CA Integrated Agent Services

Implementation Guide

Version 12.0.00

Second Edition

CA Technologies Product References

This document references the following CA Technologies products:

- CA Workload Automation CA 7® Edition, (CA WA CA 7 Edition), formerly CA Workload Automation SE and CA 7® Workload Automation
- CA ACF2™
- CA Chorus™ Software Manager (CA CSM)
- CA Chorus™
- CA Integrated Agent Services (CA IAS)
- CA Mainframe Software Manager (CA MSM)
- CA Top Secret®

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 update has been made since the last release of this documentation:

- AGENT Statement (see page 27)—The *name* keyword can now include dashes (-).

Contents

Chapter 1: Introduction

This guide describes how to implement CA Integrated Agent Services (CA IAS). This guide is written for the system programmers and personnel responsible for the installation, implementation, and maintenance of CA IAS.

This section contains the following topics:

Overview

CA Integrated Agent Services (CA IAS) is a component of your scheduling manager and not a stand-alone system. CA IAS operates under the CA WA CA 7 Edition scheduling manager. The scheduling manager drives requests to CA IAS, and CA IAS passes responses to the scheduling manager. The requests and responses are divided into two categories:

- Requests for CA IAS to process, and CA IAS responds with the results.
- Requests that CA IAS sends through the TCP/IP network to a CA WA agent, and the responses from the CA WA agents that CA IAS passes through to the scheduling manager.

CA IAS initializes on the scheduling system's main task and then establishes its own set of subtasks to handle the functions it is to perform. CA IAS functions include, on a broad scale, the following:

- Establishing connections to the agents listed in an agent definition file

  CA IAS *shakes hands* with all agents listed in the agent definition file and helps ensure that a proper connection can be established for when a job is submitted that is destined for that agent. This includes encrypting the data and helping ensure that the data can be decrypted and recognized as an *agent handshake*.

- Listening for incoming messages that are sent from the agents

  CA IAS establishes a *listening port* in which incoming messages are received. Agents use this port to send messages to the scheduling manager. The port specification is part of the agent definition file. If the message is not involved in the hand shake process, CA IAS sends the message to the scheduling manager for processing.

- Building and decoding messages

  Messages sent to and from agents are formatted into an Automated Framework Message (AFM) so that an agent and scheduling managers can understand the message and can perform the correct action. CA IAS takes a common Control Language (C-LANG) and builds the AFM message to send to the agent. CA IAS also receives the AFM messages from the agent. Upon receipt, CA IAS parses the incoming AFM and calculates lengths and displacements to the various keywords. This parsing lets the scheduling manager access the data directly instead of trying to interpret the format of the response message.

- Queuing messages to send to the agents, which include job executions and commands

  The scheduling manager determines when to submit jobs for execution. If the scheduling manager detects the job is an agent job, it invokes CA IAS to build the AFM message based on the input C-LANG. If the message is built successfully, the scheduling manager invokes CA IAS to *send* the message to the destination (agent). CA IAS queues this message, because the destination agent may not be immediately active. CA IAS may then immediately send the message, or if the agent is not available, CA IAS keeps the message in its queues.

  For commands, a similar process is used, but command messages are not queued. If the message cannot be sent immediately, CA IAS responds to the scheduling manager that the command cannot be sent.

- Storing messages received from the agents until the scheduling manager asks for the next response

  CA IAS receives the messages from agents and queues them until the scheduling manager can pick up the response. When the scheduling manager is ready, it asks CA IAS to return a response message. One or more messages may be ready for processing, and CA IAS returns one response at a time, using the highest priority or the oldest message.

- Encrypting and decrypting the data sent to and from the agents

  Messages going to or coming from agents are encrypted using the Advanced Encryption Services (AES) algorithm. CA IAS handles the encryption for outbound messages and decryption for inbound messages. The encryption key to use is defined in the Encryption Table file and stored by CA IAS.

- Storing information about user IDs and passwords

  CA IAS associates passwords with user IDs as the message is sent to the agent. The scheduling manager has its own user interface in which to define the password to use, and this information is passed to CA IAS. The password to associate with a user ID is a table lookup, which is processed based on the following:

  - User ID
  - Agent (destination)
  - Job type
  - Source (only valid for selected job types)

  For example, you could use PSWDA for USERIDA going to AGENTA, and PSWDB for USERIDA for all other agents. You can also specify to use PSWDT for USERIDA if the job type is an FTP job type (FTP_JOB). You should plan carefully how to associate the password to use with the user ID.

- Creating trace records indicated by an appropriate level

  CA IAS has an internal tracing facility that CA Support can request to aid in debugging issues. The scheduling manager has the user interface to set the trace settings and extract the data.

