Cause
 8   S44A0060  CP    03-30  10:45  Undetermined error: Undetermined

Bottom
F3=Exit   F10=Show new alerts   F11=Display user/group  F12=Cancel
F13=Change attributes  F20=Right  F21=Automatic refresh  F24=More keys

Display Alert Detail                               S44A0060
    
```

```

-----Resource Hierarchy-----
Resource Name      Resource Type
S44A0060           CP

Logged date/time . . . . . : 93-03-30 10:45:51
Problem date/time . . . . . : 93-03-30 10:45:50
User assigned . . . . . :
Group assigned . . . . . :
Alert type . . . . . : Unknown
Alert description . . . . . : Undetermined error
Probable cause . . . . . : Undetermined
Qualifiers . . . . . : AS/400 Message code CPA5305
                       AS/400 Message severity 99

More...

Press Enter to continue.

F3=Exit   F11=Display detail menu  F12=Cancel  F18=Display actions
    
```

```
Work with Alerts S44A0060

Type options, press Enter.
 2=Change  4=Delete  5=Display recommended actions  6=Print details
 8=Display alert detail  9=Work with problem

Resource
Opt  Name      Type  Date  Time  Alert Description: Probable Cause
 4   S44A0060  CP    03-30 10:45 Undetermined error: Undetermined

Bottom
F3=Exit  F10=Show new alerts  F11=Display user/group  F12=Cancel
F13=Change attributes  F20=Right  F21=Automatic refresh  F24=More keys
```

```
Display Alert Detail S44A0060

-----Resource Hierarchy-----
Resource Name      Resource Type
S44A0060           CP

Text message:
Sender ID . . . . . : Control program
Message . . . . . : Record not added. Member MEMBERNAME is null.
                  (C I)

Unique alert identifier:
Product ID . . . . . : 9406
Alert ID number . . . : 331A A4A1

Bottom
Press Enter to continue.

F3=Exit  F11=Display detail menu  F12=Cancel  F18=Display actions
```

```

Confirm Delete of Alerts                                     S44A0060

Press Enter to confirm your choices for 4=Delete.
Press F12 to return to change your choices.

Resource
Opt  Name      Type  Date  Time  Alert Description: Probable Cause
 4   S44A0060  CP    03-30 10:45 Undetermined error: Undetermined

F11=Display user/group  F12=Cancel  F20=Right

Bottom
    
```

```

Work with Alerts                                           S44A0060

Type options, press Enter.

2=Change  4=Delete  5=Display recommended actions  6=Print details
8=Display alert detail  9=Work with problem

Resource
Opt  Name      Type  Date  Time  Alert Description: Probable Cause
(No alert entries matching selection criteria)

F3=Exit  F10=Show new alerts  F11=Display user/group  F12=Cancel
F13=Change attributes  F20=Right  F21=Automatic refresh  F24=More keys

Bottom
    
```

Automation Session Programs

The automation sessions perform automatic actions on AS/400 systems and are CA Automation Point sessions defined as 3270 terminals.

Automatic Actions

Automatic actions must be performed as REXX programs. The sample programs AS400VRY and AS400CKJ are provided to illustrate how such programs should be written and structured.

In user-written automatic actions, the first step *must* be to call the supplied AS400JMP REXX program to perform AS/400 system status checking.

REXX Programs

The following REXX programs run in automation sessions:

AS400INI REXX

Invokes sign-on procedures to the AS/400 systems automatically when the AS400JMP REXX program finds that a sign-on procedure needs to be performed. You would not normally call AS400INI directly from your own REXX program.

AS400SLR REXX

Should be called from user-written REXX programs when a communications failure results in a nonzero return code from SESSCMDs.

AS400MRPREXX

Simplifies creating automatic actions to answer inquiry messages in the operator message queue.

AS400CMD REXX

Handles the sending of an OS/400 operator line command to one AS/400 system or to a group of AS/400 systems, which is enabled by the operator command facility.

REXX Program AS400INI

AS400INI is called from AS400JMP using either of the following parameters:

```
AS400INI sessname WAIT nn  
AS400INI sessname sysname
```

sessname

The session name.

nn

The number of seconds to wait before processing starts.

sysname

The system name used when AS400INI is called from AS400JMP.

Key AS400INI Functions

The AS400INI program performs these tasks:

1. If called with the WAIT parameter:
 - Waits for the specified number of seconds
 - Binds to the specified session
2. Finds automation session system status variables for each system in the session that is in RSTART mode.
 - This program checks to see if the signon menu is shown with the correct terminal ID or AS/400 system name. If either of these cases is true, the specified initialization script as defined by AS400SET is called to perform signon of the AS/400 system.
 - If the main menu is then shown for the relevant AS/400 system, the relevant automation session system status variable is set to ACTIVE.
 - If the main menu is *not* shown for the relevant AS/400 system, error recovery is performed by:
 - Calling the AS400RES script
 - Calling the initialization script
 - If the main menu is still not shown for the relevant AS/400 system:
 - An error message is written to the AP Msg and Action Msg windows.
 - The automation session system status variable is set to FAILED.
3. If called at CA Automation Point startup with the WAIT parameter:
 - The current session is unbound (UNBIND).
 - The program is terminated without any returned value.
4. If called from AS400JMP, the program is terminated with the current status value returned to AS400JMP.

Key AS400MRP Functions

AS400MRP uses the parameters supplied to do the following tasks:

1. Call AS400JMP to check the AS/400 system status, obtain the session name of the relevant automation session, and bind it. If the returned REXX variable RESULT contains the word ERROR:

- Error messages are written to the AP Msg and Action Msg windows
- Processing of AS400MRP is terminated

Otherwise, the REXX variable RESULT contains the automation session name for the relevant AS/400 system.

2. Issue the following OS/400 command shown using SESSCMD:

```
dspmsg msgq(*sysopr) msgtype(*inq) astlvl(*intermed) start(*last)
```

3. AS400MRP compares 75 characters of the message text supplied in the parameters with the inquiry message texts on the operator queue as follows:

- The message search is from the top of the screen to the bottom of the screen.
- PF7 (page backward) is performed up to five times (if necessary) until the message is found.
- If the message text is on the bottom of one screen and the input field is on the top on the next screen, a PF8 (page forward) is done.

4. The reply given in the parameters passed to AS400MRP is used as input in the unprotected field following the inquiry message.
5. PF3 (end key) is issued to exit the operator message queue.
6. The automation session is unbound.

If any SESSCMD results in a nonzero RC (return code), the REXX program AS400SLR is called to perform error notification actions.

AS400MRP Screen Examples

The following screen examples show a dialog initiated by AS400MRP.

```

MAIN                               AS/400 Main Menu
                                     System:  S44A0060

Select one of the following:

    1. User tasks
    2. Office tasks
    3. General system tasks
    4. Files, libraries, and folders
    5. Programming
    6. Communications
    7. Define or change the system
    8. Problem handling
    9. Display a menu
    90. Information Assistant options

    90. Sign off

Selection or command
==> dspmsg msgq(*sysopr) msgtype(*inq) astlvl(*intermed) start(*last)

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F13=User support
F23=Set initial menu
    
```

```

                                Display Messages
                                     System:  S44A0060
Queue . . . . . : QSYSOPR          Program . . . . . : *DSPMSG
Library . . . . : QSYS             Library . . . . . :
Severity . . . . : 40              Delivery . . . . . : *HOLD

Type reply (if required), press Enter.
Reply . . . . _____
Load form type '*STD' device PRT02 writer PRT02. (H C G I R)
Reply . . . . _____
Record not added. Member MEMBERNAME is full. (C I)
Reply . . . . I
Load form type '*STD' device PRT02 writer PRT02. (H C G I R)
Reply . . . . I
Record not added. Member MEMBERNAME is full. (C I)
Reply . . . . i
Load form type '*STD' device PRT02 writer PRT02. (H C G I R)
Reply . . . . _____
Record not added. Member MEMBERNAME is full. (C I)
Reply . . . . _____

                                     Bottom
F3=Exit          F11=Remove a message      F12=Cancel
F13=Remove all   F16=Remove all except unanswered  F24=More keys
    
```

AS/400 Default Logon Screen

A sample of the IBM-supplied AS/400 main logon screen is shown next.

```
Sign On

                                System . . . . . : S44A0060
                                Subsystem. . . . . : QINTER
                                Display. . . . . : TERM01

User . . . . . _____
Program/procedure. . . . . _____
Menu . . . . . _____
Current Library. . . . . _____
```

Note: Your System and Display fields will differ from those shown in the sample.

Chapter 16: Using the Plot Feature

This chapter discusses the CA Automation Point Plot feature.

Understanding the Plot Feature

The CA Automation Point Plot feature allows you to plot information about system events collected by rules and automation REXX programs into line or bar graphs. For example, you can plot graphs showing the number of messages having particular prefixes (such as ATM or HASP) that system resources have issued over a specified time period.

Plotting a Graph

To plot a graph showing system information, first set up a CA Automation Point window to display the graph. Choose PLOT as your window type in your function window definition.

After setting up a window from which to display the graph, you can plot the graph in one of these ways:

- Write a REXX program containing all of the PLOT command statements necessary for creating the graph. This method is the usual way to plot a graph.
- Use the Plot Feature with REXX.
- Create a menu to issue PLOT commands.
- Issue a series of PLOT commands from any of the CA Automation Point command entry points.

Summary of Steps for Plotting a Graph

Use PLOT commands to perform the following operations:

- Define the graph's basic appearance
- Define lines or bars to represent system information
- Insert data into the graph or change the data in it
- Delete obsolete data from the graph
- Draw the graph

Understanding the Elements of a Graph

The graphs that you draw with the Plot feature contain these elements:

- An optional title
- An X axis (horizontal) and a Y axis (vertical)
- Optional labels for the X and Y axes
- Lines or bars representing system information
- Optional tick marks quantifying the information indicated with lines or bars

A sample graph displaying the JES SPOOL usage for several systems over a 24-hour period is shown next. The sample application containing the PLOT statements necessary for creating the graph is available in the sample directory; see the SPLPLOTX.txt file for more information.



Most of the examples in the remainder of this chapter show actual PLOT commands for setting up or modifying the data in the SPLPLOT graph.

Using the Plot Feature with REXX

If you intend to write REXX programs containing PLOT commands, remember the following:

- PLOT command statements accept multiple operands.

You can draw a graph with only a few PLOT commands. The REXX language allows you to concatenate multiple lines whenever a line ends with a comma; many of the examples in this chapter show concatenated lines.

Keep the following points in mind:

- When concatenating PLOT command operands, do *not* exceed REXX's 256-character maximum line length.
- When assigning values to PLOT command operands, do *not* insert a space on either side of the "(" character.

- Have your REXX program check the return code from each PLOT call to simplify the debugging process.

The Plot feature assigns defaults freely for missing information. Most usage or syntax errors in a PLOT command statement simply generate return codes rather than error messages.

- Test complex plots outside of the AXCREXX window.

You can do so by manually executing them from a Windows command line while CA Automation Point is running. Enter the AXCREXX REXXNAME. This testing method allows you to scroll through REXX trace information generated by the REXX program.

Note: The example PLOT command statements in the remainder of this chapter are enclosed in double quotation marks to show how they would appear in a REXX program. (Because the sample plot file SPLPLOT.cmd uses parameter substitution, there are more quotation marks within its PLOT statements, following standard REXX syntax.)

Designing a Basic Graph

To define the basic setup for a graph, issue the PLOT DEFINE GRAPH command using this syntax:

```
PLOT plotname DEFINE GRAPH options
```

plotname

The name of the graph that you are defining.

options

You can specify any or all of the following options for *options*:

```
AXIS(XAXIS|YAXIS [,COLOR(WHITE|color)]  
[,TYPE(NUMERIC|TIME)])  
BACKGROUND(LABELAREA|PLOTAREA,COLOR(BLUE|color))  
DRAW(ON|OFF)  
LABEL(XAXIS|YAXIS|TITLE,'label' [,COLOR(color)] [,LINE(n)])  
SCALE(XAXIS|YAXIS, start, stop [,step])  
TICK(XAXIS|YAXIS, 'string', number)
```

You can specify multiple tick marks.

