CA Workload Automation System Agent for z/OS

Command Reference Guide

Release 11.4

CA Technologies Product References

This document references the following CA Technologies products:

- CA Workload Automation System Agent for z/OS (z/OS Agent)
- CA Workload Automation ESP Edition (CA WA ESP Edition)
- CA Workload Automation CA 7® Edition (CA WA CA 7 Edition)
- CA Workload Automation DE
- CA Workload Automation AE
- CA Workload Automation Restart Option ESP Edition (CA WA Restart Option ESP Edition)
- CA Workload Automation Service Governor ESP Edition (CA WA Service Governor ESP Edition)
- CA Workload Automation High Availability ESP Edition (CA WA High Availability ESP Edition)

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.

Contents

Chapter 1: About This Guide 7

Overview 7
Issue Commands 7
Entering Statements 7
    Continuation 7
    Comments 8
    Data Sets 8
    Delimiters 8
    Indentation 8
Wildcards in z/OS Agent 8
Wildcards in CA WA Restart Option ESP Edition 9
Examples 11
    Issue Console Commands 11
    Issue OPER Commands 11
    Issue Commands in Batch 12
    System Name 12
    Steplib 12
Conventions Used in This Guide 12
    Input Convention 12
    Syntax Conventions 12

Chapter 2: Commands 15

AGENTMSG Command: Send Messages to a Scheduling Manager 16
APPLJOB(AJ) Command: Control Jobs 19
CELLTRC Command: Trace Storage Cell Usage 25
CKPTRACE Command—Trace Checkpoint Data Set 27
CRYPTKEY Command: Define a Name for an Encryption Key 27
DSTREXCL Command: Exclude Data Set Triggers 30
ENCPARM QUIESCE Command: Quiesce CA WA Restart Option ESP Edition 31
ENCPARM RESTART Command: Restart CA WA Restart Option ESP Edition 32
ENCPARM STATUS Command: Display the Status of CA WA Restart Option ESP Edition 33
ESPCOM Command: Control Communications 33
INET Command: Manipulate TCP/IP Attributes 36
LDSN Command: List Data Sets 40
LISTCKPT Command: List Checkpoints 41
LISTQ Command: List Queue Data Set 42

LJ Command: List Job Status 43
LJS Command: List Job Statistics 45
LLMP Command—Display License Status 46
LMOD Command: Display Module Level 48
LOADAGDF Command: Load the Agent Definition 48
LOADJTDT Command: Load the Job-Tracking Definition Table 49
LOADUPDT Command: Load the User-Profile Definition Table 50
LOGPRT Command: Display Event Log 51
LOGTRACE Command: Consolidate Trace Data 52
LTCELL Command: List Tracking Cells 54
LTJ Command: List Tracked Job Summaries 55
MGRADDR Command: Specify Scheduling Manager's Address 57
PREALLOC Command: Preallocate Data Sets 59
QUIESCE Command: Quiesce z/OS Agent 61
RESTART Command: Restart z/OS Agent 61
ROUTING Command: Route Proxy Requests 62
SPINLOG Command—Spin a Log 63
STATUS Command: Display the Processing Status 64
TRACE Command: Activate Trace Facility 65
TRACEDEF Command: Identify Data Sets 67
TRACEPRT Command: Print Trace Data 68
XCF DISPLAY Command: View XCF Attributes 69
XCF SET TRACE Command: Specify XCF Trace Parameters 73
XCF START TRACE Command: Initialize/Activate Trace Files 75
XCF STOP TRACE Command: Stop Trace File 76
XDAB Command: Trace VSAM I/O Events 77
XFRB Command: Trace Slot I/O Events 80

Chapter 1: About This Guide

Overview

This guide contains all the commands and statements available to z/OS Agent users. Use it as a reference for when you want a complete description of the commands or statements, and all valid keywords and operands. Use the examples to see how each command and statement can be used.

Issue Commands

You issue z/OS Agent commands in a z/OS Agent client session. This procedure shows you how to access that session.

Follow these steps:

1. Issue TSO command ESPAGENT SUB(*subsysID*).

   A z/OS Agent client session starts.

2. Issue any z/OS Agent commands that you require.
3. Once you are finished issuing commands, type END.

   z/OS Agent client session ends.

Entering Statements

The content of this section also applies to commands and initialization parameters when contained in a data set.

Continuation

Type either a hyphen or a plus sign as the last non-blank character on a line to continue a line of input. The hyphen attaches the next line including any leading blank position. The plus sign strips leading blanks from a continuation line. Blanks preceding the hyphen or the plus sign are retained. Statements cannot extend beyond column 72. The continuation character can be placed in columns 72 or before.

Note: A hyphen can also be used as a wildcard character. If the wildcard hyphen is the last character on the line, it will be interpreted as a continuation character and not as a wildcard.

For the hyphen to be interpreted as a wildcard, it must be followed by a semicolon or something else on the line such as a comment: (/* */).

Comments

Enclose comments between /* and */. Comments can be written anywhere in a z/OS Agent procedure.

Data Sets

We recommend that you enclose all data set names in single quotation marks. Although this is not required, it will preserve your intentions across all environments. For example, in a TSO environment a user's TSO ID will be added as a prefix if the TSO PREFIXING option is active and single quote marks have not been used. Use LIB- or PAN- prefixes to identify Librarian and Panvalet data sets.

Delimiters

Use single quotation marks when you want to denote character strings and literal data in expressions, in assignment statements, and in built-in functions. You must include single quotation marks around a string that contains blanks.

Indentation

You can use indentation to improve readability.

Wildcards in z/OS Agent

Many statements, commands, or initialization parameters permit the use of the following wildcard characters (also called masking):

- An asterisk matches a specific single character.
- A hyphen matches zero or more characters. It must be used as the last character of the operand. If the wildcard hyphen is the last character on the line, it will be interpreted as a continuation character and not as a wildcard. For the hyphen to be interpreted as a wildcard, it must be followed by a semicolon or something else on the line such as a comment: (/* */).

Example: Wildcards in z/OS Agent

To display all calendars, use the following command:

```
LISTCAL –
```

To display all calendars with names containing XY in character positions three and four, use the following command:

```
LISTCAL **XY-
```

To display all calendars with two-character names, use the following command:

```
LISTCAL **
```

Wildcards in CA WA Restart Option ESP Edition

In certain operands of commands and statements used for CA WA Restart Option ESP Edition, you can use the wildcard characters in this topic to represent other characters.

Asterisk (*)

Type an asterisk anywhere in an operand to represent any single character, for example:

- PMIUTIL* matches the following:
- PMIUTIL1
- PMIUTIL2
- PMIUTILA
- etc.

PM*UTIL matches the following:

- PM1UTIL
- PM2UTIL
- PMAUTIL
- etc.

Hyphen (-)

Type one or more hyphens anywhere in an operand to represent zero or more characters, for example:

PMI- matches the following:

- PMI
- PMIA
- PMIUTIL1
- PMIA.Q1
- PMIA.Q1.RPT
- etc.

PMI-1 matches the following:

- PMI1
- PMIUTIL1
- PMIRPT.Q1
- PMIRPT.Q2.DLY1
- etc.

-MI-1 matches the following:

- PMI1
- TMIUTIL1
- PMI.UTIL1
- PMI.Q1.UTIL1
- etc.

PMI.-.RPT matches the following:

- PMI.Q1.RPT
- PMI.Q1.DLY.RPT
- etc.

If you type a hyphen last on a line, add a space and another character, such as a semicolon or a comment (/* */) so the hyphen is not read as a line continuation character, for example:

PMIUTIL- ;

Mixed wildcard characters

You can use the asterisk and hyphen wildcard characters in combination, for example:

*MI- matches the following:

- PMI
- TMI
- PMIUTIL
- PMI.Q1
- PMI.Q1.RPT
- etc.

**MI- matches the following:

- T1MI
- T1MIUTIL
- etc.

Examples

This section contains command syntax examples.

Issue Console Commands

To issue a command from the system console, use the MODIFY command, as in F ZAGA, where ZAGA is the started agent name. For example:

```
F ZAGA, STATUS
```

Issue OPER Commands

To issue a command that requires OPER authority, precede it with OPER. For example:

```
OPER STATUS
```

Issue Commands in Batch

To issue a command in batch, use the following JCL after your job card:

```
//EXEC PGM=ESP,REGION=4000K
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
z/OS Agent command
.
.
.
z/OS Agent command
```

System Name

If the system name for z/OS Agent is not ZOSAGENT, you need to type the system name when entering a command in batch. For example, if the system name is ZAGA, type:

```
//EXEC PGM=ZOSAGENT,PARM='SUBSYS(ZAGA)',REGION=4000K
```

Steplib

If the TSO command processor is not in a LINKLIST library, you need a STEPLIB or JOBLIB statement in your JCL. Your statement looks like the following:

```
//STEPLIB DD DSN=CYB.ZAG.LOAD,DISP=SHR
```

Conventions Used in This Guide

This section describes the conventions used in this guide.

Input Convention

You can type commands in uppercase or lowercase.

Syntax Conventions

Syntax descriptions in this guide use the following conventions:

Quotation marks " or '

Must be entered as shown.