- Performing actions based on command input, such as reconfiguring the agents, stopping, or starting the TCP/IP communications

  CA IAS has its own *commands* to perform functions, such as stopping and starting the TCP/IP communications interface or reconfiguring the agents after an update is made to the agent definition file. Because the user interface is part of the scheduling manager, see the CA WA CA 7 Edition documentation for more information.

Working with Agents

CA IAS is a component of the scheduling manager's address space that facilitates communication with the CA WA agents. The CA WA agents are a suite of products designed to execute on different platforms and with different business applications. The agent executes on the Windows or UNIX operating environment and can support different plug-ins for additional support. For example, using agent plug-ins, agents can interface with applications such as various databases, SAP, Oracle, and PeopleSoft. The agent plug-ins are separate products from the agent that runs on the Windows or UNIX operating environment.

Note: For more information about these CA WA agents, see their documentation sets.

With all these possibilities that agents supply, you can define various job types. A unique name that ends with _JOB usually identifies these types. The NT_JOB job type is for Windows platforms. You can use the UNIX_JOB job type for generic UNIX platforms. You can use the FTP_JOB job type is to transfer a file from one location to another through an FTP server. With J2EE, you can use the JMSS_JOB job type to subscribe to a Java subscription service. With databases, a stored procedure can be executed through a DBSP_JOB job type.

Note: For more information about supported job types, see the *CA Integrated Agent Services User Guide*.

The scheduling manager uses Control Language or C-LANG to instruct the agent what to execute. C-LANG statements include items such as the command (script) to execute, the service which should be subscribed to, the file to transfer, and more. Each job type has its own set of C-LANG statements that should be coded. Some statements are required and some statements are optional, again depending on the job type. The C-LANG statements are then used to build the message that is sent to the agent. For a simple example, a Windows job to start an interactive calculator would have the following C-LANG statements:

```
CMDNAME C:\WINDOWS\SYSTEM32\CALC.EXE
INTERACTIVE YES
```

A File Trigger job could have the following single C-LANG statement:

```
FILENAME 'C:\MY DOCUMENTS\EROC DATA\TEST1.TXT' UPDATE
```

Note: For more information about C-LANG statements, including the syntax rules, see the *CA Integrated Agent Services User Guide*.

When the scheduling manager schedules and submits the job, the message is built from the C-LANG statements and sent to the destination (agent). CA IAS may queue the message so that the job is waiting for the transmission (similar to a job waiting in the JES input queue). CA IAS checks agent availability and sends the message when it can. The CA WA agent starts the execution of the job and sends CA IAS a message indicating that the job has started (similar to a job initiation signal). When the job completes its function, the agent sends a message to CA IAS indicating the results of the execution (similar to a job termination signal).

The agent determines the success or failure of the job execution. The scheduling manager does not determine the success or failure of an agent job. This determination is different from z/OS jobs executing in the scheduling manager, where the user can inform the scheduling manager what condition codes to consider as failure. For jobs executing on an agent, the agent determines the success or failure and sends an appropriate message to the scheduling manager. Users can code an EXITCODE C-LANG statement to influence the outcome of success or failure, but this statement only applies with selected jobs types.

Note: For more information about the EXITCODE statement, see the *CA Integrated Agent Services User Guide*.

For selected job types, there is a concept of continuous jobs; a monitoring type of job is an example. These jobs execute for the duration of the time that the agent is active. When the scheduling manager sends the job over, the agent acknowledges the continuous nature. When the condition set by the monitoring job type is met, a signal is sent that the condition has been met, with an alert name that the scheduling manager should execute. The monitoring job is active until an appropriate action removes it or the agent is made inactive.

For example, a CPU_MON job type is defined to monitor the CPU usage on the agent machine. The definition states that when CPU usage reaches 90 percent, execute alert HIGHCPU. The scheduling system sends the job over at the determined time. The agent then sets up a monitor alert and when the CPU reaches 90 percent, it sends a message to the scheduling manager to execute alert HIGHCPU. The agent continues monitoring the CPU, and when the usage reaches 90 percent again, it sends another message back to the scheduling manager, until the job is removed from the agent.

Note: Not all scheduling managers support a continuous job type, but they support a monitor occurrence (one-time alert). For more information, see the scheduling manager's documentation.

Chapter 2: System Requirements

This section contains the following topics:

Operating System

CA IAS operates under all levels of the z/OS operating system that IBM currently supports.

CA IAS requires an interface to the TCP/IP component of the z/OS operating system.

CA IAS should execute APF authorized in the standard problem program protect key of the scheduling system manager.

CA IAS installation and maintenance require SMP/E.

To use AES256 encryption, CA IAS requires an active IBM z/OS Integrated Cryptographic Services Facility (ICSF) system whose libraries are available in the linklist or in the STEPLIB.

Note: To comply with FIPS 140-2, a hardware coprocessor must be available. CA IAS does not operate with a software or CPACF feature. CA IAS verifies that the AES 256-bit encryption algorithm, a coprocessor, and secure master key are available.