```
TICK(XAXIS|YAXIS, 'string', number) WINDOW(windowname)
```

The following section describes the PLOT DEFINE GRAPH command's options.

PLOT DEFINE GRAPH Command Options

The following sections describe the PLOT DEFINE GRAPH command's options:

AXIS Option

This option defines the characteristics of the axes displayed, including the units used to measure the data in the graph.

Use the following syntax for the AXIS option:

```
AXIS(XAXIS|YAXIS [,COLOR(WHITE|color)] [,TYPE(NUMERIC|TIME)])
```

XAXIS or YAXIS

Specifies the axis (X or Y) for which you are defining characteristics.

COLOR

Sets the color for the axis lines and tick labels. For a list of colors from which you can choose, refer to the information under the LABEL option of this table.

TYPE

Defines the units that measure data plotted in the graph. Specify TYPE(NUMERIC) to measure the data in positive integers (such as 1 or 32). Specify TYPE(TIME) to measure data in terms of time on a 24-hour clock (for example, 10:00, 0:31)

Default: COLOR(WHITE), TYPE(NUMERIC)

Example:

The following command defines the X axis of the SPLPLOT graph as a time axis to appear in turquoise:

```
"PLOT SPLPLOT DEFINE GRAPH",  
  "AXIS(XAXIS,COLOR(TURQUOISE),TYPE(TIME))"
```

BACKGROUND Option

This option sets the background fill color for areas of the screen. The PLOTAREA is the rectangle defined by the axes of the graph. The LABELAREA is the rest of the screen.

Use the following syntax for the BACKGROUND option:

```
BACKGROUND(LABELAREA|PLOTAREA,COLOR(BLUE|color))
```

For a list of colors from which you can choose, see the LABEL Option in this table. Using BLACK as the background color allows the greatest number of usable foreground colors.

Default: COLOR(BLUE)

Example:

The following command sets the background fill color of both the plot and label areas to black:

```
"PLOT SPLPLOT DEFINE GRAPH",  
  "BACKGROUND(PLOTAREA,COLOR(BLACK))",  
  "BACKGROUND(LABELAREA,COLOR(BLACK))"
```

The following command sets the background fill color of both the plot and label areas to black:

```
"PLOT SPLPLOT DEFINE GRAPH",  
  "BACKGROUND(PLOTAREA,COLOR(BLACK))",  
  "BACKGROUND(LABELAREA,COLOR(BLACK))"
```

DRAW Option

This option controls window updating. If you specify DRAW(ON), CA Automation Point updates the window every time another PLOT command executes. If you are using a REXX program or other application to plot a

Use the following syntax for the DRAW option:

```
DRAW(ON|OFF)
```

If you specify DRAW(OFF), CA Automation Point redraws the graph only when you issue the following command:

```
"PLOT plotname DRAW"
```

Default: DRAW(ON)

Example:

The following command prevents the SPLPLOT graph from being displayed while building or redrawing occurs:

```
"PLOT SPLPLOT DEFINE GRAPH DRAW(OFF) "
```

CA Automation Point redraws the graph using the current data when requested to do so by the operating system's graphic user interface, even if you specify DRAW(OFF). The requests occur when you resize the Plot window or when you move an overlapping window.

LABEL Option

This option specifies a title for the graph or labels for the X and Y axes. Indicates whether you are specifying text for one of the graph axes or for the graph title.

Use the following syntax for the LABEL option:

```
LABEL(XAXIS|YAXIS|TITLE, 'label' [,COLOR(color)] [,LINE(n)])
```

TITLE

The title of the graph displays at the top center of the window for this graph. You can specify up to three lines of label text for the X axis, and up to seven lines for the Y axis.

XAXIS

The label displays centered on the X axis.

YAXIS

The label displays centered on the Y axis.

label

The text of the label, which must be enclosed in single quotation marks. The label for the graph or for the X axis can contain as many as 80 characters. The label for the Y axis can contain as many as 32 characters.

COLOR

The color of the label. Valid *color* values are:

BLUE	BRIGHT BLUE
TURQUOISE	BRIGHT TURQUOISE
YELLOWGREEN	BRIGHT GREEN
RED	BRIGHT RED
BRIGHT PURPLE	PURPLE
BROWN	BRIGHT WHITE
WHITE (default)	
GRAY	

LINE

The number of the line in the label text. A label can contain up to three lines of text. Your PLOT command must contain one LABEL clause for each line. The LINE(*n*) operand specifies whether the line being defined is line 2 or line 3 of the label text. Because the default for *n* is 1, you need to include the LINE(*n*) operand only in the second or third LABEL clauses of a PLOT command. For example, if you are defining a two-line label, the LABEL clause defining the second line is LINE(2).

Default: If you omit the LABEL option completely, the default is no labels. If a LABEL clause specifies no color, the default label color is white. If you omit the LINE(*n*) operand, CA Automation Point assumes that the label contains only one line of text.

Examples:

These commands create the title and one of the Y-axis labels:

```
"PLOT SPLPLOT DEFINE GRAPH",  
  "LABEL(TITLE,'JES Spool Utilization',COLOR(TURQUOISE))"
```

```
"PLOT SPLPLOT DEFINE GRAPH",  
  "LABEL(YAXIS,'S008',COLOR(RED),LINE(2))"
```

Note: If you define a graph with many lines using different colors, specify a Y-axis label for each line in the color of the line to make the graph easier to read.

Usage Information:

Use labels sparingly. If you use too many labels, the size of the plot area decreases to allow more space for the label areas. Avoid long YAXIS labels and multiline XAXIS and TITLE labels as much as possible.

If you have a dedicated Plot window (used only for displaying one plotted graph), you can specify a title for the window in a CA Automation Point session definition set (rather than specifying the LABEL operand). Doing so produces a more readable title and conserves space within the plot area.

SCALE Option

This option imposes a range of values, indicated by tick marks, on the X axis or the Y axis.

Use the following syntax for the SCALE option:

```
SCALE(XAXIS|YAXIS, start, stop [,step])
```

XAXIS or YAXIS

Specifies the axis (X or Y) for which you are defining a scale.

start

Defines the lowest tick value for this axis.

stop

Defines the highest tick value for this axis.

step

Sets the interval between tick marks that CA Automation Point displays for this axis.

Note: The tick values that you specify with the SCALE option must be consistent with the type of tick value that you specified with the TYPE operand in the AXIS clause described under step one of designing a basic graph. For example, you cannot specify numeric tick values for an axis that you have defined as containing time data.

Default: The *start* value is the first *step* value lower than the smallest data value on the axis, and the *stop* value is the first *step* value greater than the largest data value on the axis.

If you do not specify a *step* value, CA Automation Point computes the difference between the *start* and *stop* values and calculates how many tick marks are needed to divide the range of values into even increments. CA Automation Point uses approximately 10 tick marks to divide a range (unless you specify otherwise).

Example:

This command sets the X-axis scale to a range of 24 hours with one tick mark every four hours, and the Y-axis scale to a range of 0 to 100 percent with one tick mark every 10 percent:

```
"PLOT SPLPLOT DEFINE GRAPH",  
  "SCALE(XAXIS,0:00,23:59,04:00)",  
  "SCALE(YAXIS,0,100,10)"
```

TICK Option

This option specifies a translation table between string-label tick marks and numerical values for an axis. Because the Plot feature accepts data and assigns tick marks to it internally, most PLOT commands do not require the TICK operand; however, you may want to use the TICK operand if you are plotting encoded data.

Use the following syntax for the TICK option:

```
TICK(XAXIS|YAXIS, 'string', number)
```

XAXIS or YAXIS

Indicates whether you are assigning a string-label tick mark to a numeric value located on the X-axis or the Y-axis.

string

Specifies the tick label for the specified axis. CA Automation Point displays tick labels only for the values specified with TICK clauses. (CA Automation Point lets you include multiple TICK clauses in a single PLOT DEFINE GRAPH command, so that you can specify all of your tick values with one command.)

number

Specifies the number of tick labels for the specified axis. CA Automation Point draws as many tick labels as space allows. If a label would overlap the tick label preceding it, CA Automation Point omits the overlapping label.

Default: Tick labels display automatically at SCALE *step* values (outlined under the SCALE option).

Example:

Suppose that you are plotting data points having these meanings:

1 = Slowest

2 = Slow

3 = Moderate

4 = Fast

5 = Fastest

You can assign meaningful aliases to the numeric tick marks by specifying the TICK option in the following PLOT command:

```
"PLOT FLOWRATE DEFINE GRAPH
TICK(YAXIS, 'Slowest', 1) ",
  "TICK(YAXIS, 'Slow', 2) ",
  "TICK(YAXIS, 'Moderate', 3) ",
  "TICK(YAXIS, 'Fast', 4) ",
  "TICK(YAXIS, 'Fastest', 5) "
```

Note: If you specify the TICK option, be sure to define string-label ticks for **every** data value that you intend to graph on the specified axis. If you insert data points with Y-axis values outside of the 1-5 range in the preceding example, the Plot feature labels those points with standard numeric labels.

WINDOW Option

This option specifies the Plot window in which your plotted graph displays.

Use the following syntax for the WINDOW option:

```
WINDOW(windowname)
```

Default: The default for *windowname* is the name that you specify for *plotname*.

Example:

Suppose that in your session definition set, you define SPLPLOT1 as the window to display graphs. To display the SPLPLOT graph in that window, issue this command:

```
"PLOT SPLPLOT DEFINE GRAPH WINDOW(SPLPLOT1)"
```

Note: If you do not specify the WINDOW option, and the name of your graph (*plotname*) does not match the name of an existing Plot window (*windowname*), CA Automation Point plots the graph but does not display it. In such a case, you can display the graph by:

- Issuing the PLOT DRAW command, or
- Including the DRAW(ON) clause (see the section entitled [DRAW Option](#) (see page 446)) in the PLOT DEFINE GRAPH command.

Defining Graph Lines or Bars

The PLOT DEFINE LINE and PLOT DEFINE BAR commands determine whether the graph uses lines or bars to represent data and assigns names to each line or bar. The commands use the following syntax:

```
PLOT plotname DEFINE LINE linename [COLOR(WHITE|color)]
```

```
PLOT plotname DEFINE BAR barname [COLOR(WHITE|color)]
```

plotname

Specifies the name of the graph for which you are designing lines or bars.

linename

Specifies the name of this line. You must name each line or bar because other graphs can use these names to identify a line or bar to insert or delete.

A graph can display up to eight different lines to represent data, and can contain one bar and multiple lines. For example, the SPLPLOT graph uses a single bar to show the current time.

barname

Specifies the name of this bar. You can define only one bar per graph.

COLOR

Specifies the color used to display lines or bars. You can choose any of the colors listed. For information on colors, see the section [LABEL Option](#) (see page 446). If you define multiple lines, each line can be in a different color.

Defaults:

- COLOR(WHITE)
- CA Automation Point defines no default lines or bars. However, the PLOT INSERT DATA command defines a default line if you have not previously issued any PLOT DEFINE LINE or PLOT DEFINE BAR commands.

Example:

These commands define a red line named S008 and a turquoise bar named NOW (which displays the current time):