Comma ,

Must be entered as shown.

Ellipsis …

The operand can be repeated. Do not enter ellipsis.

Lower Case Italics operand

An operand must be substituted. User supplied variable or character string.

Uppercase operand

The operand must be spelled as shown. You can enter the command and the operand in either upper or lower case.

OR-bar (|)

Indicates an exclusive value on left or right of bar. You must enter one of the items. You cannot enter more than one.

Underline _______

If you do not enter one of the operands, the system supplies the underlined operand, this is the default.

Parentheses ( ) and special characters

Operand enclosed in parentheses is mandatory and must be entered as shown.

Single operand in square brackets [ ]

Optional operand, do not type the brackets.

Stacked operands in braces

{ }

{ }

Mandatory, you must enter one of the operands. You cannot enter more than one.

Stacked operands in square brackets

[ ]

[ ]

Optional operand, you can enter one value, or none.

Operands with OR-bars (|) and square brackets

[ ]|[ ]

Optional, mutually exclusive operands. Enter one or none.

Stacked operands in square brackets within braces

{[ ]

[ ]}

Mandatory, you must enter one of these operands. You can enter more than one.

Note: If an operand has complex syntax, "See expanded syntax" will follow the operand. Expanded syntax is documented in a separate table after the main syntax table.

Chapter 2: Commands

This section contains the following topics:

AGENTMSG Command: Send Messages to a Scheduling Manager (see page 16)
APPLJOB(AJ) Command: Control Jobs (see page 19)
CELLTRC Command: Trace Storage Cell Usage (see page 25)
CKPTRACE Command—Trace Checkpoint Data Set (see page 27)
CRYPTKEY Command: Define a Name for an Encryption Key (see page 27)
DSTREXCL Command: Exclude Data Set Triggers (see page 30)
ENCPARM QUIESCE Command: Quiesce CA WA Restart Option ESP Edition (see page 31)
ENCPARM RESTART Command: Restart CA WA Restart Option ESP Edition (see page 32)
ENCPARM STATUS Command: Display the Status of CA WA Restart Option ESP Edition (see page 33)
ESPCOM Command: Control Communications (see page 33)
INET Command: Manipulate TCP/IP Attributes (see page 36)
LDSN Command: List Data Sets (see page 40)
LISTCKPT Command: List Checkpoints (see page 41)
LISTQ Command: List Queue Data Set (see page 42)
LJ Command: List Job Status (see page 43)
LJS Command: List Job Statistics (see page 45)
LLMP Command—Display License Status (see page 46)
LMOD Command: Display Module Level (see page 48)
LOADAGDF Command: Load the Agent Definition (see page 48)
LOADJTDT Command: Load the Job-Tracking Definition Table (see page 49)
LOADUPDT Command: Load the User-Profile Definition Table (see page 50)
LOGPRT Command: Display Event Log (see page 51)
LOGTRACE Command: Consolidate Trace Data (see page 52)
LTCELL Command: List Tracking Cells (see page 54)
LTJ Command: List Tracked Job Summaries (see page 55)
MGRADDR Command: Specify Scheduling Manager's Address (see page 57)
PREALLOC Command: Preallocate Data Sets (see page 59)
QUIESCE Command: Quiesce z/OS Agent (see page 61)
RESTART Command: Restart z/OS Agent (see page 61)
ROUTING Command: Route Proxy Requests (see page 62)
SPINLOG Command—Spin a Log (see page 63)
STATUS Command: Display the Processing Status (see page 64)
TRACE Command: Activate Trace Facility (see page 65)
TRACEDEF Command: Identify Data Sets (see page 67)
TRACEPRT Command: Print Trace Data (see page 68)
XCF DISPLAY Command: View XCF Attributes (see page 69)
XCF SET TRACE Command: Specify XCF Trace Parameters (see page 73)
XCF START TRACE Command: Initialize/Activate Trace Files (see page 75)
XCF STOP TRACE Command: Stop Trace File (see page 76)
XDAB Command: Trace VSAM I/O Events (see page 77)
XFRB Command: Trace Slot I/O Events (see page 80)

AGENTMSG Command: Send Messages to a Scheduling Manager

To send messages and issue commands to a scheduling manager, use the AGENTMSG command.

Type: General command

Authority: To issue the AGENTMSG command, you need read access to the SAF resource *prefix*.AGENTMSG.*verbsubverb*.*name*.

This command has the following format:

```
AGENTMSG {date|.} {time|.} managername {from|.} {objectname|.}
   verb subverb [(keyword)]

verb    subverb
CONTROL MGRADDR PORT(number) ADDRESS(address) [PERSISTENT(TRUE|FALSE)]}

verb    subverb
STATE   COMPLETE
  "     FAILED

verb    subverb
ACTION  HOLD
  "     RELEASE
  "     INSERT
```

date

The date the message is sent. Specify the actual date or use a period (.) as a placeholder.

time

The time the message is sent. Specify the actual time or use a period (.) as a placeholder.

managername

Specifies the name of the scheduling manager to whom you want to send messages and issue commands. This is a required parameter.

from

Who sent the message. Specify the name of z/OS Agent sending the message or use a period (.) as a placeholder.

objectname

Name of the object to which the command refers. For the commands that are not related to a specific job, specify a period (.) in place of the job name.

For z/OS Agent applications, the object name is specified in the following format:

```
oooo[.qqqq]/aaaaa.ggggg/fffff
```

- oooo is the job name.
- qqqq is the optional qualifier.
- aaaaa is the application name.
- ggggg is the generation number of the application.
- fffff is the appl file name.

  Currently, there is only one appl file and its name is MAIN.

verb

Indicates the action to be taken. The values are:

- CONTROL
- STATE
- ACTION

subverb

Further defines the action to be taken.

keyword

Modifies the verb-subverb combination.

The Control Verb

The CONTROL verb requests the scheduling manager to do the action that the subverb specifies. The subverb is:

MGRADDR

Tells the scheduling manager to respond to messages from z/OS Agent on a new TCP/IP address and port. Use the MGRADDR subverb when you move z/OS Agent to a new system. You can issue the AGENTMSG command with the MGRADDR subverb from the original z/OS Agent or from the new z/OS Agent. The following list describes the associated operands:

PORT(number)

Specifies the port number of the new z/OS Agent. PORT is a required operand.

ADDRESS(address)

Specifies the TCP/IP address of the new z/OS Agent. ADDRESS is a required operand.

PERSISTENT (TRUE|FALSE)

Indicates a permanent change or a temporary change on the agent. PERSISTENT is an optional operand.

- TRUE—Permanently saves z/OS Agent address and receiver port number on the scheduling manager. This enables the scheduling manager to know which receiver to connect when required.
- FALSE—Indicates z/OS Agent address and receiver port number only change temporarily on the scheduling manager. The address and port number are reset to the values defined in the scheduling manager topology the next time the scheduling manager restarts. FALSE is the default.

Note: Consider the following items when using the MGRADDR subverb:

- There is also an MGRADDR command, which you can issue from the new master only.
- MGRADDR is also an initialization parameter.
- This change is temporary until the agent restarts. If you must make the change permanent, use the PORT(port) and PERSISTENT(YES) operands.
- If the scheduling manager communicates with more than one z/OS Agent, each z/OS Agent must be defined with a unique name in the scheduling manager. Each of these z/OS Agent names in the scheduling manager must match the ZOSAGENT initialization parameter of the corresponding z/OS Agent.

The STATE Verb

The STATE verb requests the scheduling manager to change the state of a job to the state the subverb specifies. The subverbs are:

COMPLETE

Changes the state to COMPLETE. All successor jobs are released. No further state messages are accepted. The job can be resubmitted if the application containing it is not yet complete.

FAILED

Changes the state to FAILED. User action is required to correct the error before resubmission.

The ACTION Verb

The ACTION verb requests the scheduling manager to do the action that the subverb specifies. The subverbs are:

HOLD

Places the specified job on hold.

RELEASE

Releases the job from a hold state.

INSERT

Inserts a new job in the application.

Example: Change z/OS Agent Address and Port on the Agent

In this example, the scheduling manager named MYSCHEDULER is requested to respond to z/OS Agent at port number 5100 and address 111.11.11.1. The port number and address are permanently saved in the scheduling manager so the scheduling manager knows which receiver to connect when required.

```
AGENTMSG . . MYSCHEDULER . . CONTROL Mgraddr Port(5100)+
  Address(111.11.11.11) PERSISTENT(TRUE)
```

Example: Place a Job on Hold

In this example, the AGENTMSG command tells the scheduling manager to place the job PROGRAM1 on hold.

```
AGENTMSG "PROGRAM1/TEST.2/MAIN ACTION HOLD"
```

APPLJOB(AJ) Command: Control Jobs

To control jobs, use the APPLJOB command.

The short form of this command name is AJ.