Hardware

CA IAS is installed onto currently supported Direct-Access Storage Devices (DASD). For example, supported DASD includes 3390 and 9345 devices.

To use the hardware version of AES256 encryption, CA IAS requires an active IBM z/OS ICSF system with appropriate coprocessors for the encryption services.

DASD Requirements

CA IAS requires DASD space allocation. The three classes of data sets are the following: SMP/E distribution libraries, SMP/E target libraries, and CA IAS permanent files.

Distribution Libraries

CA IAS is installed using SMP/E. The SMP/E distribution libraries include the following:

AIASDATA

CA IAS data library

AIASMOD

CA IAS load library

AIASSAMP

CA IAS sample library

Target Libraries

The SMP/E target libraries include the following:

CIASOPTN

CA IAS sample library

CIASLOAD

CA IAS load library

CIASJCL

CA IAS sample JCL library

Permanent Files

In addition to the SMP/E controlled data sets, CA IAS requires two physical sequential files (or two members of a Partitioned Data Set (PDS)) and one VSAM Data-in-Virtual (DIV) data set.

Agent Definition File

The agent definition file is a physical sequential file (or a member of a PDS) that describes the scheduling manager and the agents to which the manager can send jobs. This file also names the encryption key name (not the actual key) that should be used to communicate with the agent.

The scheduling manager is identified by the MANAGER statement, which names the manager that the agents recognize. The agents use this manager name in their definitions to direct responses back to the manager. Each scheduling system manager must have a unique name. On the AGENTRCV (agent receiver) statement, the TCP/IP receiver port is identified. The following is an example:

```
MANAGER NAME(MANAGER_NAME)
AGENTRCV RECEIVER1 PORT(4701)
```

The agents are defined to the manager so that jobs can be routed to that agent. If the agent is not defined to the scheduling manager, no jobs can be routed to that agent. In the agent definition, the agent name, address (either the domain name or the physical address), and the receiver port identify where in the TCP/IP network the agent can be found and on what port this agent listens for incoming messages. You can provide either a domain name or a physical address. If a domain name is provided, CA IAS invokes the Domain Name Search service to resolve the address, which may cause unnecessary overhead if the physical address never changes.

Also, the agent should identify the scheduling manager and the encryption name/key in its agent parameter (agentparm.txt) file.

Note: For more information about the agent parameter file, see the Implementation Guide for the appropriate agent.

The agent definition must specify other information. Specify the language that the agent understands, either EBCDIC or ASCII, so that CA IAS translates the data appropriately. Specify the retry parameters so that the scheduling manager continues to retry based on intervals if it cannot communicate with the agent. The RETRYINTERVAL specifies the number of milliseconds before CA IAS tries to resend a message. If after the RETRYCOUNT number of times CA IAS still cannot send a message, CA IAS puts that agent into a sleep interval based on the SLEEPTIME number of seconds.

The CRYPTNAME parameter names an entry in the encryption table where the encryption key is specified. This name is a different CA IAS file that you can secure for appropriate personnel using your security system.

The following is an example of an agent definition. You can define one or more agents defined to the scheduling manager.

```
AGENT SYSAGENT +
      PLATFORM(NT) ASCII CRYPTNAME(KEYT1) +
      RETRYINTERVAL(60000) RETRYCOUNT(5) SLEEPTIME(3000) +
            ADDRESS(142.242.290.211) PORT(7520)
```

More information:

AGENT Statement (see page 27)

Encryption Table

The encryption table lists the names and keys for the encryption algorithms. These algorithms use 16-byte or 32-byte keys to transform the data from readable text into data that is not easily discernible before transmission over a TCP/IP network. All data sent to or from an agent is processed with an encryption algorithm. Multiple agents can use a name in the agent definition file, but the name can have only one key value in the encryption table. Each key value can be associated with one or more names.

The key value is also provided to the agent on the agent platform. This value is a secured item. Protect the key value and give it to the agent installer in a secured fashion.

The following statement is a sample from the encryption table. CRYPTNAME is the statement. The NAME keyword identifies the key name that is used in the agent definition file. The KEY is the actual key value that is used in the encryption algorithm. TYPE(AES) is the encryption algorithm name and is the default value.

```
CRYPTNAME NAME(KEYT1) KEY(01020304050607089AABBCCDDEEFFAD) TYPE(AES)
```

More information:

Encryption Table File (see page 28)

Communication Queue

The communication queue is a Data-in-Virtual (DIV) VSAM data set that CA IAS uses to store information for its use. The information includes the messages sent and received from agents, the agent and encryption definitions, and passwords associated with user IDs sent to the agents. Allocate this file through JCL before starting a scheduling manager with CA IAS.