```
"PLOT SPLPLOT DEFINE LINE S008 COLOR(RED)"  
"PLOT SPLPLOT DEFINE BAR NOW COLOR(TURQUOISE)"
```

Inserting Data into a Graph

Issue the PLOT INSERT DATA command to add or change data for a line or bar within a graph. Use this syntax:

```
PLOT plotname INSERT DATA linename|barname XY(xvalue,yvalue)
```

plotname

The name of the graph in which you are inserting data.

linename

The name of the line that displays new data. If you have not already defined the name, CA Automation Point defines it as a white line.

barname

The name of the bar that displays the new data. If you have not already defined the name, CA Automation Point defines it as a white bar.

XY

Specifies the point on the graph where CA Automation Point inserts the data. The value for the *xvalue* or *yvalue* variables in a XY(*xvalue,yvalue*) clause can be a string or a positive integer. If you specify a string, CA Automation Point translates that string into a number by matching it against names specified with the TICK operand on the PLOT DEFINE GRAPH command. If you have not specified tick names, CA Automation Point calculates implicit tick positions for the inserted data.

You can specify the XY operand repeatedly in a single PLOT INSERT DATA statement, as long as the length of the entire statement does not exceed 256 characters.

Defaults: None

Examples:

- The following command inserts two numeric data points into the S008 line of the SPLPLOT graph:

```
"PLOT SPLPLOT INSERT DATA S008 XY(0:00,72) XY(0:15,70)"
```
- The following command inserts an IMSP character string into the FLOWBAR bar of a graph named FLOWRATE:

```
"PLOT FLOWRATE INSERT DATA FLOWBAR XY(IMSP,5)"
```

Deleting Data from a Graph

Issue the PLOT DELETE command to remove data from a graph. You can issue any of the following versions of the command:

```
PLOT plotname DELETE DATA linename|barname  
PLOT plotname DELETE DATA [linename or barname]RANGE([XAXIS or YAXIS,min,max)  
PLOT plotname DELETE LINE linename  
PLOT plotname DELETE BAR barname  
PLOT plotname DELETE GRAPH
```

PLOT DELETE DATA command deletes all of the data for the named line or bar. However, this line or bar and its non-data characteristics (such as color) are still defined to CA Automation Point. If you insert new data into the named line or bar later, CA Automation Point displays it using the characteristics of the old line or bar.

PLOT DELETE DATA RANGE command deletes a range of data along the named line or bar.

The PLOT DELETE DATA LINE and PLOT DELETE DATA BAR commands delete both the definition for and the data in the named line or bar. You can reuse the names of deleted lines or bars later to define new lines or bars.

The PLOT DELETE GRAPH command deletes the graph and all information associated with it. The command also clears the display window. After deleting a graph, you can reuse its name later for a new graph.

plotname

The name of the graph that you want to delete or that contains the data that you want to delete.

linename

In a PLOT DELETE LINE command, the name of the line to delete or to remove data from.

barname

In a PLOT DELETE BAR command, the name of the bar to delete or to remove data from.

RANGE

In a PLOT DELETE DATA command, the range of data that you want to delete. You need to specify which axis to delete data from, and the upper and lower limits for the data that you want to delete.

You define the RANGE using these values:

XAXIS or YAXIS

The axis to delete data from.

min, max

The lower and upper limits, respectively, of the data to be deleted. The *min* value must always be smaller than the *max* value. The values must also be consistent with the type of data defined for the named axis (through the PLOT DEFINE GRAPH command). For example, you cannot specify *min* and *max* time values of 0:00 and 10:00 for an axis that you defined as having numeric data.

Examples:

The following command deletes data from the NOW bar in the SPLPLOT graph:

```
"PLOT SPLPLOT DELETE DATA NOW"
```

This command deletes the S008 line:

```
"PLOT SPLPLOT DELETE LINE S008"
```

This command deletes the SPLPLOT graph and produces a blank window:

```
"PLOT SPLPLOT DELETE GRAPH"
```

Redrawing the Graph

The PLOT DRAW command redraws the graph *plotname* in the display window using the current data. Issuing the PLOT DRAW command does not affect the screen updating controlled by the DRAW operand of the PLOT DEFINE GRAPH command. When you issue the PLOT DRAW command, CA Automation Point refreshes the display even if you specified the DRAW(OFF) option (see the section entitled [DRAW Option](#) (see page 446)) when you defined the graph.

Use this syntax:

```
PLOT plotname DRAW
```

plotname

Represents the name of the graph that you want to redraw.

Default: None

Example:

This command refreshes the display of the SPLPLOT graph:

```
"PLOT SPLPLOT DRAW"
```

Sample PLOT Statements in a REXX Program

The following listing shows typical PLOT command statements that you could place in a REXX program to create a graph called MYPLOT. To better illustrate the operations necessary for plotting a graph, the code fragment shown does not contain miscellaneous REXX processing statements and does not check PLOT command return codes.

```
/* This statement erases any existing graph data*/  
"PLOT MYPLOT DELETE GRAPH"  
  
/* Define basic graph appearance */  
"PLOT MYPLOT DEFINE GRAPH WINDOW(PLOTWIN)"  
  
"PLOT MYPLOT DEFINE GRAPH",  
  "AXIS(XAXIS, COLOR(TURQUOISE), TYPE(TIME))",  
  "AXIS(YAXIS, COLOR(TURQUOISE), TYPE(NUMERIC))"
```

```
"PLOT MYPLOT DEFINE GRAPH BACKGROUND(PLOTAREA,COLOR(BLACK))",
  "BACKGROUND(LABELAREA,COLOR(BLACK))"

"PLOT MYPLOT DEFINE GRAPH",
  "LABEL(TITLE, 'Test of Graph Facility',COLOR(RED))"
"PLOT MYPLOT DEFINE GRAPH",
  "LABEL(YAXIS, 'STSO',COLOR(RED))",
  "LABEL(YAXIS, 'STSO-LINE 2',COLOR(BRIGHT PURPLE),LINE(2))"
"PLOT MYPLOT DEFINE GRAPH",
  "LABEL(XAXIS, 'XAXIS',COLOR(BRIGHT GREEN))",
  "LABEL(XAXIS, 'XAXIS-line2',COLOR(yellow),LINE(2))"

"PLOT MYPLOT DEFINE GRAPH",
  "SCALE(XAXIS,0:00,23:59,04:00)",
  "SCALE(YAXIS,0,10,1)"

/* Define lines or bars to represent information */
"PLOT MYPLOT DEFINE LINE LINE1 COLOR(YELLOW)"
"PLOT MYPLOT DEFINE LINE LINE2 COLOR(RED)"
"PLOT MYPLOT DEFINE LINE LINE3 COLOR(PURPLE)"
"PLOT MYPLOT DEFINE BAR BAR1 COLOR(BRIGHT GREEN)"

/* Insert data into graph */
"PLOT MYPLOT INSERT DATA LINE1 XY(00:00,1) XY(04:00,2) XY(08:00,3),
  XY(12:00,4) XY(16:00,5) XY(20:00,4) XY(23:59,3)"
"PLOT MYPLOT INSERT DATA LINE2 XY(00:00,5) XY(04:00,4) XY(08:00,3),
  XY(12:00,2) XY(16:00,3) XY(20:00,4) XY(23:59,3)"
"PLOT MYPLOT INSERT DATA LINE3 XY(00:00,5) XY(04:00,4) XY(08:00,5),
  XY(12:00,4) XY(16:00,3) XY(20:00,2) XY(23:59,1)"
"PLOT MYPLOT INSERT DATA BAR1 XY(00:00,1) XY(04:00,1) XY(08:00,1),
  XY(12:00,4) XY(16:00,2) XY(20:00,2) XY(23:59,2)"