Type: General command

This command has the following syntax:

```
{APPLJOB|AJ} jobspec
       [LIST]
       [APPLICATION(applspec)]
       {COMPLETE}
       {READY}
       {RESUBMIT}                         See Expanded Syntax following
       [DISPLAY]
       [EXPEDITE]
       [CANCEL[(DUMP|PURGE)]]

RESUBMIT   [DATASET(dsname)]
           [MEMBER(membername)]
           [JOBNAME(name)]
```

```
[RRJOBID(jobid)]
[USER1(userinfo)]
[USER2(userinfo)]
[USER3(userinfo)]
[USER4(userinfo)]
[ENCPARM1(encparmoperand)]
[ENCPARM2(encparmoperand)]
[ENCPARM3(encparmoperand)]
[ENCPARM4(encparmoperand)]
[ENCPARM5(encparmoperand)]
[ENCPARM6(encparmoperand)]
[ENCPARM7(encparmoperand)]
[ENCPARM8(encparmoperand)]
[ENCPARM9(encparmoperand)]
[ENCPARMA(encparmoperand)]
[ENCPARMB(encparmoperand)]
[ENCPARMC(encparmoperand)]
[FROMSTEP(stepname)]
[TOSTEP(stepname)]
[EXCLSTEP(stepname,...)]
[VERIFY|NOVERIFY]
[RESTART]
[CASESENSITIVE]
```

jobspec

Specifies the full name of the job.

LIST

Requests detailed information about the specified job. LIST is the default. If you specify LIST, APPLICATION is the only other valid operand.

APPLICATION(*applspec*)

Specifies the name and generation of the application containing the job that the APPLJOB command operates on. This operand is optional; however, it is mandatory when you specify REQUEST or RESUB. If the job is not yet submitted, specify the application name. If more than one generation of the application is active, you can specify the relative or absolute application generation (or the OLDEST keyword).

Format: Use one of the following formats for *applspec*:

- *aaa.nnn,* where *aaa* is the application name and *nnn* is the absolute generation number.
- *aaa.-rrr,* where *rrr* is the relative generation (the default for *rrr* is zero, which is the current generation).

COMPLETE

Simulates normal job completion. COMPLETE immediately posts successor jobs and external references. You can use COMPLETE when the job failed, but a rerun is unnecessary. You can also use COMPLETE before a job starts executing.

READY

Removes all job dependencies (except resources) from a job (including time, predecessors, manual hold).

RESUBMIT

Specifies the job is resubmitted. For more information about the RESUBMIT operands, see the section "RESUBMIT."

DISPLAY

Displays the JES status for the job.

EXPEDITE

Expedites a job manually. Requires the CA WA Service Governor ESP Edition.

CANCEL[(DUMP|PURGE)]

Cancels an executing job and optionally, purges it or requests a dump.

RESUBMIT

The RESUBMIT operand requests that the job is resubmitted. Use the last data set from which the job was submitted, unless you use the DATASET keyword to specify an alternate data set.

The following list describes the operands for the RESUBMIT operand:

DATASET(*dsname*)

Specifies an explicit JCL library name from which a job is resubmitted. You can include the member name in *dsname*.

Note: If, when resubmitting a job, you specify a JCL library name, the JCL library that you supply in the RESUBMIT DATASET operand will continue to be used for future resubmits. To revert to the original JCL library, specify an asterisk in the RESUBMIT DATASET operand. This operand applies to the resubmitted job and any future resubmitted jobs when you code the asterisk.

Except for RESUBMIT, entering an asterisk in the DATASET operand has the same effect as entering a blank.

MEMBER(*membername*)

Specifies a member name of the JCL data set for the job to be resubmitted.

JOBNAME(*name*)

Specifies the job name for the rerun, if the name is different from the original name.

RRJOBID(*jobid*)

Specifies the JES job number of the job being rerun. You can use RRJOBID only for jobs being restarted using CA WA Restart Option ESP Edition. By default, RRJOBID corresponds to the previous run of the job.

USER1(*userinfo*) and USER2 to USER4

Specifies user data for resubmitted jobs. This data passes to the USER1-USER4 symbolic variables. Initially, these variables are set to a null string (' ').

You can specify up to 68 characters of information for each value.

If the user data must be in lower or mixed case, specify the CASESENSITIVE option.

ENCPARM1 (*encparmoperand*) and ENCPARM2 to ENCPARMC

Specifies an operand of the ENCPARM command. You can specify up to 60 characters of information for each operand. The ENCPARM command is available only for jobs whose restart manager is CA WA Restart Option ESP Edition.

FROMSTEP (*stepname*)

Specifies the first step of the job to be rerun. Use (stepname.procstepname) for steps in a z/OS Agent procedure. You can use FROMSTEP only with CA WA Restart Option ESP Edition. The ENCPARM command is available only for jobs whose restart manager is CA WA Restart Option ESP Edition.

TOSTEP (*stepname*)

Specifies the last step of the job to be rerun. Use (stepname.procstepname) for steps in a z/OS Agent procedure. The default is the last step. You can use TOSTEP only with CA WA Restart Option ESP Edition. The ENCPARM command is available only for jobs whose restart manager is CA WA Restart Option ESP Edition.

VERIFY

Verifies that a data set and member exist by opening and closing it.

NOVERIFY

Does not verify that a data set and member exist.

EXCLSTEP(*stepname*)

Excludes a specific step or a list of specific steps from the list specified with FROMSTEP and TOSTEP. You can exclude a maximum of 100 steps.

RESTART

Restarts the job using CA WA Restart Option ESP Edition.

CASESENSITIVE

Disables case translation. Without this option, z/OS Agent converts the values of all operands to uppercase. Use this option if the user parameters USER1-USER4 must be in lowercase or mixed case.

You can abbreviate CASESENSITIVE to CS.

Example: List Detailed Information About a Job

The following AJ LIST command requests a detailed display of job JOBNN, in application WS24002, generation 3.

```
AJ JOBNN APPL(WS24002.3)
```

Here is a sample response:

```
Job=JOBNN(23663), Completed->CC 0
  Submitted ................ 2006/10/25 10:39
  Started .................. 2006/10/25 10:40
  Completed ................ 2006/10/25 10:40
  JobProfile=JOBNN.WS24002, Class=A, Priority=7, Step=STEP05,
  SrvClass=JES SLOW, SysName=SYSA, Sysplex=SYSAPLEX, Asid=00EA,
  SmfId=SYSA, JesNode=CYBJES, EspProc=CYBER.CNTL.ESPPROCS(WS24002)


Appl=WS24002.3
  Scheduled ................ 2006/10/25 07:17
  Built .................... 2006/10/25 07:17
  DueOutPropagation, Event=CYBER.WS24002, TrigUser=CYBUSR1
```

The following AJ command requests a detailed display of job JOB003 that has the job qualifier J3 in application APPL001, generation 4.

```
APPLJOB JOB003.J3 LIST APPLICATION(APPL001.4)
```

```
Job=JOB003.J3, PredWait, Hold=1
  Expected completion ...... 2003/12/04 17:02
  EspProc=CYBUSR1.ESPPROC(PROC001)

Predecessors:
  Job=JOB001.J1(42255), Started, Step=1

Successors:
  Job=JOB004.J4, PredWait, Hold=2

Resources:
  3 RES001
```

```
Enqueues:
  Shr ENQRES01, Excl ENQRES02, Shr ENQRES03, Excl ENQRES4,
  Shr ESPENQ_JOB(JOB001.J1)_APPL(-.-), Shr ESPENQ_JOB(JOB002.J2)_APPL(-.-),
  Excl ESPENQ_JOB(JOB003.J3)_APPL(APPL001.4)

Appl=APPL001.4
  Scheduled ................ 2003/12/04 16:51
  Built .................... 2003/12/04 16:51
  DueOutPropagation, Event=CYBUSR1.EVENT01, TrigUser=CYBUSR1
```

Example: Resubmit a Job

The following AJ command resubmits PAYJOB18 using CA WA Restart Option ESP Edition.

```
AJ PAYJOB18 RESUB RESTART APPL(PAYROLL.0)
```

Example: Resubmit a Job and Select Which Steps are Rerun

The following AJ command resubmits PAYJOB2 and selects which steps are rerun.

```
AJ PAYJOB2 APPL(PAYROLL.0) RESTART RESUB -
FROMSTEP(STEP020) TOSTEP(STEP060) -
EXCLSTEP(STEP0030 STEP0050)
```

The AJ command reruns only the steps marked with an asterisk. PAYJOB2 has the following steps:

- STEP005
- STEP020 *
- STEP030
- STEP040 *
- STEP050
- STEP060 *
- STEP070

Example: Resubmit a Job From a Different Library

The following AJ command resubmits PAYJOB2 from the CYBER.COPY.JCL library.

```
AJ PAYJOB2 RESUB APPL(PAYROLL.0) -
DATASET('CYBER.COPY.JCL')
```

Example: Resubmit a Job With a Variable

The following AJ command resubmits PAYJOB10 using original JCL library. The character string ',RESTART=STEP20' is passed to the JCL as the USER1 variable.

```
AJ PAYJOB10 RESUB USER1(',RESTART=STEP20') -
APPL(CYBER1)
```

The following statement is a sample job statement and the corresponding job statement that is used for resubmission.

```
PAYJOB10 JOB...MSGCLASS=X%USER1
```

The job statement resolves to the following example:

```
PAYJOB10 JOB...MSGCLASS=X,RESTART=STEP20
```

Example: Resubmit a Job and Pass a Run Type Operand to a Restart Program

PAYJOB11 is resubmitted using the USER2 symbolic variable. USER2 passes a run type operand to a restart program. For the first run of the job, the run type operand is P. The run type operand is R when the job is resubmitted.

```
In the JCL
     //STEP1 EXEC UCC11RMS,RUNTYPE='%USER2'
In the Symbol Library
     IF USER2='' THEN USER2='P'
On Resubmit
     AJ PAYJOB11 RESUB USER2('R') APPL(PAYROLL)
```

CELLTRC Command: Trace Storage Cell Usage

To diagnose problems involving uncontrolled storage allocation, use the CELLTRC command. The CELLTRC command traces storage cell usage.