The size of this file depends on the following:

- the number of agents defined to the scheduling manager
- the number of jobs being sent to the agents
- the number of passwords to define for these jobs

The general recommendation is to allocate this file initially at about ten primary cylinders and two secondary cylinders. As a guideline, use the following:

- Each agent requires about 500 bytes of storage.
- Each job requires 100 bytes + size of the AFM (AFMs vary in size according to job types and number of parameters being passed).
- System overhead uses approximately 1000 bytes.
- Each password entry requires 350 bytes.

For 25 agents, 250 average jobs, and 50 passwords, the approximate space requirement would be about 8 percent of 2 cylinders (1,474,560 bytes). Bring down the scheduling manager while the communication queue data set is moved if space is exhausted.

Sample JCL to allocate this file is provided in the CIASJCL data set as member IASCKPT.

Memory

CA IAS executes in part of a scheduling manager's address space. In addition to the memory requirements for the scheduling manager, CA IAS requires an additional 320 KB for programs, in addition to the memory required for the agent, job, and password definitions.

CA IAS uses 31-bit and 64-bit mode storage. If your installation has restrictions about using 64-bit memory, the scheduling manager execution JCL may need to indicate MEMLIMIT=10M, as the CA IAS logging storage is in 64-bit storage. The default storage allocation for logging is 10 MB, but you can adjust the allocation at the direction of CA Support when pursuing a problem.

Scheduling Managers

CA IAS executes within the CA WA CA 7 Edition scheduling manager's address spaces.

Review the appropriate CA WA CA 7 Edition documentation to see how to activate CA IAS within the address space. There are options to activate the CA IAS interface, DD statements to be coded for CA IAS (such as for the Agent Definition file), and special commands to interface with CA IAS. Because CA IAS interfaces with the CA WA agents, install one or more of the following agents on its appropriate platform. The CA WA agents include the following agents:

- CA Workload Automation Agent for Databases
- CA Workload Automation Agent for HP Integrity NonStop
- CA Workload Automation Agent for i5/OS
- CA Workload Automation Agent for Linux
- CA Workload Automation Agent for Oracle E-Business Suite
- CA Workload Automation Agent for PeopleSoft
- CA Workload Automation Agent for SAP
- CA Workload Automation Agent for UNIX
- CA Workload Automation Agent for Windows
- CA Workload Automation Agent for z/Linux
- CA Workload Automation for Application Services
- CA Workload Automation for Web Services

Security Requirements for AES256

Using the AES256 encryption option invokes the IBM ICSF. The ICSF enciphers and deciphers the messages that are sent and received between the scheduling manager and the CA WA Agent. Some customers use a security package to control access to ICSF, such as CA Top Secret, CA ACF2, or IBM RACF. In this case, grant appropriate access to the scheduling manager on behalf of CA IAS.

Note: For more information about ICSF security details, see the *IBM ICSF Administrator's Guide*.

Two classes are associated with ICSF Security. CA IAS, through the scheduling manager, requires only read access to the callable services, named CSFSERV.

Within the CSFSERV callable services class, CA IAS uses only the following ICSF callable services:

CSFIQA

ICSF Query Algorithm callable service

CSFCKM

Multiple clear key import callable service

CSNBSAD

Symmetric Algorithm Decipher

CSNBSAE

Symmetric Algorithm Encipher

Many items require consideration when dealing with ICSF security implementation. Depending on the site security system that you use, consult one of the following appropriate guides:

- CA Top Secret: *Command Functions Guide*
- CA ACF2: *Administration Guide*
- IBM RACF: *IBM Security Server RACF Security Administrator's Guide*

Chapter 3: Implementation

CA IAS is packaged with the scheduling manager because it cannot execute on its own. As a result, follow the installation guide of the scheduling manager. After the scheduling manager is installed, the CA IAS function is usually also present. Verify through the SMP/E dialog that function (SYSMOD) CIASC00 is installed.

After you verify the function, ensure that the latest maintenance is applied so that you do not incur any problems that are already resolved. If you are using CA CSM, log in and request that the maintenance is downloaded and applied for CA IAS.

If you received the scheduling manager package through the Electronic Software Download (ESD) process or through a tape, log in to CA Support Online. Request the download for the maintenance applicable to CIASC00, the CA IAS function identifier.

Before starting the scheduling manager with CA IAS activated, verify that the CA IAS module library, CIASLOAD, is APF authorized in your system. To authorize the data set, add it to the IEAAPF*xx* or PROG*xx* members in SYS1.PARMLIB. If you use the IEAAPF*xx* member, an IPL is required.

Chapter 4: Configuration

This section contains the following topics:

Syntax Rules

The statements in the agent definition and encryption table files are 80-byte records found in a physically sequential file or as members of a Partitioned Data Set (PDS). These statements have syntax rules so that CA IAS can parse them correctly.