/* Draw the graph */
"PLOT MYPLOT DRAW"
/* Delete old data from the midnight shift */
"PLOT MYPLOT DELETE DATA BAR1 RANGE(XAXIS,0:00,08:00)"
/* Redraw the graph */
"PLOT MYPLOT DRAW"
```


Appendix A: Customizing Special CA Automation Point Files

This appendix describes how you can customize the following types of files:

- Keyboard parameter
- Scan code
- Script

Customizing Keyboard Parameter Files

A CA Automation Point keyboard parameter file defines keyboard operations for both the mainframe and workstation environments.

Some keys that apply in the mainframe environment, such as PF15, have no meaning in a workstation environment. Likewise, some keys that apply in the workstation environment have no meaning in the mainframe environment. Because CA Automation Point windows communicate with both the workstation operating system (through CA Automation Point function windows) and the mainframe operating system (through terminal emulator windows), using the same physical keyboard, you can customize a keyboard parameter file to define the keyboard operations that apply in each environment for your particular site or use the CA Automation Point default keyboard parameter file. The enhanced 101-key keyboard is the default keyboard.

This section describes how you can customize keyboard parameter files.

Keyboard Mapping File

CA Automation Point includes a default keyboard parameter file, `key_101.key`, for a standard workstation keyboard. You can customize this file later, or use it as a model to create a new one.

Note: CA also provides keyboard parameter files for some international keyboards. The files reside in either the `%AP_SITE%\MyFiles` directory or the `installdir\Distrib` directory and have a `.key` extension.

Looking at the Keyboard Parameter Files

The `key_101.key` file defines the Enhanced 101-key keyboard. Each line in the keyboard parameter file defines a single key. This key definition consists of two required parameters and the `MAP` parameter, which is optional. The following list briefly describes these parameters. See the *Command and Keyword Reference Guide* for details about parameter syntax.

KEY

Specifies the operation being defined (for example, A, PF1, QUIT)

SCAN

Specifies the location of the key on the keyboard

MAP

Specifies the types of windows in which this operation applies. By default, the operation applies to both function and terminal emulator windows.

Use the key maps that follow as references for your keyboard. If you have customized one of the files, indicate any changes in the margins.

Alt+

Press the Alt key and the specified key at the same time.

Ctrl+

Press the Ctrl key and the specified key at the same time.

Keypad+

Use the keys on your workstation keypad.

Shift+

Press a shift (left or right Shift or Caps Lock) key and the specified key at the same time.

101E+

Use the extra keys on the Enhanced 101-key keyboard.

Keys Mapped to Terminal Operations

The following table displays the keys mapped to terminal operations:

Operation	What It Does	Enhanced 101-keys Mapped to Operation
0 to 9	Types the indicated number (from 0 to 9) on the host screen	0 to 9

Operation	What It Does	Enhanced 101-keys Mapped to Operation
A to Z	Types the indicated uppercase letter (from A to Z) on the host screen	A to Z
a to z	Types the indicated lowercase character (from a to z) on the host screen	a to z
ATTN	Issues an attention to the terminal	Shift+Esc
BACK_QUOTE	Types an accent grave (backward quote) character on the host screen	`
BACK_SLASH	Types a back slash (\) on the host screen	\
BACK_SPACE	For a terminal emulator window, moves the cursor one space left	Back
BACK_TAB	Moves the cursor backwards one position	Shift+Tab
CLEAR	Clears the current screen of text	Keypad+Minus
COMMA	Types a comma on the host screen	,
CURSOR_BLINK	Turns blinking cursor on or off	User-definable
CURSOR_SEL	Changes the cursor from a block to an underline character or, conversely, from an underline character to a block	Scroll Lock
DEV_CNCL	Issues a DEV_CNCL instruction	User-definable
DOUBLE_QUOTE	Types double quotation marks on the host screen	"
DOWN	Moves cursor down one line	↓
DUP	Issues a DUP instruction	101E+Shift+Insert
END	Moves cursor to the bottom left corner of the screen	End
ENTER	Issues a carriage return	Enter

Operation	What It Does	Enhanced 101-keys Mapped to Operation
EQUAL	Types an equal sign on the host screen	=
ERASE_EOF	Erases all characters immediately following the cursor on this screen field	101E+End
ERASE_INPUT	Erases all unprotected fields on the host screen	101E+Alt+End
FAST_LEFT	Moves cursor two spaces to the left	101E+Alt+Left
FAST_RIGHT	Moves cursor two spaces to the right	101E+Alt+Right
FIELD_EXIT	Issues "Field Exit" to the iSeries console (5250 only)	Alt+E
FM	Issues a field mark instruction	101E+Shift+Home
HAT	Types the not equal character (~) on the host screen Note: Your workstation screen displays the ^ character.	Shift+6
HOME	Moves cursor to the first unprotected field on the host screen	Home
IDENT	Issues an IDENT instruction	User-definable
INS	Turns insert mode on	Insert
LEFT	Moves cursor left one character position	←
LEFT_BRACKET	Types the vertical bar character () on the host screen Note: Your workstation screen displays the [character.	[
LEFT_PAREN	Types a left parenthesis on the host screen	(
LINE_INS	Issues a Line Insert instruction	User-definable
LINE_DEL	Issues a Line Delete instruction	User-definable

Operation	What It Does	Enhanced 101-keys Mapped to Operation
MODE_SEL	Switches between a hardware console and software console session on an CA Automation Point terminal emulator	User-definable
NEWLINE	Line feed; moves cursor to the next line (3270 only)	Ctrl+Enter
PA1	Invokes the command assigned to the PA1 key	101E+Page Up
PA2	Invokes the command assigned to the PA2 key	101E+Page Down
PA3	Invokes the command assigned to the PA3 key	101E+Shift+Page Down
PF1 to PF9	Invokes the command assigned to the indicated PF key	F1 to F9
PF10	Invokes the command assigned to the PF10 key	F10
PF11	Invokes the command assigned to the PF11 key	F11
PF12	Invokes the command assigned to the PF12 key	F12
PF13 to PF24	Invokes the command assigned to the indicated PF key	Shift+F1 to Shift+F12
PF26 to PF36	Invokes the command assigned to the indicated PF key	User-definable
Pgdn	Moves the cursor down one full page	Page Down
PGDN	Issues a Page Down to the iSeries cosole (5250 only)	PgDn
PgUp	Moves the cursor up one full page	Page Up
PGUP	Issues a Page Up to the iSeries cosole (5250 only)	PgUp
POWER_RESET	Sends a reset-terminal instruction to the controller	Ctrl+Page Down
RESET	Issues a RESET instruction	Esc

Operation	What It Does	Enhanced 101-keys Mapped to Operation
RIGHT	Moves cursor right one character position	→
RIGHT_BRACKET	Types a cent-sign character (¢) on the host screen Note: Your workstation screen displays the] character.]
RIGHT_PAREN	Types a right parenthesis on the host screen)
SINGLE_QUOTE	Types a single quotation mark character (') on the host screen	'
SPACE	Types a blank space on the host screen	Space bar
SPLIT_BAR	Types a splitbar character () on the host screen	
START	Starts up a mainframe system connected to the workstation through a terminal emulator	User-definable
STOP	Shuts down a mainframe system connected to the workstation through a terminal emulator	User-definable
SYS_REQ	Issues a SYS_REQ instruction	Alt+F1
TAB	Moves cursor forward one tab position	Tab
TEST	Issues a TEST instruction	Alt+F8
UNDER	Types the underline character on the host screen	_
UP	Moves the cursor up one line	↑
!	Types an exclamation point on the host screen	!
#	Types a pound sign on the host screen	#
\$	Types a dollar sign on the host screen	\$

Operation	What It Does	Enhanced 101-keys Mapped to Operation
%	Types a percent sign on the host screen	%
&	Types an ampersand on the host screen	&
*	Types an asterisk on the host screen	*
+	Types a plus sign on the host screen	+
-	Types a minus sign on the host screen	-
.	Types a period on the host screen	.
/	Types a slash on the host screen	/
:	Types a colon on the host screen	:
;	Types a semicolon on the host screen	;
?	Types a question mark on the host screen	?
@	Types an at sign on the host screen	@
{	Types a left brace on the host screen	{
}	Types a right brace on the host screen	}
	Types a vertical bar on the host screen	
~	Types a tilde on the host screen	~
3270_ASSIGN	Issues an ASSIGN instruction	User-definable
3270_CANCEL	Issues a CANCEL instruction	User-definable
3270_IRPT	Issues an IRPT instruction	User-definable
3270_ISTEP	Issues an ISTEP instruction	User-definable
3270_PLAY	Issues a PLAY instruction	User-definable
3270_RECORD	Issues a RECORD instruction	User-definable

Operation	What It Does	Enhanced 101-keys Mapped to Operation
3270_RESTART	Issues a RESTART instruction	User-definable
3270_RULE	Issues a RULE instruction	User-definable
3270_SETUP	Issues a SETUP instruction	User-definable
3270_SVPCE	Issues an SVPCE instruction	User-definable
3270_SWAP	Issues a SWAP instruction	User-definable
3270_TOD	Issues a TOD instruction	User-definable
3270_1 to 3270_25	User-defined	User-definable
6530_ROLL_UP	Issues a Roll Up instruction (6530 only)	User-definable
6530_SROLL_UP	Issues a Shifted Roll Up instruction (6530 only)	User-definable
6530_ROLL_DOWN	Issues a Roll Down instruction (6530 only)	User-definable
6530_SROLL_DOWN	Issues a Shifted Roll Down instruction (6530 only)	User-definable
6530_NEXT_PAGE	Issues a Next Page instruction (6530 only)	User-definable
6530_SNEXT_PAGE	Issues a Shifted Next Page instruction (6530 only)	User-definable
6530_PREV_PAGE	Issues a Prev Page instruction (6530 only)	User-definable
6530_SPREV_PAGE	Issues a Shifted Prev Page instruction (6530 only)	User-definable
6530_RETURN	Issues a Return instruction (6530 only)	User-definable
6530_SRETURN	Issues a Shifted Return instruction (6530 only)	User-definable

Key Operations Specific to CA Automation Point

The following key operations control functions specific to CA Automation Point only:

Operation	What It Does	Enhanced 101-keys Mapped to Operation
CONNECT	Issues a connect request for the session. For more information, see the discussion of SESSCNTL CONNECTION in the <i>Command and Keyword Reference Guide</i> .	
CMD_HOST	Lets you send console commands to specific terminal emulator sessions. (This operation does not work from a REXX program.)	User-definable
DISCONNECT	Issues a disconnect request for the session. For more information, see the discussion of SESSCNTL CONNECTION in the <i>Command and Keyword Reference Guide</i> .	
EXECUTE	Executes a script from the current session window. (This operation does not work from a REXX program.)	User-definable
FONT_LARGER	Selects the next larger font in the current window and increases the window size proportionately	Ctrl+ →
FONT_SMALLER	Selects the next smaller font in the current window and decreases the window size proportionately	Ctrl+ ←
NEXT	Switches to the next session controlled by CA Automation Point. (This operation does not work from a REXX program; it only works in the red command area.)	Alt+N
PAUSE	Tells CA Automation Point to pause automation by releasing control of automated sessions. (This operation does not work from a REXX program.)	User-definable

Operation	What It Does	Enhanced 101-keys Mapped to Operation
SCREEN_LAYOUTS	Opens the Layouts dialog. For more details, see the section Layouts (see page 76)	
SCREEN_LOAD	Loads the current active layout (if it exists). For more information, see the section Active Layout . (see page 77)	
SCREEN_SAVE	Saves the current layout under last active layout (if it is set). For more information, see the section Active Layout . (see page 77)	
QUIT	Closes the specified window (but does not halt automation of the session in that window)	User-definable
WIN_NEXT	Jumps to the next CA Automation Point window	Alt+J

Key Operations for Asynchronous Console Sessions

The following table lists the key operations available for asynchronous sessions and the keystrokes mapped to those operations.

Keys	Operation	What It Does
Ctrl+A	SOH	Sends an SOH character
Ctrl+B	STX	Sends an STX character
Ctrl+C	ETX	Sends an ETX character
Ctrl+D	EOT	Sends an EOT character
Ctrl+E	ENQ	Sends an ENQ character
Ctrl+F	ACK	Sends an ACK character
Ctrl+G	BEL	Sends a BEL character
Ctrl+K	VT	Sends a VT character
Ctrl+L	FF	Sends an FF character
Ctrl+N	SO	Sends an SO character
Ctrl+O	SI	Sends an SI character
Ctrl+P	DLE	Sends a DLE character

Keys	Operation	What It Does
Ctrl+Q	DC1	Sends a DC1 character
Ctrl+R	DC2	Sends a DC2 character
Ctrl+S	DC3	Sends a DC3 character
Ctrl+T	DC4	Sends a DC4 character
Ctrl+U	NAK	Sends a NAK character
Ctrl+V	SYN	Sends a SYN character
Ctrl+W	ETB	Sends an ETB character
Ctrl+X	CAN	Sends a CAN character
Ctrl+Y	EM	Sends an EM character
Ctrl+Z	SUB	Sends a SUB character
Ctrl+Shift+2	NULL	Sends a NULL character
Ctrl+6	RS	Sends an RS character
Ctrl+[ESC	Sends an ESC character
Ctrl+\	FS	Sends an FS character
Ctrl+]	GS	Sends a GS character
Ctrl+ -	US	Sends a US character
User-definable	BREAK	Sends a break signal

Customizing Your Keyboard File

If you want to customize a keyboard file by remapping some of the keys, you can use the CA Automation Point keyboard configuration program to automatically generate a keyboard parameter file based on your responses to specific questions. You can also modify a keyboard parameter file using an editor.

The keyboard configuration program and CA Automation Point cannot run at the same time. If you want to run the keyboard configuration program, verify that CA Automation Point is not running. Likewise, if you want to run CA Automation Point, verify that the keyboard configuration program is not running.

To specify a keyboard file for CA Automation Point to use, go into Configuration Manager, Expert Interface, Automation, Automation Point Desktop Settings. Specify the keyboard file in the Mapping File field.

Running the Keyboard Configuration Program

The CA Automation Point keyboard configuration program can help you customize the keyboard parameter file. If the standard files do not exactly meet your needs, use the utility to redefine the appropriate keys.

To run the CA Automation Point keyboard configuration program

1. Start Configuration Manager.
2. Choose Expert Interface, Automation, Automation Point Desktop Settings.
3. Click Customize Keyboard. The Keyboard dialog displays.
4. In the Input Key File field, enter an existing keyboard file name or use the default. In the Output Key File, type the new file name (with the extension .key) where your new definitions will be stored.
5. Click OK. The keyboard configuration program starts.
6. When prompted, press the key that you want to define. Do **not** press the Alt or Shift key.

The program determines the value associated with the location of the key and displays that number on the first line. Both the current and default definitions for the key appear. For example, if you press the A key, the following output is produced:

```
Scan: 30
Current key definitions:
KEY=SOH,SCAN=30(ctrl),map=(automate)
KEY=A,scan=30(upper),map=(terminal,automate)
KEY=a,SCAN=30(lower),map=(terminal,automate)
KEY=ALT_A,SCAN=30(alt),map=(automate)
Default key definitions:
KEY=SOH,SCAN=30(ctrl),map=(automate)
KEY=A,SCAN=30(upper),map=(terminal,automate)
KEY=a,SCAN=30(lower),map=(terminal,automate)
KEY=ALT_A,SCAN=30(alt),map=(automate)
```

```
Current definitions are listed. Press key:
K to keep, A to add, D to delete, M to modify,
R to use defaults, Q to write file:
```

7. Select any of the options that the program presents next, such as:

Option	Explanation
K	Keep the current definition (if any).
A	Add a definition.

Option	Explanation
D	Delete all current definitions. The program asks you to confirm your selection. Answer Y to delete all current definitions.
M	Delete one or more definitions. The program displays the definitions one at a time and lets you keep or delete each definition. To change an incorrect definition, delete it and replace it with a correct one.
R	Replace the current definitions with the default definitions. Answer Y to replace the current definitions.
Q	Exit the program.

After you select an option, answer any additional questions that the program asks.

- When you have defined all appropriate keys, exit the program by pressing a key and selecting Q for quit.

Example 1:

The following example shows how to create a key definition.

Suppose you want to associate an IRPT instruction for your mainframe session with keystrokes CTRL+ALT+2. According to the previous key map table, the operation for an IRPT instruction is 3270_IRPT.

To create a key definition

- Run the keyboard configuration program.
- Press the key to which you want to assign the IRPT instruction (in this example, 2).
- Type A to add a definition.
- Type the operation for the instruction (in this example, the operation is 3270_IRPT).
- Type the first keystroke (In this example, CTRL is the first shift operation).
- Type the remaining keystrokes (In this example, ALT is the next shift operation).
- Press Enter to indicate there are no more shift operations.
- Enter your map request. See the *Command and Keyword Reference Guide* for details on the MAP parameter.
- When you are finished with the map prompt, press Enter.
- To save your definition, type Y.

Example 2:

The following table displays more sample key definitions and how they work:

Key Definition	How it works
KEY=a,SCAN=30(LOWER)	Define key 30 as a lowercase a.
KEY=A,SCAN=30(UPPER)	Define key 30, shifted, as an uppercase a.
KEY=SINGLE_QUOTE,SCAN=40(SHIFT)	Define key 40, shifted, as a single quotation mark.
KEY=PF1,SCAN=2(ALT), MAP=TERMINAL	Define Alt+2 as PF1 on terminal emulator windows.
KEY=QUIT,SCAN=61	Define key 61 as the Quit key.
KEY=3270_IRPT,SCAN=2(CTRL,ALT), MAP=TERMINAL	Define Ctrl+Alt+2 as a 3270_IRPT operation.

Defining Your Own Operations

If you are using a scan code file to define a nonstandard terminal keyboard (such as a processor console keyboard), the list of valid operations provided in this appendix may not contain all of the operations you need. For that reason, CA Automation Point provides 25 different operation names that you can map to operations of your choice. The user-definable operations are valid only for terminal emulator keyboards identified in scan code files.

By default, operations 3270_1 to 3270_25 are dummy names for undefined operations.

To map actual operations to default dummy operation names

1. Use the utility program to map a dummy operation, such as 3270_2, to an available key on your keyboard.

Your keyboard parameter entry may look like this:

```
KEY=3270_2,SCAN=35(CTRL,ALT),MAP=TERMINAL
```

In the preceding example, SCAN=35 is the scan code for the H key.

2. Specify the dummy operation name (3270_2) and the scan code of the actual operation in the scan code file for the terminal emulator to which this operation applies.

Your scan code entry might look like this:

```
3270_KEY=3270_2,3270_SCAN=(4f,0c,cf)
```

For more information, see the section [Customizing Your Scan Code Files](#) (see page 472) in this appendix.

Using Key Operations in Rules, REXX Programs, and Scripts

Note the following when using key operations in CA NSM rules, REXX programs, and scripts:

- The key operations for asynchronous sessions are defined with the MAP=AUTOMATE keyboard parameter statement.
- Key operations defined with the MAP=TERMINAL parameter are valid only for 3270 terminal emulator sessions.
- If you do not specify a MAP parameter statement, or if you specify MAP=(AUTOMATE,TERMINAL), the key operations are valid for both asynchronous and 3270 sessions.

Example 1:

To invoke a key operation (such as QUIT or RESET) in a 3270 session from CA Automation Point rules, REXX programs or scripts, use this syntax:

```
@"operation"
```

Example 2:

Suppose that you want to send an end-of-text (ETX) character to an asynchronous session whenever CA Automation Point receives a message reporting that the keyboard is locked. CA Automation Point maps the ETX operation to the Ctrl+C keystroke sequence, so the following rule transmits an ETX operation to the session:

```
MSGID(KEYBOARD LOCKOUT) , SESSCMD(@"ETX")
```

VT52, VT100, and VT320 Sessions

There are special considerations for VT52, VT100, and VT320 sessions. The Ctrl+H key maps to an operation named DELTA (represented by the ASCII character D, which has a decimal value of 127), a *nondestructive* backspace. The syntax for sending the non-destructive backspace from a REXX program is @"DELTA".

The Backspace key maps to the BACK_SPACE operation, a *destructive* backspace. When specifying a destructive backspace from a REXX program, you need to suppress the Enter key. The syntax is:

```
'SESSCMD /@"BACK_SPACE"@"SEND"/ SESSION(sessname)'
```

Customizing Scan Code Files

This section describes how you can customize scan code files.

Understanding Scan Codes

To understand scan code files, you must first understand the *keyboard parameter file*. When you press a key, you expect the operating system to perform some operation; for example, when you press the A key, you want a lowercase a to appear on your screen. The keyboard parameter file maps a keystroke to a particular operation. For a host session to recognize an operation, it is necessary to map the operation to terminal-specific *scan codes*. You may need up to three scan codes to represent a 3270 key operation.

If you configure CA Automation Point to emulate standard 3278 and 3279 terminals, CA Automation Point translates the scan codes for you. If you configure CA Automation Point to emulate other types of terminals (such as a 3279-2C terminal), you need to specify the appropriate *scan code file*. If CA Automation Point does not provide a specific scan code file for a particular non-3278/non-3279 terminal, you must create one for it.

Scan Code Parameter Summary

Each line in a scan code file defines a single operation. This operation definition consists of the following two required parameters that identify the name of the operation that is being defined and the scan code for that operation. See the *Command and Keyword Reference Guide* for parameter syntax.

3270_KEY

Specifies the operation being defined (for example, A, PF1, and so on)

3270_SCAN

Specifies the terminal-specific scan code for the operation

Customizing Your Scan Code Files

CA Automation Point provides sample scan code files in the Distrib directory. The sample files (with .cod file name extensions) offer mappings for several types of terminals. Use your editor to modify these scan code files, if necessary, or to create new ones.

Note: User-created scan code files (with .cod as the file name extension) should be moved to the %AP_SITE%\MyFiles directory.

Sample Operation Definitions

The following example shows some sample operation definitions for a 3270-type terminal:

Operation Definition	What it Does
3270_KEY=a,3270_SCAN=60	Map a lowercase "a" to scan code 60.
3270_KEY=A,3270_SCAN=(4d,60,cd)	Map an uppercase "A" to scan code 4d,60,cd.
3270_KEY=SINGLE_QUOTE,3270_SCAN=12	Map a single quotation mark to scan code 12.
3270_KEY=PF1,3270_SCAN=(4f,21,cf)	Map PF1 to scan code 4f,21,cf.
3270_KEY=MODE_SEL,3270_SCAN=50	Map the MODE_SEL operation to scan code 50.

Identifying Terminal Emulators

When you define a session to CA Automation Point the Terminal window attribute identifies the type of terminal emulator that you intend to use for the specified session. If you want to use the standard 3278 or 3279 terminal emulators, choose 3278 or 3279 from the list of terminals on the Automation Point Session Definition dialog.

Note: Define the terminal type and other session characteristics *before* starting CA Automation Point.

Associating a Scan Code File with a Session

To associate a scan code file with a session definition

1. Go into Configuration Manager and choose Expert Interface, Automation, Session Definition Sets. Choose the session for which you want to assign a new scan code file.
2. On the Automation Point Session Definition dialog, click Customize Session Settings; then, on the dialog that displays, click Advanced.

If the type of session you have chosen allows you to assign a scan code file, the Host Scan File field will be enabled on the dialog.

3. Type the name or browse to choose a scan code file from the list of discovered scan code files.

Customizing Script Files

This section describes how you can set up script files to automate certain tasks.

Understanding How Script Files Automate Tasks

CA Automation Point uses files called *scripts* to automate these tasks:

- Establishing or recovering communications with the system
- Configuring a console for CA Automation Point to operate
- Performing a complex series of actions, such as an IPL, at the operator's request

A script file contains the sequence of keystrokes that an operator normally enters to perform a console function. Without script files, you would have to put each console in the mode appropriate for CA Automation Point before starting the product.

Types of Scripts

CA Automation Point uses the following types of scripts:

- Session state scripts
- Initialization scripts
- Pause scripts
- Operator-initiated scripts

Specifying Scripts to Use

You specify the file names of scripts you want CA Automation Point to use on the Scripts dialog in Configuration Manager.

To specify scripts you want CA Automation Point to use

1. Select Expert Interface, Automation, Scripts.
2. Choose a console type or state and associate it with the script you want, using the HTML help to guide you.

You can associate a console type with a script for a specific session when you create or modify a session definition.

Using Session State Scripts

A session state script configures a session so that CA Automation Point can control it. For example, for CA Automation Point to work properly with an MCS console, the console must be in nondelete mode, must have no display area, and its messages must have time stamps and job numbers. A session state script automatically issues the appropriate commands to put the MCS console in non-delete mode.

CA Automation Point checks the screen continuously to make sure that the session has not changed states. For example, suppose that console session S008 is currently in an MCS console and you defined this session to CA Automation Point, and the ConsoleType/State list includes IPL, MCS, PAUSE, and INIT.

The session definition causes CA Automation Point to manage this session as an MCS console. If z/OS initializes and session S008 receives the first IPL message, CA Automation Point determines that session S008 is now an IPL console and immediately begins processing messages from the session using the rules file. When the IPL progresses to the point where the session changes to an MCS console, CA Automation Point processes the script named MCS.scr. Next, it resumes processing messages from session S008 using rules defined in the rules file.

Note this additional information about session state scripts:

- The VSE state script always executes when the host console changes to the VSE state from any other state.
- The MCS state script always executes when CA Automation Point determines that an MCS console is in any mode other than non-delete mode.

Using Initialization Scripts

An initialization script recovers the console, if possible, when processing stops because CA Automation Point cannot determine the console state. Initialization scripts are most useful for applications that a user ID or device can log on to, such as a Remote Console/MVS VTAM console or a z/OS guest under VM.

Normally, a session defined as type MCS uses the initialization script file MCSINIT.scr (supplied with CA Automation Point) to issue the PA2 key when the console is in the initialization state.

Using Pause Scripts

A pause script can restore the console for manual use. When an operator presses the PAUSE key, CA Automation Point executes any pause scripts you have defined. For example, you could use a pause script to put an MCS console back into roll delete mode to prevent console messages from backing up while CA Automation Point is not managing the session.

Using Operator-initiated Scripts

From a command area, you can select the Start REXX program or script option to start a script manually. CA Automation Point asks for the name of the script file to execute. The specified script file executes in the session indicated by the command area. You can use operator-initiated scripts to execute predefined functions on command.

Creating a Script

A script consists of a series of statements with one script keyword statement per line.

CA Automation Point executes script statements line by line. Each time a script executes, CA Automation Point evaluates all SEARCH statements in the order that they occur in the script.

Script Keyword Summary

You can use the following keywords in a script. See the *Command and Keyword Reference Guide* for keyword syntax.

ENDSEARCH

Terminates the preceding SEARCH

ERROR

Specifies what happens if a search fails

KEY

Sends a keystroke string to the target session

SEARCH

Searches the entire screen for a string of data

WAIT

Causes the script to wait for the preceding keystroke string to be accepted by the system

XKEY

Sends a single CA Automation Point operation instruction to a host session, regardless of whether the keyboard is locked

Specifying Keystrokes

To instruct a script to press specific keys for you, you can specify the following keystrokes in a KEY keystroke string:

- A key abbreviation
- An operation instruction

You can specify keystroke strings with or without quotes. For example, these keystroke strings are equivalent:

```
KEY=(K A,NONE@E)
KEY=('K A,NONE@E')
```

Key Abbreviations

A key abbreviation consists of the @ character followed by an alphabetic letter (uppercase or lowercase), a digit, or the character itself. For example, the abbreviation for the Enter key is @E. The abbreviations are mnemonic; that is, they correspond as much as possible to the actual name of the key. For example, @C is the abbreviation for the Clear key, @A is the abbreviation for the Alt key, and so on. The following table provides a comprehensive list of valid key abbreviations.

Note: When you press a shift key (such as Alt) in combination with another key, the meaning of that key changes. For example, the Alt+Clear keys constitute the Test key; to send the Test keystroke, type @A@C.

The following is a list of key abbreviations.

Abbrev.	For Key	Abbrev.	For Key	Abbrev.	For Key
@u	PageUp	@PgUp			
@v	PageDown	@PgDn			
@A	ALT	@V	Cursor Down	@c	PF12
@B	Backtab	@X	Reserved	@d	PF13
@C	CLEAR	@Z	Cursor Right	@e	PF14
@D	Delete	@@	@ character	@f	PF15
@E	ENTER	@0	Home	@g	PF16
@F	Erase EOF	@1	PF1	@h	PF17
@H	Help	@2	PF2	@i	PF18
@I	Insert	@3	PF3	@j	PF19
@K	Copy	@4	PF4	@k	PF20

Abbrev.	For Key	Abbrev.	For Key	Abbrev.	For Key
@L	Cursor Left	@5	PF5	@l	PF21
@N	New Line	@6	PA6	@m	PF22
@Q	Finish (Quit)	@7	PF7	@n	PF23
@R	Reset	@8	PF8	@o	PF24
@S	Shift	@9	PF9	@x	PA1
@T	Tab	@a	PF10	@y	PA2
@U	Cursor Up	@b	PF11	@z	PA3

Operation Instructions

An operation instruction consists of the @ character followed by a keyboard operation name enclosed in quotation marks. For example, @"MODE_SEL" is the instruction for the MODE SELECT key.

For a list of valid keyboard operation names, see the section [Customizing Keyboard Parameter Files](#) (see page 457) in this appendix .

How to Start a Script

You can start scripts from:

- REXX programs
- CA Automation Point rules
- An CA Automation Point menu option
- The CA Automation Point command area

The following sections describe each method for starting scripts.

Starting Scripts From REXX Programs

Start a script from a REXX program by specifying the SCRIPT command, as shown:

```
"SCRIPT scriptname SESSION(sessname)"
```

The following example invokes a script, called PAUSE.scr, that restores the MCS console for session S028 for manual use:

```
"SCRIPT PAUSE.SCR SESSION(S028)"
```

For details, see the description of the SCRIPT command in the *Command and Keyword Reference Guide*.

Starting Scripts From Rules

Start a script from a rule by specifying the SCRIPT rule keyword, as shown:

```
SCRIPT(filename)
```

or

```
SCRIPT((filename) SESSION(sessid))
```

The following example starts the MYSCRIPT script when message \$HASP308 is issued:

```
MSGID ($HASP308), SCRIPT(MYSCRIPT.SCR)
```

For details, see the description of the SCRIPT keyword in the *Command and Keyword Reference Guide*.

Starting Scripts From Menus

You can issue scripts by selecting a menu item from the menus. For more information about adding menu items to existing menus, see the chapter "[Managing Sessions Using CA Automation Point Windows](#). (see page 75)"

The following example shows sample menu control statements that you could add to your USER.MNU file to invoke the SCRIPT command processor:

```
SUBMENU=UserApps, NAME=(~UserApps, help_window),  
ITEM=('~My Script', XCCMD('SCRIPT MYSCRIPT.SCR SESSION(HELPDESK)'))
```

Sample Script and Console Screen

The following session state script configures an MCS console session for use by CA Automation Point.

```
*-----*
* FILENAME: MCS.SCR *
*
* USE:      This script configures an MCS console for control by Automation *
*           Point by issuing 'K' (control) commands.  To execute RULES *
*           against MCS console messages, and to send those messages to the *
*           Automation Window, Automation Point requires the console to be in *
*           non-delete mode, with time and job stamps.  Automation will not *
*           commence until the console is in this mode. *
*
* UPDATED:  7-JUL-94 *
*
* NOTES:    One characteristic of MCS consoles is that if you enter an invalid *
*           or out-of-sequence 'K' command, the console will not accept any *
*           new commands until you enter the command correctly or press the *
*           PA2 key to reset. *
*-----*

ERROR=IGNORE          * If search fails, proceed to next search

*-----*
* Put the console in non-delete mode, 20 lines/segment, etc. *
*
* Note! The number of lines per segment will vary from console to console. *
*       Make sure that that the number used below matches your console. *
*-----*

SEARCH=(IEE151I)      * Delete request inconsistent
KEY=(@y)
WAIT=1

ENDSEARCH
```



```

*-----*
* The following message appears on the status line of a healthy pre MVS/SP 4.1 *
* MCS console. If found, set the console characteristics for AP's use.      *
*                                     *                                     *
* IEE152I  ENTER  CANCEL  D C,K                                           *
* If we attempt to set it to roll more lines than it has, the status line is: *
* IEE156I K INVALID OPERAND -20                                           *
*-----*
SEARCH=(IEE152I)      * Look for 'IEE152I' - if found, reconfigure console
KEY=(k s,del=n,seg=39,con=n,mform=(t,j)@E)
WAIT=1               * Wait 1 second for the command to be accepted
SEARCH=(IEE156I)    * Look for 'IEE156I' - if found, try again
KEY=(@y)             * (@y=PA2)
WAIT=1               * Wait 1 second for the command to be accepted
KEY=(k s,del=n,seg=20,con=n,mform=(t,j)@E)
WAIT=1               * Wait 1 second for the command to be accepted
SEARCH=(IEE156I)    * Look for 'IEE156I' - if found, try again
KEY=(@y)             * (@y=PA2)
WAIT=1               * Wait 1 second for the command to be accepted
KEY=(k s,del=n,seg=19,con=n,mform=(t,j)@E)
WAIT=1               * Wait 1 second for the command to be accepted
ENDSEARCH            * End of commands to perform if 'IEE156I' is found
ENDSEARCH            * End of commands to perform if 'IEE156I' is found
ENDSEARCH            * End of commands to perform if 'IEE152I' is found

*-----*
* If the IEE152I is still there, we want to delete the display area, if one is *
* defined, because display areas cause problems for rules processing.      *
*-----*
SEARCH=(IEE152I)    * Look for 'IEE152I' - if found, reconfigure console

KEY=(k e,d@E)      * Clear the display area (@E=ENTER)
WAIT=1              * Wait 1 second for the command to be accepted

```

```

*-----*
* If the console didn't have a display area, the following error message *
* will have been displayed in the console status line: *
* * * * *
* IEE151I DELETE REQUEST INCONSISTENT-NO DISPLAY ON SCREEN  ENTER  CANCEL *
* * * * *
* Just in case that happened, send a PA2 using the mnemonic '@y' *
*-----*

KEY=(@y) * (@y=PA2)
WAIT=1 * Wait 1 second for the command to be accepted
KEY=(k a,none@E) * Delete the area so we won't get more area displays
WAIT=1 * Wait 1 second for the command to be accepted

KEY=(axc here@E) * Tell CA-OPS/MVS that AP is active at this console
WAIT=1 * Wait 1 second for the command to be accepted

ENDSEARCH * End of commands to perform if 'IEE152I' is found

*-----*
* The following message appears on the status line of a healthy MVS/SP 4.1 or *
* above (including all z/OS versions) MCS console. If found, set the console *
* characteristics for AP's use. *
* * * * *
* IEE612I CN=02 DEVNUM=60D SYS=S032 CMDSYS=S032 *
* If we attempt to set it to roll more lines than it has, the status line is: *
* IEE156I K INVALID OPERAND -20 *
*-----*
SEARCH=(IEE612I) * Look for 'IEE612I' - if found, reconfigure console
KEY=(k s,del=n,seg=39,con=n,mform=(t,j))@E
WAIT=1 * Wait 1 second for the command to be accepted
SEARCH=(IEE156I) * Look for 'IEE156I' - if found, try again
KEY=(@y) * (@y=PA2)
WAIT=1 * Wait 1 second for the command to be accepted
KEY=(k s,del=n,seg=20,con=n,mform=(t,j))@E
WAIT=1 * Wait 1 second for the command to be accepted
SEARCH=(IEE156I) * Look for 'IEE156I' - if found, try again
KEY=(@y) * (@y=PA2)
WAIT=1 * Wait 1 second for the command to be accepted
KEY=(k s,del=n,seg=19,con=n,mform=(t,j))@E
WAIT=1 * Wait 1 second for the command to be accepted
ENDSEARCH * End of commands to perform if 'IEE156I' is found
ENDSEARCH * End of commands to perform if 'IEE156I' is found
ENDSEARCH * End of commands to perform if 'IEE612I' is found

```

```

*-----*
* If the IEE612I is still there, we want to delete the display area, if one is *
* defined, because display areas cause problems for rules processing.      *
*-----*
SEARCH=(IEE612I)          * Look for 'IEE612I' - if found, reconfigure console

KEY=(k e,d@E)           * Clear the display area (@E=ENTER)
WAIT=1                  * Wait 1 second for the command to be accepted

*-----*
* If the console didn't have a display area, the following error message *
* will have been displayed in the console status line:                    *
*-----*
* IEE151I DELETE REQUEST INCONSISTENT-NO DISPLAY ON SCREEN  ENTER  CANCEL *
*-----*
* Just in case that happened, send a PA2 using the mnemonic '@y'         *
*-----*

KEY=(@y)                * (@y=PA2)
WAIT=1                  * Wait 1 second for the command to be accepted
SEARCH=(IEE612I)
KEY=(k a,none@E)       * Delete the area so we won't get more area displays
WAIT=1                  * Wait 1 second for the command to be accepted

SEARCH=(IEE612I)
KEY=(axc here@E)       * Tell CA-OPS/MVS that AP is active at this console
WAIT=1                  * Wait 1 second for the command to be accepted
ENDSEARCH
ENDSEARCH
ENDSEARCH               * End of commands to perform if 'IEE612I' is found

*-----*
* Search for error messages that the MCS console driver recognizes as     *
* bad modes and clear them.                                               *
*-----*

SEARCH=(IEE160I)        * UNVIEWABLE MESSAGE
KEY=(k e,d)
WAIT=1
ENDSEARCH

SEARCH=(IEE163I MODE= RD)
KEY=(@y)
WAIT=1
ENDSEARCH

```

```
SEARCH=(IEE163I MODE=)
KEY=(@E)
WAIT=1
ENDSEARCH
```

```
SEARCH=(IEE164I)
KEY=(@y)
WAIT=1
ENDSEARCH
```

MCS Console Screen After the Sample Script Executes

The following screen shows how an MCS console appears after the preceding sample script executes; the MCS.SCR script has placed the console in non-delete mode and the display area (if one was defined) has been deleted. Note that the bottom of the screen no longer displays message IEE163I indicating roll delete mode.