Type: General command

Authority: To issue the CELLTRC command, you require OPER authority.

This command has the following format:

```
CELLTRC
```

A snapshot data set must first be created. This data set contains the storage cell snapshots that the CELLTRC command generates. Use the following DCB attributes:

```
DSORG=PS,RECFM=VB,LRECL=137,BLKSIZE=6144
```

The following DD statement must be added to z/OS Agent cataloged procedure:

```
//CELLTRC  DD  DISP=SHR,DSN=celltrc.dataset
```

Start z/OS Agent with the following command. The @CELLTRC keyword must be the first parameter in the PARM string. Once activated, cell tracing remains active only during the current execution of z/OS Agent. To keep the cell tracing active, the @CELLTRC keyword must be used each time z/OS Agent is started:

```
S ZAG,PARM='@CELLTRC,...'
```

The CELLTRC command causes z/OS Agent to generate a storage cell snapshot and place it in the cell trace data set. The cell trace data set is named in the CELLTRC DD statement. The CELLTRC command can be issued from a z/OS console or from line mode.

To deactivate the cell tracing, restart z/OS Agent without using the @CELLTRC keyword.

The storage cell snapshot consists of three sections: the cell pool header display, the cell display, and the cell summary.

The cell pool header display shows the available cell size and how many of each size are currently allocated. The COUNT field indicates how many cells of a particular size are allocated. The ACTIVITY field shows the number of gets and frees done for this cell size:

```
SIZE      8, COUNT     0, ACTIVITY      0
SIZE     16, COUNT    17, ACTIVITY     42
SIZE     32, COUNT    59, ACTIVITY    104
SIZE     64, COUNT    31, ACTIVITY     85
SIZE    128, COUNT    32, ACTIVITY    427
```

The cell display describes each individual cell that is presently allocated. Its size and requestor CSECT are shown, along with up to 64 bytes of its contents. Also shown is the CSECT that called the requestor CSECT. The offset within each CSECT from which the request was made is also shown:

```
CELL SIZE   128, REQUESTED BY CYBXI000+0054, CALLED BY CYBXR003+0CA6
     +0 00039F80  00000000 000076F3 00873C54 00000000 ...
    +20 00039FA0  00000000 00000000 003D2448 003D2450 ...

CELL SIZE    32, REQUESTED BY CYBXI003+0CF4, CALLED BY CYBJS107+0DB8
     +0 003D2448  00000000 008876F3 00000000 00000000 ...

...
```

Cell Summary

The cell summary lists the number of cells that are currently allocated, but arranged by requestor's base address and requesting address, and by the requestor's caller's base address and calling address. The following example shows that three cells of size 32 are currently allocated, having been requested by CYBXI003 at offset 0CF4 who was called by CYBJS107 at offset 0DB8. The base address of the CYBXI003 CSECT is at 02357708, and the request was made from address 023583FC:

```
---- STORAGE ALLOCATED BY REQUESTOR'S ADDRESS
ADDR 0234574C, BASE 023456F8, COUNT     1, BYTES   128 - CYBXI000+0054
ADDR 023583FC, BASE 02357708, COUNT     3, BYTES    32 - CYBXI003+0CF4

---- STORAGE ALLOCATED BY REQUESTOR'S CALLER'S ADDRESS
ADDR 02365E38, BASE 02365080, COUNT     3, BYTES    32 - CYBJS107+0DB8
ADDR 02358C36, BASE 02357F90, COUNT     1, BYTES   128 - CYBXR003+0CA6
```

Example: Trace Storage Cell Usage

In the following example, storage cell usage is generated and placed in the cell trace data set named in the CELLTRC DD statement:

```
CELLTRC
```

CKPTRACE Command—Trace Checkpoint Data Set

The CKPTRACE command traces the allocation and freeing of storage in the checkpoint data set CKPT. Use the CKPTRACE command only for diagnostic purposes.

Type: OPER command

Authority: Access to the security profile OPER.

This command has the following format:

```
CKPTRACE
```

Note: Use the CKPTRACE command only when requested by CA Support.

CRYPTKEY Command: Define a Name for an Encryption Key

The CRYPTKEY command defines and securely stores an encryption key that is used for communication by z/OS Agent with scheduling managers.

Type: Authorized command

Authority: You must have UPDATE authority to the PASSWORD security profile to delete or update an encryption key without supplying the old encryption key.

Applicability:

- CA Workload Automation DE
- CA Workload Automation AE, Release 11.3 or higher

This command has the following format:

```
CRYPTKEY{DEFINE KEYNAME(name) KEY(key) type}
        {DELETE KEYNAME(name) KEY(key) type}
        {LIST [KEYNAME(name)][KEY(key)] type}
        {UPDATE KEYNAME(name) KEY(key) OLDKEY(okey) type}
```

DEFINE

Defines a new encryption key.

DELETE

Deletes an encryption key and removes the connection to the scheduling manager.

LIST

Displays a list of encryption keys.

UPDATE

Updates an encryption key.

KEYNAME(*name*)

Defines a name for the encryption key. KEYNAME is required for DEFINE, UPDATE, and DELETE and is optional for LIST.

Limits: The maximum length is 16 characters.

KEY(*key*)

Specifies the encryption key. KEY is required for DEFINE, DELETE, and UPDATE and ignored with LIST.

Limits: The limit depends on the type of key. For details, see the *type* operand.

OLDKEY(*okey*)

Specifies the previous encryption key that is related to the name in KEYNAME. OLDKEY is required for UPDATE when you do not have UPDATE access to the host security CRYPTKEY profile. OLDKEY is ignored with DEFINE, DELETE, and LIST.

type

Specifies the type of encryption key. The values are as follows:

- DES—Maximum encryption key of 16 hexadecimal characters.
- BLOWFISH—Maximum encryption key of 64 hexadecimal characters.
- AES—Encryption key of exactly 32 hexadecimal characters (AES128 standard, 128 bits).

Important! If you enter more than the maximum number of characters for a DES or BLOWFISH key, the key is truncated. For AES, you get an error message for not specifying the exact key length.

Notes:

- This command encrypts an encryption key and stores it in a data set. The encryption standard that is used to encrypt the key is not related to the encryption standard used for the key itself (specified in the *type* operand).
- If you delete or change an encryption key, specify the old encryption key. If you have UPDATE authority to the PASSWORD security profile, you do not need to specify the old encryption key.
- If you specify an incorrect key in OLDKEY, the request fails. In this case, the request fails even if you have UPDATE authority to the PASSWORD security profile.

Example 1: DES Encryption

This example shows *type* DES encryption.

```
CRYPTKEY DEFINE KEYNAME(agntr6) KEY(X'0102030405060708') DES
```

Example 2: Blowfish Encryption

This example shows *type* BLOWFISH encryption.

```
CRYPTKEY DEFINE KEYNAME(agblf) KEY(X'12345678901234567890') BLOWFISH
```

Example 3: AES Encryption

This example shows *type* AES encryption.

```
CRYPTKEY DEFINE KEYNAME(aeskey) KEY(X'’000102030405060708090A0B0C0D0E0F') AES
```

DSTREXCL Command: Exclude Data Set Triggers

To flag a program entry in the Data set Trigger Exclude list as invalid, use the DSTREXCL command. These entries are not deleted from the list. Entries that are flagged as invalid are converted to lowercase.

Type

OPER command

Authority

To issue the DSTREXCL command, you require OPER authority.

This command has the following format:

```
DSTREXCL {pgmname}
         [(pgmname[,pgmname]...)]
```

pgmname

Indicates the name of a program to be flagged as no longer part of the Data set Trigger Exclude list.

The Data set Trigger Exclude list consists of a list of programs that are not eligible for data set triggering. The DSTREXCL initialization parameter specifies these programs. Once this list has been built, you use the DSTREXCL command to flag an entry in the Data set Trigger Exclude list as invalid.

Related Information

For information on adding an entry to the Data set Trigger Exclude list, see the DSTREXCL initialization parameter in the *Installation and Configuration Guide*.

Example: Flag an Entry in the Data Set Trigger Exclude List

This example shows how to flag entry PGM999 in the Data set Trigger Exclude list as invalid.

```
DSTREXCL PGM999
```

PGM999 is not deleted from the list, its entry is converted to lowercase as follows:

```
DSTRIG exclude PGM=pgm999
```

ENCPARM QUIESCE Command: Quiesce CA WA Restart Option ESP Edition

You can use the ENCPARM QUIESCE command to:

- Suspend CA WA Restart Option ESP Edition operation.
- Deallocate the EXH data set from z/OS Agent address space.
- Stop the recording of tracking information to the EXH data set.