- Each statement should begin on a new line in column 1.
- Columns 73 through 80 are ignored and can contain sequence numbers if preferred.
- A continuation character, the plus sign (+), must occur on or before column 71.
- The continued statement must begin beyond column 1.
- Comments are added by placing an asterisk (*) in column 1 before the beginning or after the completion of a statement (MANAGER, AGENRCV, or AGENT). In other words, the comment cannot be embedded into the parameters that continue from a previous line.

The agent definition file also contains statements naming the scheduling manager and its listening port. You can choose to separate the manager name from the agent definitions and concatenate the two files together.

The encryption table is a separate file. Restrict the file to those users who need to know the encryption keys. You can secure access to this file using the installation's security system, CA Top Secret, or CA ACF2, for example.

Agent Definition File

The agent definition file identifies the scheduling manager and the CA WA agents with which to establish communication. The SMP/E controlled CIASOPTN(IASAGENT) contains a sample member of an agent definition file.

The following are the attributes of the agent definition file:

DDNAME

Defined in the JCL of the scheduling manager. Code the name as IASAGENT.

Attributes

RECFM=FB, LRECL=80. Only bytes 1-71 are used (allowing for sequence numbers if PDS member).

Contents

Manager specifications and agent definitions.

The agent definition file has the following three main statements:

- MANAGER
- AGENTRCV
- AGENTDEF

MANAGER Statement

The MANAGER statement names the scheduling manager and must be unique across the *agent network*. Agents recognize this name and have it in their definitions. The agent definition file of the scheduling manager can contain only one MANAGER statement.

This statement has the following format:

```
MANAGER NAME(name) [ADDAGENT(nnnnn)]+

RETRYINTERVAL(nnnnnn) RETRYCOUNT(nnnnnn) SLEEPTIME(nnnnnn)
```

MANAGER

Specifies the required keyword to define a scheduling manager.

NAME(*name*)

Identifies the 1- to 16-byte required name of this scheduling system manager. The name is alphanumeric characters and must begin with an alphabetic character. The name can include an underscore (_) character and cannot include spaces. This name is used in defining the manager on the appropriate agents.

ADDAGENT(*nnnnn*)

(Optional) Specifies the number of additional agent definitions that can be added through reconfiguration commands that are entered on the scheduling manager after a CA IAS initialization.

Default: 5

Limits: Maximum is 10000 minus the number of agent definitions currently in the IASAGENT DD.

Note: The maximum value that is permitted for ADDAGENT is 10000. A system that permits many agent definitions typically sees a noticeable performance degradation for many CA IAS-related functions. These functions include sending jobs for execution and receiving job feedback. If you have a stable agent definition environment, keep the default of 5. If, however, adding many agents between recycles of CA WA CA 7 Edition is required, we recommend that you place a reasonable value in the ADDAGENT parameter so that the performance of CA IAS and the scheduling manager is not severely impacted.

Each of the following parameters is used as a default for any agents that do not specify them explicitly. If they are not specified at the manager level, the system default applies.

RETRYINTERVAL(*nnnnnn*)

Specifies the number of milliseconds between each retry of a send message function to the agent.

Default: 30000 (30 seconds)

Limits: 0 means no retry, maximum is 999999

RETRYCOUNT(*nnnnnn*)

Specifies the number of retries for all RETRYINTERVAL parameters for a send message function after which the agent is placed in sleep mode.

Default: 3

Limits: 0 means no retry, maximum is 999999

SLEEPTIME(*nnnnnn*)

Specifies the number of seconds before another attempt is made to connect to the agent.

Default: 900 (15 minutes)

Limits: maximum is 999999

AGENTRCV Statement

The AGENTRCV statement identifies the listening port on which the manager listens for incoming messages from agents. The agent definition file can contain only one AGENTRCV statement.

This statement has the following format:

```
AGENTRCV name PORT(nnnnn)
```

AGENTRCV

Specifies the required keyword to define a listening port for the scheduling manager.

name

Specifies the required name of the listening port. This name field is positional and is 1- to 16-alphanumeric characters and must begin with an alphabetic character.

PORT(*nnnnn*)

Specifies the port in which the scheduling manager listens for incoming messages. The number can range from 1-65535 and should be unique in the port numbers used in the z/OS operating system where the scheduling manager executes. Specify this port number in the agent that connects to this scheduling manager. This required parameter has no default.

AGENT Statement

The AGENT statement defines each agent to which this scheduling manager can send jobs. If the agent is not defined here, the scheduling manager receives a message that the job cannot be sent because the agent is not defined. Each agent requires one AGENT statement. Specify as many AGENT statements as necessary to define the agents where you are sending jobs.