```
*05.45.47 OX *IEA994A ALL SYS1.DUMP DATA SETS ARE FULL AND NO SVC
* DUMPS CAN BE TAKEN
*06.00.20 OX S 877 *DPM436I - USER DATA SET FULL AND NOW CLOSED
- 07.14.58 OX J 925 $HASP395 DSIMP11B ENDED
  07.14.58 OX JES2 $HASP309 INIT 2 INACTIVE ***** C=GDCBA
  07.18.04 OX J 926 $HASP100 DSIMP11B ON INTRDR DSI
- 07.18.11 OX J 926 $HASP373 DSIMP11B STARTED - INIT 2 - CLASS G - SYS
- IPOX
- 07.18.57 OX J 922 $HASP395 DSIFE11R ENDED
  07.18.57 OX JES2 $HASP000 OK
  07.18.57 OX JES2 $HASP309 INIT 1 INACTIVE ***** C=ABCDG
  07.18.57 OX J 923 $HASP317 DSIMP11B 0004 DATA SETS CANCELLED
  07.18.57 OX J 923 $HASP250 DSIMP11B IS PURGED
  07.18.58 OX JES2 $HASP000 OK
  07.18.59 OX J 925 $HASP317 DSIMP11B 0004 DATA SETS CANCELLED
  07.18.59 OX J 925 $HASP250 DSIMP11B IS PURGED
- 07.20.11 OX J 926 IEF450I DSIMP11B SAS CUSTRPT - ABEND=S000 U0999 REASON
- 00000000
- 07.20.13 OX J 926 $HASP395 DSIMP11B ENDED
  07.20.13 OX JES2 $HASP309 INIT 2 INACTIVE ***** C=GDCBA
IEE6121 CN=05 DEVNUM=B4C SYS028 CMDSYS=S0
```

Appendix B: Using the TPF - CA Automation Point REXX Interface

This section contains the following topics:

[Overview](#) (see page 485)

[TPF Components](#) (see page 485)

[Setup Requirements](#) (see page 486)

[Message Processing](#) (see page 487)

Overview

The TPFCON.REX application is a REXX-based interface for a TPF system to be automated by CA Automation Point. The interface processes all system-and user-initiated messages through the CA Automation Point rules engine, allowing normal automation tasks to be executed against those messages. Through the use of the CA Automation Point Global Variable Environment (GLV), the entire text of a TPF message regardless of length is available to the system to manipulate and act on as required.

While the interface is running, a log is generated that contains all information processed by the interface in the format seen by the interface. The log is called TPFLOG.log. Each night at midnight this log is cut and renamed to TPFLOG`today`sdate.log, and a new version of TPFLOG is started.

TPF Components

The following are components of the TPF REXX Interface

TPFCON.REX

TPFCON.REX is the main module that manages the interface. It processes all system- and user-initiated messages through the CA Automation Point rules engine. This file resides in the *installDir*\Distrib directory.

TPFREAD.REX

TPFREAD.REX is the REXX program that parses the TPF message into a REXX stem variable called TPFWD. This variable gives you access to every word in a message to act on. This file resides in the *installDir*\Distrib directory.

axcrules.rul

The file `axcrules.rul` contains sample rules to run against the TPF session. This file resides in the `installDir\Distrib` directory.

Setup Requirements

This section describes setup requirements for the TPF - CA Automation Point REXX Interface.

On the TPF System

For the interface to work correctly, you must ensure that the following three requirements are met on the TPF system:

- The TPF system console must be running in 3270 mode.
- The TPF administrator must configure the system to add a plus character (+) to the end of system messages, and also make sure that all user programs end their messages with this character. This is necessary because the TPFCON interface determines that it has reached the end of a message when it finds a plus character (+).
- CA Automation Point must have control of the console. Specifically, CA Automation Point must control when a MORE... state is cleared. If something other than CA Automation Point clears a MORE..., CA Automation Point loses its place and messages can potentially be lost or incomplete. To ensure that only CA Automation Point clears a MORE., the clear action capability must be removed from the Enter key. Then, when an operator types a command and presses Enter, the command is placed on the stack to be executed, but it will not clear any MORE... state existing at the point where the command is entered.

On the CA Automation Point Machine

For the TPF REXX program to work with CA Automation Point, you must create a TPF session using Configuration Manager.

To do create a TPF

1. From Configuration Manager, choose Expert Interface, Automation, Session Definition Sets.
2. In the Session Definition Sets dialog, add a new session.
3. Choose a console type of TPF3270 in the session definition
4. Choose the appropriate terminal.

5. Customize your session settings in the 3270 Session Settings dialog.
6. Either copy sample rules for a TPF session into your active rules file (default is AXCRULES.RUL), or add custom rules to the active rules file using the rules in axcrules.rul as a guide.

When CA Automation Point starts up, the TPFCON.REX application controls a TPF session on the desktop.

Message Processing

TPF messages can be very long, and a limit exists for the length of a message that is processed by rules. For this reason, the interface creates a non-volatile GLV that holds messages that are greater than two lines. Two-line messages are handled in the normal fashion, without a GLV being created. To ensure that the GLV variable names are unique, the interface adds a word in the second word place in the message. The format of the added word is simply the MSGID followed immediately by a timestamp. For example:

Original message:

```
DSYS0001I 09.44.59 THE SYSTEM IS IN NORM STATE
```

New message:

```
DSYS0001I DSYS0001I094459123 09.44.59 THE SYSTEM IS IN NORM STATE
```

TPFREAD.REX is written to accept the new WORD2 and the entire message as its two arguments. TPFREAD looks to see if a GLV exists, matching WORD2. If so, it retrieves the value of the GLV and parses it into the stem variable called TPFWD. If not, it uses the message passed as the second argument and parses it into the stem variable. The parsing process removes WORD2, thereby converting the message to its original format.

You can then use the REXX parsed words and pass them to another REXX program, or take action on it in TPFREAD.

Appendix C: TN3270 and TN5250 Considerations

This appendix discusses the following TN3270 issues:

- How to interpret the TN3270 Status line
- How to set CA Automation Point to record a TN3270 session for diagnostic purposes.

The TN3270E Status Line

The Status Line (or OIA - Operator Information Area) is the line at the bottom of a TN3270 session where status information about the session displays.

The meaning of each field on the TN3270E Status line (OIA) is as follows:

Column 1-3: Connection status indicator:

4B■	A TN3270E connection has been established.
NVT	NVT-mode connection has been established. This generally occurs while the session is first initialized.
N	The session is not connected.

Columns 10-17: System status indicator:

X SYSTEM	Input from the session is inhibited (for example, the host system is busy)
X ?+	Input from the session is inhibited (for example, the keyboard is locked)
XPROG505	The session is not connected.
XPROG572	A TN3270 command (SBA, RA, or EUA) contains a screen address pointing outside of the current screen.

Columns 37-39: Security indicator.

SSL	The session has been configured with SSL enabled.
<blank>	The session has not been configured or configured without SSL enabled.

Columns 42-53: Terminal type indicator.

Columns 56-71: LU/Device name. This can be either the LU/device name assigned by the TN3270E server for this session during protocol negotiation, or the LU/Device Name specified in the session definition using Configuration Manager.

Columns 74-79: Cursor position indicator.

rr/ccc Row (*rr*) and column (*ccc*) number of current cursor position.

Note: On a model 5-type session, the cursor position indicator starts at column 125.

The TN5250 Status Line

The Status Line (or OIA - Operator Information Area) is the line at the bottom of a TN5250 session where status information about the session displays.

The meaning of each field on the TN5250 Status line (OIA) is as follows:

Column 1-3: Connection status indicator:

4B ■ A TN5250 connection has been established.

NVT NVT-mode connection has been established. This generally occurs while the session is first initialized.

N The session is not connected.

Column 7: Insert Mode indicator:

^ Insert Mode is active.

Columns 10-17: System status indicator:

X SYSTEM Input from the session is inhibited (for example., the host system is busy)

X ?+ Input from the session is inhibited (for example, the keyboard is locked)

XPROG505 The session is not connected.

Columns 23-24: Message Waiting indicator

MW System host turned on Message Waiting

Columns 26-28: Security indicator.

SSL The session has been configured with SSL enabled.

<blank> The session has not been configured or configured without SSL enabled.

Columns 41-52: Terminal type indicator.

Columns 55-70: Device name. This can be either the Device name assigned by the TN5250 server for this session during protocol negotiation, or the Device Name specified in the session definition using Configuration Manager.

Columns 73-78: Cursor position indicator.

rr/ccc Row (*rr*) and column (*ccc*) number of current cursor position.

Support for TN5250 Auto Sign-On

IBM i5 (iSeries) Operating System's support a feature called "Auto Sign-On". The purpose of this feature is to allow a login to bypass the I5/OS initial sign-on screen. Also, several additional startup options (library name, menu name and initial program) and be specified that will run at sign-on time and take the screen directly to the given menu or run the given program.

In order to enable auto sign-on, check the Enable Auto Sign-on checkbox on the 5250 Session Settings dialog and specify at least a user ID and password.

Note: The user ID and password values specified on this dialog are re-loaded from configuration every time CA Automation Point tries to connect to the session. In other words, changes made to these fields are applied immediately and do not require a restart of the Automation Point Desktop.

Auto Sign-on supports three different password encryption techniques:

Plaintext

Indicates that the password is sent in clear text (i.e., not encrypted) to the i5/OS host.

Standard

Indicates that the password will be encrypted using the DES* encryption technique before it is sent to the i5/OS host.

Elevated

Indicates that the password will be encrypted using the SHA encryption technique before it is sent to the i5/OS host.

Note: For a detailed description of these encryption algorithms, see section 5 of IBM document RFC 4777.

You decide how to encrypt the password by first determining how the i5/OS host has Password Level configured.

Note: To understand how Password Level effects your decision about which encryption to use for Auto Sign-on, see IBM's documentation on Password Level and QPWDLVL.

To query the current host value for QPWLVL, issue the following i5/OS command:

```
WRKSYSVAL QPWLVL
```

Possible returned values are:

QPWLVL 0 or 1

Password length is restricted to 10 characters. CA Automation Point can use Plaintext or Standard encryption. Elevated encryption does not work with this host.

QPWLVL 2 or 3

Password length can be up to 128 characters and are case sensitive. CA Automation Point can use Plaintext or Elevated encryption. Standard encryption does not work with this host.

Recording a TN3270 or TN5250 Session

Input/output data stream associated with TN3270 or TN5250 sessions can be recorded into a log file. This will help CA Technical Support diagnose customer problems, providing the ability to "see" the console just as it appeared at the customer site. At CA Technical Support's direction, you may be asked to enable recording of a TN3270 or TN5250 session.

To enable recording of a TN3270 or TN5250 session

1. Create or modify the OICTNET.INI configuration file as follows:

```
[Log]
sessname_EventLog=1
sessname_File=record.log
```

sessname

Specifies the name of the session you wish to record

There is a sample OICTNET.INI provided in the %AP_HOME%\Sample\config folder.

2. Save the OICTNET.INI file in the %AP_SITE%\Config folder.
3. Restart CA Automation Point and run session *sessname* as usual.

All TN3270 communication traffic to the specified session will be recorded in a file named record.log in the %AP_DATA%\Logs directory.

Index

2

- 2-way paging • 235
 - component privileges • 19

A

- access permissions
 - defining for a session • 118
 - for AS/400 alerts • 415
 - MS Windows security • 116, 117
 - Remote Viewer • 115, 116
- action messages • 51, 55
- ADDRESS TNG command environment • 351
- ALARMSAY rules keyword • 232
- alerts, AS/400 • 409
- aliases, creating in Notification Manager • 279
- ANSWER TREE • 226
- AP Listener • 232
- AP Log Window • 78, 105
- AP Notification Messages window • 78, 106
- AP Speak program • 232
- Apache Tomcat Server, configuring • 267
- APCMOSI.REX • 43, 333
- application files • 182
- APVIEW/FONTS utility program • 97
- AS/400 Manager
 - commands • 425
 - environmental variables • 419
 - managing multiple systems • 414
- asynchronous sessions • 61
- Automation Point Desktop • 75
- automation programs, AS/400 • 435
- Automation Setup Wizard • 24
- AXC command environment • 186
- AXCREXX program • 186
- AXCREXX session, creating • 39

B

- backup methods, Notification Manager • 276
- broadcasting with Notification Manager • 281
- browsing recalled messages • 103

C

- CA AP Remote Manager • 116
- CA NSM • 351

- keywords for • 154
- CA OPS/MVS
 - defining nodes • 397
- CA OPS/MVS interface
 - configuring • 389
 - recycling • 398
 - stopping and starting • 398
- CA Remote Console, mainframe communication using • 50
- CA-AP NSM Gateway Service • 361
- CAICCI (CA Common Communications Interface) • 389
- call progress analysis
 - customizing • 211
 - parameters • 220
- call-in feature, configuring • 269
- changing the font selection list • 95, 97
- changing window • 81
- channel groups • 223, 229
- character strings, specifying in rules • 167
- clocks, synchronizing • 45
- CMOS processors • 333
- command area • 99
- command dialog box, issuing commands with • 127
- command environments
 - ADDRESS TNG • 351
 - AXC • 186
 - GLV • 190, 194
 - VOX • 237
- command rules • 176
- command summary, Notification Manager • 292
- commands
 - AS/400 Manager • 425
 - GLV environment • 194
 - issuing CA Automation Point • 98, 127
 - PPQ • 204
 - recalling previously entered • 101
 - VOX • 237, 238
- configuration information, AS/400 • 427
- Configuration Manager • 23
- configuration settings, importing and exporting • 32
- configuring
 - Answer Tree • 226
 - AS/400 Manager • 406, 417
 - CA Automation Point • 19

- CA OPS/MVS • 392
- call progress analysis parameters • 220
- channel groups • 229
- Dialogic voice card • 218
- HAF • 217, 343
- mail feature • 231
- Notification Manager • 265
- REXX • 181
- security for Event Traffic Controller • 375
- SSL • 42
- visual attributes • 142
- connections, mainframe and workstation • 402
- console commands • 100
- console types • 47
 - asynchronous • 72
- contact Notification Manager object • 262
- CPA parameters • 220
- CPC • 211
- CTS signal • 68
- customizing
 - keyboard parameter files • 457, 467
 - scan code files • 471, 472
 - screen fonts • 94
 - special CA Automation Point files • 457

D

- date and time, synchronizing with mainframe • 45
- deleting recalled messages • 103
- diagnostic information • 106
- Dialogic
 - installing software • 219
 - installing voice card • 218
- directory structure • 28
- display, keywords for controlling • 152
- dynamic status variables • 164

E

- environmental variables • 48, 157
 - environmental variables--AS/400 • 419
- error recovery, AS/400 • 430
- escalation, Notification Manager • 281
- Event Manager messages • 371
- event monitoring sessions, creating external • 40
- Event Traffic Controller • 361
 - configuring • 365
 - keywords • 154
- exporting configuration settings • 32

F

- file structure • 28
- fonts
 - and window size • 81
 - changing message fonts in WebMV • 142
 - customizing screen • 94
 - window menu options • 79
- forwarding
 - messages to CA NSM • 366
 - notifications • 278
- function windows, CA Automation Point • 78
 - enabling • 38

G

- GLV command environment • 190, 194
- groups
 - access to CA Automation Point workstation • 118
 - groups--configuring channel (notification server)
 - 229
 - Notification Manager • 280

H

- HAF (HMC Access Facility) • 343
- Hardware Management Console (HMC) • 43
- HMC (Hardware Management Console)
 - API • 333
- Host Log window • 78

I

- Import Sessions Utility • 37
- importing configuration settings • 31, 32
 - Export and Import utility • 31, 32, 34
- initialization scripts • 475
- IPL automation • 44
- IPL process, automating • 43

J

- Java Servlet environment • 267
- JES3 consoles • 51
- job status, automate checking of • 424
 - KEY keyboard parameter • 458
- JolBeep panel emulation • 330
- JRE (Java Runtime Environment), installing • 133

K

- KEY keyboard parameter • 458
- keyboard mapping

- for AS/400 • 409
- for terminal operations • 458
- keyboard parameter files • 457, 467
- keywords, rules • 150
- keywords, used for AS/400 • 420

L

- languages, user-specified in AS/ • 401
- layouts • 76
 - and Remote Viewer • 78
- LDAP
 - authentication • 310
 - permissions • 274
 - resources • 326
- log files
 - displaying • 105
 - Notification Manager • 327
- login Notification Manager object • 262

M

- M • 215
- mainframe host
 - automating the IPL process • 43
 - communicating with • 41
 - connecting to CA Automation Point • 392
 - connecting to using TN3270 • 41
- MAP keyboard parameter • 458
- MCS consoles • 49
 - screen example • 484
- memory requirements for PPQs • 202
- menu
 - customizing • 82
 - defining a new • 93
- menu options
 - adding new • 90
- MENU statement • 86
- Merged Msg window • 78, 104
- message format, HMC Access Facility • 344
- message handling • 50, 51, 52, 54, 55, 56
- message rules, writing • 176
- messages
 - color mapping with Web MV • 144
 - forwarding to CA NSM • 373
 - messages--displaying recalled • 101
 - Notification Manager and Notification Server • 106
 - OS/400 • 419
 - terminal emulator • 102

- viewing with Web MV • 131, 137
- method Notification Manager object • 262
- methods
 - backup • 276
 - defining to Notification Manager • 284

N

- NM objectClass definitions • 319
- NMPAGE2WAY method • 236, 292
- Normal Message Recall window • 78, 102
- notification • 27
 - notification--keywords for • 153
 - policies • 304, 307
- Notification Manager
 - command summary • 292
 - configuring • 265
 - database • 265
 - groups • 280
 - objects • 262
- notification server • 213
 - mail function • 231
 - options • 217
- Notification Setup wizard • 25
- notification schedules, guidelines • 293
- Notification Website
 - installing and configuring • 266, 268
 - troubleshooting • 307
 - using • 304
- null modem, switching signals with • 62

O

- objectClass, Notiification Manager • 317
- objects, Notification Manager • 262
- operator-initiated scripts • 476
- OS/400 commands • 425, 428
- OSA-ICC • 60
- OSA-ICC and MCS session automation • 60

P

- paging • 215
 - two-way • 235
- parameter Notification Manager object • 262
- parameters
 - APCMOSI.REX • 340
 - Call Process Analysis (CPA) • 220
 - import and export • 35
 - Notification Manager • 297, 298, 299, 300
 - scan code • 472

- pause scripts • 52, 475
- paused sessions, viewing in Web MV • 131
- Plot feature • 106, 441
 - deleting data from a graph • 453
 - sample statements • 455
- plot window • 78, 106
 - AS/400 system status • 427
- PPQs • 201
 - problem escalation, using the VOX environment for • 210
- primary control mode • 121

R

- RCS consoles • 47, 50
- Readlog • 74
- recalled messages, displaying • 101
- remote operations, starting • 124
- Remote Viewer • 112
- Remote_NM_Setup • 268
- requests, sending to CA NSM • 379
- RESULT variable (AS400JMP) • 424
- return codes
 - ACREXX • 185
 - BIND and UNBIND REXX programs • 429
 - from user-written Notification Manager code • 303
- REXX
 - and AS/400 automation • 424, 430
 - and the Plot feature • 443, 455
 - APCMOSI.REX • 339
 - AXCREXX sessions • 39
 - configuring and writing • 181
 - issuing PPQ commands from • 204
 - Notification Manager programs • 283
 - starting scripts from • 479
 - stopping a program • 101
 - TPF interface • 485
 - using Key operations in • 471
- rule files
 - creating • 156
 - sample • 486
- rules • 149
 - AXCRULES.rul • 486
 - issuing command processors from • 187
 - issuing PPQ commands from • 205
 - starting scripts from • 479
 - using key operations in • 471
 - writing for AS/400 automation • 419

- rules keywords • 150
 - rules keywords, used for AS/400 • 420

S

- scan code files, customizng • 471, 472
- scan code parameters • 472
- schedules, creating with Notification Manager • 293
- screen fonts, customizng • 94
- scripts
 - operator-initiated • 476
 - VSE console • 54
- scrolling
 - CA Automation Point windows • 82
 - in log files • 105
- security
 - and CA OPS/MVS interface • 123
 - and Notification Website • 309
 - Event Traffic Controller • 375
 - WebMV • 134
- session definition sets, managing • 37
- shedules, guidelines for notification • 293
- single sign-on • 120
- SMTP, configuring mail to use • 231
- special CA Automation Point files, customizng • 457
- SSH • 65
- SSL, configuring • 42
- starting CA Automation point
 - remotely • 124
- status line
 - asynchronous console session • 68
 - TN3270 session • 489
 - TN5250 session • 490
- status variables
 - AS/400 Manager • 421
 - dynamic • 164
 - in rules • 162
 - referencing from REXX • 189
- stopping CA Automation Point • 107
- submenus
 - adding • 92
- SYSPLEX consoles • 47, 48

T

- TAPI support • 216
- telnet sessions, asynchronous • 64
- terminal emulators • 104
 - identifying to a session • 473
- time block Notification Manager object • 262

TN3270

- connecting to mainframe using • 41
- session status line • 489

TN5250

- connecting to mid-range host • 42
- session status line • 490

TPF interface • 485

troubleshooting

- Dialogic card and software • 220
- Event Traffic Controller • 380
- Notification Website • 307
- text-to-speech notification • 234

two-way paging • 235

U

user rights, Event Traffic Controller • 375

V

voice card, installing and configuring • 218

VOX command environment • 237

W

Web Message Viewer • 112, 129

Window menu options, CA Automation Point • 79

windows, CA Automation Point Desktop • 75

Z

z/OS • 58