Job tracking continues, but data is stored in the Auxiliary Address Space until the EXH data set is available again.

Important: Issue ENCPARM QUIESCE for the master and all proxies.

Type: CA WA Restart Option ESP Edition command

Authority: To issue the ENCPARM QUIESCE command, you require OPER authority.

This command has the following format:

```
ENCPARM QUIESCE
```

Notes:

Use the ENCPARM QUIESCE command in the following situations:

- When you need exclusive control (DISP=OLD) of the EXH data set.
- Before you issue the AUX_AS CANCEL command to stop the Auxiliary Address Space.
- Before you issue the AUX_AS BOUNCE command to stop and then start the Auxiliary Address Space.

Example: Issue the ENCPARM QUIESCE Command

This example illustrates the effect of issuing the ENCPARM QUIESCE command.

The ENCPARM STATUS command is issued to display the status of CA WA Restart Option ESP Edition.

```
oper encparm status
ESP ENCORE COMPRESSING ACTIVE
```

The ENCPARM QUIESCE command is issued.

```
oper encparm quiesce
ESP7103I OK
```

The ENCPARM STATUS command is issued again to display the status of CA WA Restart Option ESP Edition after the quiesce.

```
oper encparm status
ESP ENCORE COMPRESSING QUIESCED
```

ENCPARM RESTART Command: Restart CA WA Restart Option ESP Edition

You can use the ENCPARM RESTART command to:

- Allocate the EXH data set to z/OS Agent address space.
- Resume the recording of tracking information to the EXH data set.
- Drain tracking data that was accumulated in the Auxiliary Address Space and write the data to the EXH data set.

Important: Issue ENCPARM RESTART for the master and all proxies.

Type: CA WA Restart Option ESP Edition command

Authority: To issue the ENCPARM RESTART command, you require OPER authority.

This command has the following format:

```
ENCPARM RESTART
```

Notes:

Use the ENCPARM RESTART command in the following situations:

- When you no longer need exclusive control (DISP=OLD) of the EXH data set.
- After you issue the AUX_AS START or AUX_AS BOUNCE command to start the Auxiliary Address Space.
- After z/OS Agent or CA WA Restart Option ESP Edition was quiesced because the EXH data set was full.

Example: Issue the ENCPARM RESTART Command

This example illustrates the effect of issuing the ENCPARM RESTART command.

The ENCPARM STATUS command is issued to display the status of CA WA Restart Option ESP Edition.

```
oper encparm status
ESP ENCORE COMPRESSING QUIESCED
```

The ENCPARM RESTART command is issued.

```
oper encparm restart
ESP7103I OK
```

The ENCPARM STATUS command is issued again to display the status of CA WA Restart Option ESP Edition after the restart.

```
oper encparm status
ESP ENCORE COMPRESSING ACTIVE
```

ENCPARM STATUS Command: Display the Status of CA WA Restart Option ESP Edition

To display the status of CA WA Restart Option ESP Edition, use the ENCPARM STATUS command.

Type: CA WA Restart Option ESP Edition command

Authority: To issue the ENCPARM STATUS command, you require OPER authority.

This command has the following format:

```
ENCPARM STATUS
```

Example: Issue the ENCPARM STATUS Command

This example shows how to display the status of CA WA Restart Option ESP Edition.

```
oper encparm status
ESP ENCORE COMPRESSING QUIESCED
```

ESPCOM Command: Control Communications

To display and control communications activity between z/OS Agent and the scheduling manager, use the ESPCOM command.

Type: General command

Authority: To issue the ESPCOM command, you require OPER authority.

This command has the following format:

```
ESPCOM
|[[LIST|QUIESCE|RESTART|FLUSH] DEST(dest_name)]
```

ESPCOM

Lists the attributes of the COMMQ checkpoint data set in use when issued by itself.

LIST

Lists the attributes of the COMMQ checkpoint data set in use. LIST is the default. When the DEST keyword is included, the state of the specified scheduling managers is displayed.

QUIESCE

Quiesces the communication to the scheduling manager that the DEST keyword specifies. The scheduling manager continues processing.

RESTART

Restarts the communication with the scheduling manager that the DEST keyword specifies.

FLUSH

Purges all pending messages for the scheduling manager that the DEST keyword specifies.

DEST(*dest_name*)

Specifies the scheduling manager names that you want to view or manipulate. Wildcards can be used when specifying dest_name.

Example: List COMMQ Attributes

This example shows how to list the COMMQ data set attributes.

```
OPER ESPCOM
CommQ data set name: CYBER.ZAG.ESPCOMQ
               size: 1474560 bytes
               used: 320 bytes
1 senders are defined, 0 of them quiesced
```

Example: List COMMQ Attributes Using Wildcard

This example shows how to list the COMMQ data set attributes of all the scheduling managers whose names begin with AIX.

```
ESPCOM DEST(AIX-)
```

Here is a sample response:

```
CommQ data set name: CAB2.TST.VXXX.ZAG.ESPCOMQ
               size: 1474560 bytes
               used: 42952 bytes
Destination name|status|     last operation     | sent |queued|address
AIXTCNP         |active|Conn err 16.24.27 03 APR|     0|     0|111.111.30.144:81
AIXV3NP         |active|Conn err 16.24.27 03 APR|     0|     0|111.111.25.12:806
47 senders are defined, 0 of them quiesced
```

Example: Quiesce Multiple Scheduling Managers

This example shows how to quiesce all scheduling managers whose names begin with AIX.

```
ESPCOM QUIESCE DEST(AIX-)
```

Here is a sample response:

```
ESP1519I Quiesce requested for two destinations
```

You can verify that the scheduling managers have been quiesced by issuing this command:

```
OPER ESPCOM DEST(AIX-)
```

Here is a sample response:

```
CommQ data set name: CAB2.TST.VXXX.ZAG.ESPCOMQ
               size: 1474560 bytes
               used: 42952 bytes
Destination name|status|     last operation     | sent |queued|address
AIXTCNP         |quiet |Conn err 16.24.27 03 APR|     0|     0|111.111.30.144:81
AIXV3NP         |quiet |Conn err 16.24.27 03 APR|     0|     0|111.111.25.12:806
47 senders are defined, 2 of them quiesced
```

Example: Restart Multiple Scheduling Managers

This example shows how to quiesce all scheduling managers whose names begin with AIX.

```
ESPCOM DEST(AIX-) RESTART
```

Here is a sample response:

```
ESP1520I Restart requested for two destinations
```

You can verify that the scheduling managers have been restarted by issuing the following command:

```
ESPCOM DEST(AIX-)
```

Here is a sample response:

```
CommQ data set name: CAB2.TST.VXXX.ZAG.ESPCOMQ
               size: 1474560 bytes
               used: 51528 bytes
Destination name|status|     last operation     | sent |queued|address
AIXTCNP         |active|Conn err 16.24.27 03 APR|     0|     0|111.111.30.144:81
AIXV3NP         |active|Conn err 16.24.27 03 APR|     0|     0|111.111.25.12:806
47 senders are defined, 0 of them quiesced
```

Example: Flush Scheduling Manager Messages

This example shows how to flush the message queues of all scheduling managers whose names begin with AIX.

```
ESPCOM FLUSH DEST(AIX)
```

Here is a sample response:

```
ESP1521I Queue, flush requested for 2 destinations.
```

INET Command: Manipulate TCP/IP Attributes

To display and manipulate TCP/IP attributes, use the INET command.

Authority: To issue the INET command, you require OPER authority.

This command has the following format:

```
INET {DISPLAY} {ACTIVE}
               {CLOSE}
               {HOST_CACHE}
               {TCPIP}
               {TRACE}
               {BIND}
            {FLUSH HOST_CACHE}
     {HELP}
     {PURGE {ID socketid|TASK taskid}}
     {QUERY HOST hostname}
     {SET} {TRACE} SYSOUT(class)
           {CLOSE {RESET|SHUTDOWN}}
           {HOST_CACHE TTL(seconds)}
           {BIND {REUSEADDR|NOREUSEADDR}}
     {SPIN} {TRACE}
     {START}{TRACE}
     {STOP} {TRACE}
```

DISPLAY

Displays a TCP/IP object. The following options are supported:

- INET DISPLAY ACTIVE
- CLOSE
- INET DISPLAY HOST_CACHE
- INET DISPLAY TCPIP
- INET DISPLAY TRACE
- INET DISPLAY BIND

ACTIVE

Displays all active TCP/IP sockets.

CLOSE

Displays the current TCP/IP close option (RESET or SHUTDOWN).

HOST_CACHE

Displays z/OS Agent TCP/IP host cache entries.

TCPIP

Displays information on the current TCP/IP stack.

TRACE

Displays TCP/IP tracing facility.

BIND

Displays the current TCP/IP bind.

FLUSH HOST_CACHE

Flushes z/OS Agent TCP/IP host cache. Flushing is useful when a host cache entry has become invalid because of a modification to a host entry in the TCP/IP network Domain Name System (DNS) configuration.