This statement has the following format:

```
AGENT name ADDRESS(IP address or DNS Name) +

PORT(nnnnn) +

PLATFORM(type) +

CRYPTNAME(encryption key name) +

RETRYINTERVAL(nnnnnnn) RETRYCOUNT(nnnnnn) SLEEPTIME(nnnnnn) +

{ASCII|EBCDIC}
```

AGENT

Specifies the required keyword to start a new agent definition.

name

Specifies the required name of the agent (listener). The name can range from 1- to 16-alphanumeric characters and must begin with an alphabetic character. The name can include an underscore (_) and dash (-) character and cannot include spaces. Job definitions use this name to route the job's execution to the appropriate agent. Identify each agent uniquely in the network. Duplicate agent names, even though at different hosts, are not permitted in the agent definition for a manager.

ADDRESS(*IP Address*|*DNS Name*)

Specifies the IPv4 or IPv6 address or DNS name for this agent. This keyword is required. Using a DNS name requires extra overhead to resolve the name into an IP Address. The maximum length is 100 bytes.

PORT(*nnnnn*)

Specifies the required IP port on which the agent listens for incoming requests. The number can range 1-65535.

PLATFORM(*type*)

Specifies the optional operating environment on which the agent executes. The following values are valid:

UNIX (generic term for UNIX and Linux operating systems)

NT (generic term for Windows operating systems)

AS400 (i5/OS operating systems)

TANDEM (HP Integrity NonStop operating systems)

CRYPTNAME(*name*)

Specifies the required name of the key found in the encryption table file. Multiple agents can use the same name, or each agent can have its own encryption key name.

RETRYINTERVAL(*nnnnnn*)

Specifies the number of milliseconds between each retry of a send message function to the agent.

Default: Defers to the MANAGER statement

Limits: 0 means no retry, maximum is 999999

RETRYCOUNT(*nnnnnn*)

Specifies the number of retries for all RETRYINTERVAL parameters for a send message function after which the agent is placed in sleep mode.

Default: Defers to the MANAGER statement

Limits: 0 means no retry, maximum is 999999

SLEEPTIME(*nnnnnn*)

Specifies the number of seconds before another attempt is made to connect to the agent.

Default: Defers to the MANAGER statement

Limits: Maximum is 999999

{ASCII|EBCDIC}

Specifies the character set in which to transmit data to this agent. The default is ASCII.

Encryption Table File

The administrator defines the actual encryption keys to use when encrypting and decrypting messages to or from agents. This information is kept in a separate file so that a limited number of people can access the file as required. You can find a sample member of an encryption table file in the SMP/E-controlled CIASOPTN(IASCRYPT).

This file has the following attributes:

DDNAME

Defined in the JCL of the scheduling manager. Code the name as IASCRYPT.

Attributes

RECFM=FB, LRECL=80. Only bytes 1-71 are used (allowing for sequence numbers for a PDS member).

Contents

Key name and encryption algorithm and key.

The encryption name is listed in the agent definition file. Many agents can use the same encryption name, or each agent can have a unique encryption name. The file consists of one or more of the following statements:

```
CRYPTNAME NAME(name) KEY(0102030405060708090A0B0C0D0E0F00) TYPE(aaaaaa)
```

CRYPTNAME

Specifies the keyword identifying the beginning of an encryption key definition.

NAME(*name*)

Specifies the name that is associated with the encryption key. The name can range from 1-16-alphanumeric characters and must begin with an alphabetic character. This name is referenced in the AGENT definitions as CRYPTNAME(*name*).

KEY(*data*)

Specifies the 32 or 64 hexadecimal characters (0-9, A-F) that form one of the following keys:

- The 16-byte key to use in the AES encryption algorithm.
- The 32-byte key to use in the AES256 encryption algorithm.

This data must match the same key that is defined on the agent side. When the data matches, the scheduling system and the agent encrypt the data with the same key.

TYPE(AES|AES256|NONE)

Specifies the type of encryption used. AES and AES256 are the only valid options for almost all agents. Although the system agent can support multiple encryption types, the only supported types are AES and AES256. The default value is AES.

The only valid exception is for the CA WA Agent for HP Integrity NonStop, which does not currently support encryption. If you are using the HP Integrity NonStop agent, use a value of NONE instead.

Communication Queue

The communication queue, also referred to as the checkpoint file, maintains the data for use by CA IAS. This queue maintains a copy of the messages waiting to be sent to an agent, and keeps data sent from the agents until the scheduling manager can process the incoming messages. This process is used mainly across starts and stops of the scheduling manager.

The following are the attributes of this file:

DDNAME

Defined in the JCL of the scheduling manager. Code the name as IASCKPT.

Attributes

DIV data set (4K pages) backed by VSAM Linear.

Contents

Checkpoint data, message queues, and password information.

To allocate this data set, use the JCL member IASCKPT found in the CA IAS CIASJCL data set. The size depends on the number of agents defined and the number of jobs that are sent to these agents.