HELP

Displays all the INET command options.

PURGE

Purges a TCP/IP socket or all TCP/IP sockets running under a specified task.

ID

Requests the purging of an active socket.

Note: This option is not supported if the TCP/IP stack is IUCV from IBM.

socketid

Displays a list of currently active TCP/IP sockets.

TASK

Displays a list of currently active TCP/IP sockets and their respective task IDs.

taskid

Displays a list of currently active TCP/IP sockets and their respective task IDs.

QUERY HOST *hostname*

Displays the TCP/IP address of a host. The short form is: INET Q H *hostname*.

SET

Sets a TCP/IP object.

SYSOUT(*class*)

Indicates a print data set for the TRACE log SYSOUT.

CLOSE

Use this operand only under the direction of CA Support. CLOSE tells z/OS Agent how to close a TCP/IP connection, either gracefully (SHUTDOWN) or forcefully (RESET). The default is SHUTDOWN.

HOST_CACHE TTL(*seconds*)

Activates or de-activates TCP/IP host caching, or resets the time-to-live (TTL) interval. *Seconds* is the time-to-live value within the range 0-99999999. A value of 0 de-activates TCP/IP host caching.

BIND {REUSEADDR|NOREUSEADDR}

Sets the TCP/IP bind option. When you use REUSEADDR, the TCP/IP bind does not fail if TCP/IP connection sockets are active using the requested TCP/IP port.

Note: If another listener socket is bound to the requested port, the TCP/IP bind always fails.

SPIN

Spins a TCP/IP object. Currently the only supported option is INET SPIN TRACE.

The TCP/IP trace sysout file is closed and deallocated. A new one is then immediately allocated and opened.

START

Starts a TCP/IP object. Currently the only supported option is INET START TRACE.

A TCP/IP trace sysout file is allocated and opened. Before the TCP/IP trace file can be started, it must be set to a sysout class by the INET SET TRACE SYSOUT(*class*) command.

STOP

Stops a TCP/IP object. Currently the only supported option is INET STOP TRACE.

The TCP/IP trace file is closed and de-allocated.

A DNS host name or IP address identifies z/OS Agent. If you use the host name, z/OS Agent converts it to an IP address before connecting. By default, the GetHostByName lookup routine does the connection.

z/OS Agent exploits the TCP/IP host caching mechanism to perform the host name to the IP address lookup process. When activated the first time:

- A lookup is done for a given host.
- A memory host cache entry is created.
- The GetHostByName lookup is performed.

z/OS Agent maintains a memory cache of a host name and the corresponding IP address after the first DNS lookup. The cache is used for subsequent lookups, sparing the overhead of unnecessary DNS lookups.

If you frequently use the same set of host names, activating host caching minimizes the number of real DNS lookups.

Host caching significantly improves z/OS Agent TCP/IP performance when:

- Scheduling managers are identified to z/OS Agent by host names rather than IP dotted decimal address.
- Connections are frequent.
- TCP/IP name server resides on a different system from z/OS Agent.

Example: List INET Options

This example shows how to display all of the INET command operands.

```
INET HELP
```

Example: Display TCP/IP Sockets

This example shows how to display all active TCP/IP sockets.

```
INET DISPLAY ACTIVE
```

Example: Purge All Sockets

This example shows how to purge all TCP/IP sockets running under task ID 78E0.

```
INET PURGE TASK 78E0
```

Example: Activate Host Caching

This example shows how to use the short form of the INET SET HOST_CACHE command to activate TCP/IP host caching. This command also sets the TTL interval for each host cache entry to one hour.

```
INET T HC TTL(3600)
```

Example: Deactivate Host Caching

This example shows how to use the short form of the INET SET HOST_CACHE command to deactivate TCP/IP host caching.

```
INET T HC TTL(0)
```

LDSN Command: List Data Sets

To produce a display of the current system data sets that are allocated to z/OS Agent, use the LDSN command.

Type: General command

This command has the following format:

```
LDSN
```

Related Information

For information on defining z/OS Agent data sets, see the *Installation and Configuration Guide*.

Example: Display the System Data Sets

This example shows how to display the current system data sets that are allocated to z/OS Agent.

```
CHECKPOINT: CYBER.ZAG.CKPT
     INDEX: CYBER.ZAG.INDEX BKUP CYBER.ZAG.BACKUP.INDEX SHR
  JOBINDEX: CYBER.ZAG.JOBINDEX BKUP CYBER.ZAG.BACKUP.JOBINDEX SHR
     QUEUE: CYBER.ZAG.QUEUE SHR
  TRAKFILE: CYBER.ZAG.TRAKFILE SHR
  JOBSTATS: CYBER.ZAG.JOBSTATS BKUP CYBER.ZAG.BACKUP.JOBSTATS
  EVENTSET: CYBER.ZAG.EVENT1 SHR
  HISTFILE: CYBER.ZAG.HISTFILE
  APPLFILE: CYBER.ZAG.APPLFILE SHR
      JTDT: CYBER.ZAG.PARMLIB(JOBDEF)
      UPDT: CYBER.ZAG.PARMLIB(PROFILE)
     COMMQ: CYBER.ZAG.COMMQ
  NETQUEUE: CYBER.ZAG.NETQUEUE
```

LISTCKPT Command: List Checkpoints

The LISTCKPT command lists statistics for the checkpoint data set.

Type: General command

Authority: To issue the LISTCKPT command, you require OPER authority.

This command has the following format:

```
LISTCKPT
```

The LISTCKPT command displays the following information:

- The checkpoint data set name
- The maximum size available
- The current usage

Related Information

For information on defining z/OS Agent data sets, see the *Installation* and *Configuration Guide*.

Example: Display the Checkpoint Data Set Statistics

This example shows how to display the statistics for the checkpoint data set.

```
OPER LISTCKPT
PRIMARY CHECKPOINT CAB2.TST114.CKPT
MAX CHECKPOINT SIZE 737280
HIGHEST ADDRESS USED 3232
32 BYTES IMBEDDED FREE SPACE
```

LISTQ Command: List Queue Data Set

To display information about the current state of the QUEUE data set, use the LISTQ command.

Type: General command

Authority: To issue the LISTQ command, you require OPER authority.

This command has the following format:

```
LISTQ
```

The LISTQ command displays the following information:

- The time the QUEUE data set was formatted.
- The amount of space that is allocated and the amount of space used.
- The MINHOLD, MINDORM, and MAXDORM values.

Related Information

For information on defining z/OS Agent data sets, see the *Installation* and *Configuration Guide*.

Example: Display the Queue Data Set

This example shows how to display the status of the QUEUE data set.

```
OPER LISTQ
QUEUE DATASET CAB2.TST114.QUEUE
FORMATTED AT 12.33.25 ON MONDAY JULY 29TH, 2013
MAXIMUM SIZE 737280 BYTES
HIGHEST ADDRESS USED 72160, 6112 BYTES IMBEDDED FREE SPACE
MINHOLD 100, MINDORM 100, MAXDORM 100
```

LJ Command: List Job Status

To list the job status information from the job tracking data set (TRAKFILE), use the LJ command.

Type: General command

This command has the following format:

```
LJ job
   [ALL]
   [STEPS]
   [SUBMIT]
   [PNODE]
   [STATUS]
   [DUEOUT]
   [EXTENDED|X]
   [DUMP]
```

job

The *job* operand indicates a full name (job name), job number or both. The full name represents 64-character job names. For a complete definition, see the Full name in the Glossary.

You can specify a job name followed by a job number in parentheses. When the *job* operand is only numeric, it is considered a job number. Otherwise, it is considered a job name.

When you specify a JES job number that both a job and a started task share, the LJ command displays the most recently executed.

Specify a job name when you specify a JES job number because JES job numbers are reused. In particular, avoid using the job number that is obtained from the output of the LTJ command as the job number for the LJ command. The job number that is obtained from the output of the LTJ command may have been reused and assigned to a different job.

ALL

Displays all the information for the job. The ALL operand is the same as specifying STEPS, SUBMIT, PNODE, and INFODATA.

STEPS

Displays statistics for the job steps.

SUBMIT

Displays the job submission information, including the JCL source data set, Event name, and data set trigger name (if any). This information is available only if z/OS Agent submitted the job.

PNODE

Displays the pnodes for each job.

STATUS

Displays the status of the job.

DUEOUT

Displays pnode dueout times for the job.

EXTENDED or X

Displays detailed date and time information.

DUMP

Displays job records from the job tracking data set in hexadecimal format for diagnosing problems.

z/OS Agent can track the progress of all jobs or selected jobs, even jobs it did not schedule. Real-time information is provided for each job step, including stepname, completion code, CPU time, elapsed time, and number of tape mounts.

You can request z/OS Agent to track jobs, started tasks (STCs), TSO users (TSUs), and system messages.

The LJ command displays information from the tracking file (TRAKFILE). The amount of data that is stored in this file depends on its size. The TRAKFILE slots are reused as required.

Related Information

For information on displaying tracked jobs, see the LTJ Command (see page 55).

For information on the job tracking, see the *Installation and Configuration Guide*.

Example: List a Specific Job

This example shows how to display the status of PAYJOB1.

```
LJ PAYJOB1 STATUS
```

Example: List a Job by the JES Number and Full Name

This example shows how to display the status of job PAYJOB2.SPECIALACCOUNTINGJOB, whose JES job number is 1234.

```
LJ PAYJOB2.SPECIALACCOUNTINGJOB(1234)
```

Example: List Step-Level Statistics

This example shows how to display step-level statistics.

```
LJ PAYJOB2 STEPS
```

Example: Display Job Status Information

This example shows how to display the date and time information for PAYJOB4 using the EXTENDED or X operand of the LJ command.

```
LJ PAYJOB4 X
PAYJOB4 J08324, IN OUTPUT QUEUE SINCE 16.01.03 28JAN, CC 0
SUBMITTED BY ESP AT 16.01.03 ON WED 28JAN, EVENT CATST.PAYROLL
JCL FROM CATST.JCL.CNTL(PAYJOB4)
PROGRAMMER USR01, ACCOUNT CAB3000
JOB IS IN APPL PAYROLL
STEPNAME-PROCSTEP-PROGNAME--EXCP-#T-S-N-CPU-TIME-SUNITS-REGION-CMPC
S1                IEFBR14      0  0 0 0  0:00.02    168     4K    0
PNODE----OUT------DATE--QTIME-POST_BY--SYS-
INPUT    16.01.02 28JAN     0          SYSA
EXEC     16.01.03 28JAN     1 SYSTEM   SYSA
OUTPUT     *            10M54
```

LJS Command: List Job Statistics

The LJS command lists the latest job submission or execution statistics for z/OS Agent jobs.

Type: General command

This command has the following format:

```
LJS JOB(fullname)
    [INDEX]
    [DUMP]
```

fullname

Indicates the full name of the job you want to list.

INDEX

Lists the job index entries. For each indexed job, the following information displays: job number, submission date and time, status, completion code, and execution time.

DUMP

Lists job records from the job statistics data set in hexadecimal format for diagnosing problems. z/OS Agent ignores DUMP when you use a wildcard in *fullname.*

When a failed or abended job is resubmitted with the same job name, application name, and application generation number, the index entry in the job statistics data set is reused.

Related Information

For information on the job statistics data set (JOBSTATS), see the DFLTDSN initialization parameter in the *Installation and Configuration Guide*.

Example: List Job Index Entries

This example shows how to list ten job index entries for job PAYJOB1.

```
LJS JOB(PAYJOB1) INDEX
JOB PAYJOB1, OWNER NONE, 10 JOBS INDEXED
 JOB27329 ON RDR AT 07.47.38 FRI 28JUL2006, COMPLETED, CC 0, 5.0 MINS_EXEC, 0:0
 JOB26227 ON RDR AT 00.47.39 FRI 28JUL2006, COMPLETED, CC 0, 5.6 MINS_EXEC, 0:0
 JOB25514 ON RDR AT 17.47.40 THU 27JUL2006, COMPLETED, CC 0, 4.8 MINS_EXEC, 0:0
 JOB25071 ON RDR AT 10.47.38 THU 27JUL2006, COMPLETED, CC 0, 4.2 MINS_EXEC, 0:0
 JOB24808 ON RDR AT 08.04.39 THU 27JUL2006, COMPLETED, CC 0, 4.8 MINS_EXEC, 0:0
 JOB24332 ON RDR AT 03.47.37 THU 27JUL2006, COMPLETED, CC 0, 4.0 MINS_EXEC, 0:0
 JOB23453 ON RDR AT 20.47.37 WED 26JUL2006, COMPLETED, CC 0, 3.8 MINS_EXEC, 0:0
 JOB23150 ON RDR AT 13.47.36 WED 26JUL2006, COMPLETED, CC 0, 3.6 MINS_EXEC, 0:0
 JOB22371 ON RDR AT 06.47.37 WED 26JUL2006, COMPLETED, CC 0, 3.8 MINS_EXEC, 0:0
 JOB20491 ON RDR AT 23.47.36 TUE 25JUL2006, COMPLETED, CC 0, 2.6 MINS_EXEC, 0:0
```

LLMP Command—Display License Status

The LLMP command displays the CA License Management Program (CA LMP) licensing status of z/OS Agent. The LLMP command also displays any optional products that are installed on top of z/OS Agent.

Type: OPER command or general command

Authority: You require OPER authority only if you want to issue the LLMP command as an OPER command. You do not require any special authority to issue the LLMP command as a general command.

This command has the following format:

```
LLMP
```

Note: When you issue LLMP as a general command, only the license status for the product is displayed.

The license statuses are as follows:

- Valid

  The license is valid.

- Expires within 30 days

  The license expires within the next 30 days.

- Expired or not found

  The license has expired or the product is unknown to CA LMP.

- License query return code *return-code*

  return-code represents a return code from CA LMP. If you see this message, contact CA Support.

Example: Issue LLMP as an OPER Command

The following example shows the type of information you receive when LLMP is entered as an OPER command.

You receive the CA LMP product code ("Code" column), product name ("Product" column), product release, and the license status for the main product. In this example, the main product is z/OS Agent. Additionally, you receive any optional products that are installed on top of the main product.

```
OPER LLMP
Code Product                                Release    License
WX   CA WA System Agent for z/OS            11.4       Valid
WU   CA WA High Availability ESP Edition    11.4       Valid
WV   CA WA Service Governor ESP Edition     11.4       Valid
WW   CA WA Restart Option ESP Edition       11.4       Valid
```

Example: Issue LLMP as a General Command

The following example shows the type of information you receive when LLMP is entered as a general command.

As noted, you receive only the license status for the main product. In this example, the main product is z/OS Agent.

```
LLMP
Code Product                                Release    License
WX   CA WA System Agent for z/OS            11.4       Valid
```

LMOD Command: Display Module Level

The LMOD command is typically issued at the request of CA Support to determine your module level (PTF level).

Type: General command

This command has the following format:

```
LMOD modulename
```

modulename

Specifies the name of the module you want to know about.

If you precede the LMOD command with OPER, z/OS Agent looks in the address space. If you do not precede the LMOD command with OPER, z/OS Agent looks in the client space.

Examples

The following two examples show the LMOD command and the response. In these examples, the modules belong to the product CA WA ESP Edition.

```
LMOD CYBES056
Module CYBES056 found at address 331E1600 in load module ESP
Load module ESP is at address 331E1600, entry point is B31E4750
Module eyecatcher: CYBES056 14.22 20110909 T5HC025
Load module found in STEPLIB CYB1.xxxxx.xxxx.CD7YLOAD
```

```
OPER LMOD CYBES056
Module CYBES056 found at address 330388F8 in load module CYBJS000
Load module CYBJS000 is at address 32F00000, entry point is B2F00000
Module eyecatcher: CYBES056 13.21 20111012 T5HC025
Load module found in STEPLIB APC.MOTM.xxx.xxxx.CD7YLOAD
```

LOADAGDF Command: Load the Agent Definition

To load the agent definition data set (AGENTDEF), use the LOADAGDF command.

Type: OPER command

Authority: To issue the LOADAGDF command, you require OPER authority.

This command has the following format:

```
LOADAGDF 'dsname[(member)]' [TEST]
```

dsname

Specifies the name of a sequential data set or a member of a partitioned data set; must have an LRECL=80.

TEST

Indicates that all syntax and logical checks are performed, but the table is not loaded.

The LOADAGDF command is coded in the initialization parameters to load the agent definition each time z/OS Agent initializes.

You can issue the LOADAGDF command from line mode to pick up changes that are made to the agent definition data set.

Example: Load the Agent Definition Data Set from Line Mode

The following example shows how to load the agent definition data set from line mode.

```
OPER LOADAGDF 'CYBER.ESPPARM(AGENTDEF)'
```

LOADJTDT Command: Load the Job-Tracking Definition Table

To load a job-tracking definition table into the system, use the LOADJTDT command. z/OS Agent uses this table to determine what it must track.

Type: OPER command

Authority: To issue the LOADJTDT command, you require OPER authority.

This command has the following format:

```
LOADJTDT 'dsname[(member)]'
```

dsname

Specifies the name of a sequential data set or PDS member that contains the job-tracking definition table.

Notes:

- The job-tracking definition table is typically loaded when z/OS Agent initializes.
- You can replace the currently active job-tracking definition table with a new table. At any time, reissue the LOADJTDT command with the data set name that contains the new table.
- If more than one copy of z/OS Agent uses the same job tracking definition table, issue the LOADJTDT command for each master, proxy, and shadow manager (if in use).
- The job-tracking definition table overrides any other tracking definitions, such as, tracking definitions that are defined with the DEFTJ command.

Related Information

For information on job-tracking definition tables, see the *Installation and Configuration Guide*.

Example: Load the Job-Tracking Definition Table

The following example shows how to load the job-tracking definition table JOBDEF1 from CYBER.PARMLIB.

```
OPER LOADJTDT 'CYBER.PARMLIB(JOBDEF1)'
```

LOADUPDT Command: Load the User-Profile Definition Table

To load the user-profile definition table from a data set, use the LOADUPDT command. The user-profile definition table contains one or more PROFILE statements.