Chapter 5: Backup and Recovery Consideratons

This section contains the following topics:

Overview

The communication queue is a Data-In-Virtual (DIV) VSAM linear data set that CA IAS uses to store information for its use. The data consists of the following:

- Messages that agents are sending and receiving but are not yet acknowledged
- Agent and encryption definitions
- Password definitions

One backup job is provided. The job uses IDCAMS to create a physical sequential backup file.

Two reload jobs are provided. The first job is considered the *normal* reload, and it uses IDCAMS to reload all the data contained on a backup file. The second reload job reloads only password definitions. This second reload job is used in cases where you must reinitialize the communication queue but do not want to lose your password definition information.

Note: To run the backup job, either communications to IAS must be stopped by the scheduling manager or the scheduling manager online task must be down. To run the restore jobs, the scheduling manager online task must be down.

As part of installation, three sample jobs are provided for backup and restore of the communication queue. These jobs can be found in the CIASJCL data set.

- IASCKPBK – back up the DIV VSAM linear data set to a physical sequential file.
- IASCKPRL – reload the DIV VSAM linear data set from a backup file.
- IASCKPRP – reload the DIV VSAM linear data set from a backup file with only password definition information.

Communication Queue Backup Job - IASCKPBK

The sample backup job, IASCKPBK, is an execution of IDCAMS REPRO to back up the contents of the communication queue file to tape.

Job IASCKPBK

The following is a sample IASCKPBK job.

```
//IASCKPBK JOB ...
//*
//*********************************************************************
//*                                                                   *
//*  CA IAS BACKUP THE IAS CHECKPOINT FILE                            *
//*                                                                   *
//*  USE YOUR EDITOR'S CHANGE ALL COMMAND TO CHANGE THE FOLLOWING     *
//*  VALUES:                                                          *
//*                                                                   *
//*  'CAI.IAS'               PREFIX OF DATASET                        *
//*                                                                   *
//*  *** NOTE ***                                                     *
//*  *** NOTE *** TO RUN THIS JOB, EITHER COMMUNICATIONS TO IAS       *
//*  *** NOTE *** MUST BE STOPPED BY THE SCHEDULING MANAGER OR        *
//*  *** NOTE *** THE SCHEDULING MANAGER ONLINE TASK MUST BE DOWN.    *
//*  *** NOTE ***                                                     *
//*                                                                   *
//*********************************************************************
//*
//*-------------------------------------------------***
//* IDCAMS REPRO IASCKPT TO SEQUENTIAL FILE         ***
//*-------------------------------------------------***
//BACKUP   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//INPUT    DD DSN=CAI.IAS.IASCKPT,
//            DISP=SHR
//OUTPUT   DD DSN=CAI.IAS.IASCKPT.BACKUP,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,LABEL=(1,SL),
//            DCB=(RECFM=FB,LRECL=4096,BLKSIZE=24576)
//SYSIN    DD *
 REPRO INFILE(INPUT) OUTFILE(OUTPUT)
/*
```

IASCKPBK DD Statements

The BACKUP step has the following DD statements:

INPUT

Specifies the IAS Communication Queue data set.

OUTPUT

Identifies the sequential backup file.

SYSPRINT

Used for message output.

SYSIN

IDCAMS control statements.

Communication Queue Reload Job - IASCKPRL

The sample reload job, IASCKPRL, contains two IDCAMS steps. The first step deletes and redefines the communication queue. If you want to increase the size of the communication queue, adjust the space parameters on the DEFINE CLUSTER in this step before submitting the job. The second step copies all records from a backup tape to the communication queue.

Job IASCKPRL

The following is a sample IASCKPRL job.

```
//IASCKPRL JOB ...
//*
//*******************************************************************
//*                                                                 *
//*  CA IAS RELOAD CHECKPOINT FILE                                  *
//*                                                                 *
//*  USE YOUR EDITOR'S CHANGE ALL COMMAND TO CHANGE THE FOLLOWING   *
//*  VALUES.                                                        *
//*                                                                 *
//*  'CAI.IAS'               PREFIX OF DATASET                      *
//*  'PRIME'                 PRIMARY SIZE                           *
//*  'SEC'                   SECONDARY SIZE                         *
//*  'VOLUME'                TARGET VOLUME FOR DATASET              *
//*                                                                 *
//*                                                                 *
//*  *** NOTE ***                                                   *
//*  *** NOTE *** TO RUN THIS JOB, THE SCHEDULING MANAGER ONLINE    *
//*  *** NOTE *** TASK MUST BE DOWN.                                *
//*  *** NOTE ***                                                   *
//*******************************************************************
//*
//*-------------------------------------------------***
//* IDCAMS DELETE/DEFINE CHECKPOINT FILE             ***
//*-------------------------------------------------***
//INIT     EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
   DELETE (CAI.IAS.IASCKPT) CLUSTER
   DEFINE CLUSTER                          -
          (NAME(CAI.IAS.IASCKPT)           -
           CYLINDERS(PRIME SEC)            -
           LINEAR                          -
           VOLUMES(VOLUME))
/*
//*
```

```
//*-------------------------------------------------------***
//* IDCAMS REPRO FROM SEQUENTIAL BACKUP TO CHECKPOINT FILE ***
//*-------------------------------------------------------***
//RELOAD   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//INPUT    DD DSN=CAI.IAS.IASCKPT.BACKUP,
//            DISP=SHR,
//            UNIT=TAPE,LABEL=(1,SL)
//OUTPUT   DD DSN=CAI.IAS.IASCKPT,
//            DISP=SHR
//SYSIN    DD *
 REPRO INFILE(INPUT) OUTFILE(OUTPUT)
/*
```

IASCKPRL DD Statements

The INIT step has the following DD statements:

SYSPRINT

Used for message output.

SYSIN

IDCAMS control statements.

The RELOAD step has the following DD statements:

INPUT

Specifies the sequential file created by the backup job.

OUTPUT

Specifies the IAS Communication Queue data set.

SYSPRINT

Used for message output.

SYSIN

IDCAMS control statements.

Communication Queue Password Reload Job - IASCKPRP

The sample password reload job, IASCKPRP, contains two steps. The first step uses IDCAMS to delete and redefine the communication queue. The second step uses program CAIASDVP to copy only password records from a backup tape to the communication queue. An optional parameter on the second step specifies how much storage to acquire for internal use.

CAIASDVP PARM Value

The CAIASDVP PARM has the following value:

```
//stepname EXEC PGM=CAIASDVP,PARM=nnnK|nnnM
```

The PARM value specifies the size used on a storage obtain request. Make the area large enough to contain the entire input file specified by the DIVIN DD statement. The default is 15M.

If you do not specify a large enough value, you receive the following message:

CAIAS0207E Input file DIVIN size exceeds getmain value, increase PARM value

If this message occurs, Increase the PARM value and rerun.

***nnn*K**

Specifies kilobytes of storage. Valid values are 1K – 999K.

***nnn*M**

Specifies megabytes of storage. Valid values are 1M – 999M.

Job IASCKPRP

The following is a sample IASCKPRP job.

```
//IASCKPRP JOB ...
//*
//*********************************************************************
//*                                                                   *
//*  CA IAS RELOAD ONLY PASSWORD RECORDS TO CHECKPOINT FILE           *
//*                                                                   *
//*  USE YOUR EDITOR'S CHANGE ALL COMMAND TO CHANGE THE FOLLOWING     *
//*  VALUES:                                                          *
//*                                                                   *
//*  'CAI.IAS'               PREFIX OF DATASET                        *
//*  'PRIME'                 PRIMARY SIZE                             *
//*  'SEC'                   SECONDARY SIZE                           *
//*  'VOLUME'                TARGET VOLUME FOR DATASET                *
//*                                                                   *
//*  *** NOTE ***                                                     *
//*  *** NOTE *** TO RUN THIS JOB, THE SCHEDULING MANAGER ONLINE      *
//*  *** NOTE *** TASK MUST BE DOWN.                                  *
//*  *** NOTE ***                                                     *
//*                                                                   *
//*********************************************************************
//*
//*-------------------------------------------------***
//* IDCAMS DELETE/DEFINE CHECKPOINT FILE             ***
//*-------------------------------------------------***
//INIT     EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
   DELETE (CAI.IAS.IASCKPT) CLUSTER
   DEFINE CLUSTER                          -
          (NAME(CAI.IAS.IASCKPT)           -
           CYLINDERS(PRIME SEC)            -
           LINEAR                          -
           VOLUMES(VOLUME))
/*
//*
```

```
//*-------------------------------------------------------***
//* RELOAD ONLY USERID/PASSWORD RECORDS TO CHECKPOINT FILE ***
//*-------------------------------------------------------***
//PWLOAD  EXEC PGM=CAIASDVP
//STEPLIB  DD  DISP=SHR,DSN=CAI.IAS.CIASLOAD
//SYSPRINT DD SYSOUT=*
//DIVIN    DD DSN=CAI.IAS.IASCKPT.BACKUP,
//            DISP=SHR,
//            UNIT=TAPE,LABEL=(1,SL)
//DIVOUT   DD DSN=CAI.IAS.IASCKPT,
//            DISP=SHR
```

IASCKPRP DD Statements

The INIT step has the following DD statements:

SYSPRINT

Used for message output.

SYSIN

IDCAMS control statements.

The PWLOAD step has the following DD statements:

DIVIN

Specifies the sequential file created by the backup job.

DIVOUT

Specifies the IAS Communication Queue data set.