Type: OPER command

Authority: To issue the LOADUPDT command, you require OPER authority.

This command has the following format:

```
LOADUPDT 'dsname[(member)]'
```

dsname

Indicates a fully qualified data set name that can include a member name. The data set can be any fixed-length or variable-length data set with a record length of 80 or greater.

Notes:

- You can issue the LOADUPDT command from your console when z/OS Agent is active. After you have loaded the table, z/OS Agent refers to the table, not to the data set. z/OS Agent always refers to the currently loaded table.
- You can also code the LOADUPDT command in the initialization parameters.
- The user-profile definition table must be loaded after changes are made.
- If you are running more than one z/OS Agent system, load the user-profile definition table on each system.

Example: Load the User-Profile Definition Table

The following example shows how to load the user-profile definition table PROFILE from CYBER.PARMLIB.

```
LOADUPDT 'CYBER.PARMLIB(PROFILE)'
```

LOGPRT Command: Display Event Log

To display event log data that is collected in the trace data sets during a specified time period, use the LOGPRT command.

Type

General command

This command has the following format:

```
LOGPRT DSN(dsname[,dsname]...)
       [FROM(schedule)]
```

dsname

Indicates the names of the trace data sets to access for event log information. Separate data set names with a comma.

schedule

Indicates schedule criteria. The keyword UNTIL can be used to restrict the time range. If UNTIL is not used, the current time is used as the end time.

This command is not normally used, as the same information is stored in z/OS Agent audit log. Immediately before using the LOGPRT command, issue the TRACE WRITE, TRACE CLOSE, and TRACE OPEN operator commands to ensure that the LOGPRT command accesses all current trace data. Trace data set information is displayed via the TRACE STATUS operator command.

Related information

For information on printing a trace data set, see the TRACEPRT command.

Example: Display time range

This example shows how to display information that is logged between 6pm the previous day and 5am this morning in data set CYB.ZAG.TRACE1.

```
LOGPRT DSN('CYB.ZAG.TRACE1') FROM('6PM YESTERDAY -
UNTIL 5AM TODAY')
```

LOGTRACE Command: Consolidate Trace Data

The LOGTRACE command consolidates certain trace data into a single sysout data set. Currently, LOGTRACE consolidates trace data produced by the following commands or initialization parameters:

- XDAB
- XFRB

Type

OPER command

Authority

To issue the LOGTRACE command, you require OPER authority.

This command has the following format:

```
LOGTRACE SET|T SYSOUT(class)
```

```
LOGTRACE {START|S}
         {STOP|P}
         {RESTART|RES|RS|SPIN|SP}
         {DISPLAY|D}
         {HELP|?}
```

SET|T SYSOUT(*class*)

Set the sysout class for trace data to *class.*

START

Start consolidating trace data.

STOP|P

Stop consolidating trace data.

RESTART|RES|RS|SPIN|SP

Stop and restart consolidating trace data.

DISPLAY|D

Display the current LOGTRACE status.

HELP|?

Display LOGTRACE command options.

Set Up Trace Data Consolidation

Set the sysout class for the trace data, before you start the trace data consolidation. For example,

```
LOGTRACE SET SYSOUT(A)
```

To include data from a particular trace in the LOGTRACE consolidation, specify the LOGTRACE operand in the command that sets up that trace. For example, to include VSAM I/O event trace data in the consolidation, you issue the following command:

```
XDAB SET TRACE LOGTRACE
```

Start Trace Data Consolidation

Data for a trace is not generated and is not consolidated with data from other traces until you do the following tasks:

- Issue the LOGTRACE command to start the consolidation facility.
- Issue the command to start the trace.

For example,

```
LOGTRACE START
XDAB START TRACE
```

Alternatives to the LOGTRACE Command

Instead of issuing the LOGTRACE command, you can:

- Code the LOGTRACE parameter in the START command.
- Code the LOGTRACE initialization parameter.

Related Information

See the LOGTRACE initialization parameter in the *Initialization and Configuration Guide*.

Example: Set the Trace Data SYSOUT Class

This example shows how to set the sysout class for trace data to A.

```
LOGTRACE SET SYSOUT(A)
```

Example: Start Consolidating Trace Data

This example shows how to start consolidating trace data to the sysout data set.

```
LOGTRACE START
```

Example: Stop Consolidating Trace Data

This example shows how to stop consolidating trace data to the sysout data set.

```
LOGTRACE P
```

Example: Restart Consolidating Trace Data

This example shows how to restart consolidating trace data to the sysout data set.

```
LOGTRACE RES
```

LTCELL Command: List Tracking Cells

To display the tracking cell (TCELL) information, use the LTCELL command. The TCELL command includes the size and current usage of each cell.

Type: General command

This command has the following format:

```
LTCELL
```

TCELLs are required to pass the job start, step end, job end, and job purge information to z/OS Agent.

Related Information

For information on defining tracking storage cells, see the TCELL initialization parameter in the *Installation and Configuration Guide*.

Example: Display Tracking Cell Information

The following example shows the tracking cell information:

```
CELSIZE   POOLSZ  MAXSIZE   CURRUSE  MAXUSED  GETMAINS  OVERFLOW

104         100    5100       0        53      0         0
168         100    5100       0        28      0         0
208         100     100       0        31      0         0
```

LTJ Command: List Tracked Job Summaries

To list tracked job summaries that are stored on the job index data set (JOBINDEX), use the LTJ command.

Type: General command

This command has the following format:

```
LTJ jobname
    [PREFIX]
    [OWNER(ownerid)]
    [MODEL(modelname)]
    [INDEX]
    [DUMP]
```

jobname

Specifies the name of the job or jobs to be listed. You can use masking.

PREFIX

List the tracking parameters the DEFTJ command and its PREFIX operand defines, for jobs starting with *jobname*. The list does not include tracking parameters that are defined in the job tracking definition table.

OWNER*(ownerid)*

List only jobs with ownership strings equal to the specified ownership string. You can use masking. The list does not include job summaries for jobs that are defined in the job tracking definition table.

MODEL(*modelname)*

List only jobs having a matching tracking model name. You can use masking.

INDEX

List the job index data set entries for the job. For each indexed job, the list includes job number, submission date and time, the status, and completion code.

The DEFTM INDEX initialization parameter or the DEFTJ *indexcount* command limits the number of index entries in the list. Sometimes, the number of entries that the LTJ command displays exceeds the specified limit. Exceeding the specified limit can happen when older job entries are not complete before a new job entry is added. After a job associated with the older entry has completed, and when a new job entry is added, the oldest entry among the completed jobs is removed.

DUMP

List the job records from the job index data set in hexadecimal format for diagnosing problems. DUMP is ignored when masking is used in *jobname*.

Note: The LTJ command lists the tracking characteristics of a job, as defined in the job tracking definition table or through the DEFTJ command, rather than the tracking data on a particular submission of a job.

Related Information

For information on displaying the status of a job that z/OS Agent is tracking, see the LJ Command (see page 43).

Example: List the Job Status

This example shows how to list the tracking status of PAYJOB1.

```
LTJ PAYJOB1
```

Example: Use Specific Tracking Models

This example shows how to list all jobs using tracking models with names beginning with PROD.

```
LTJ - MODEL(PROD-)
```

Example: List the Job Index Information

This example shows how to list the job index information for job PAYJOB1.

```
LTJ PAYJOB1 INDEX
JOB PAYJOB1, MODEL MODEL1, OWNER NONE, 10 JOBS INDEXED, 10 MAX
 JOB08720 ON RDR AT 20.00 THU 29JAN98, ENDED, CC 0, 0.0 MINS_EXEC, 0:00 CPU
 JOB08719 ON RDR AT 20.00 THU 29JAN98, ENDED, CC 0, 0.0 MINS_EXEC, 0:00 CPU
 JOB08717 ON RDR AT 20.00 THU 29JAN98, ENDED, CC 0, 0.0 MINS_EXEC, 0:00 CPU
 JOB08673 ON RDR AT 18.00 THU 29JAN98, ENDED, CC 0, 0.0 MINS_EXEC, 0:00 CPU
 JOB08630 ON RDR AT 16.01 THU 29JAN98, ENDED, CC S806, 0.0 MINS_EXEC, 0:00 CPU
 JOB08474 ON RDR AT 08.00 THU 29JAN98, ENDED, CC 0, 0.0 MINS_EXEC, 0:00 CPU
 JOB08414 ON RDR AT 20.00 WED 28JAN98, ENDED, CC 0, 0.0 MINS_EXEC, 0:00 CPU
 JOB08408 ON RDR AT 20.00 WED 28JAN98, ENDED, CC 0, 0.0 MINS_EXEC, 0:00 CPU
 JOB08365 ON RDR AT 18.00 WED 28JAN98, ENDED, CC 0, 0.0 MINS_EXEC, 0:00 CPU
 JOB08364 ON RDR AT 18.00 WED 28JAN98, ENDED, CC 0, 0.0 MINS_EXEC, 0:00 CPU
```

MGRADDR Command: Specify Scheduling Manager's Address

To notify one or more scheduling managers of the address of z/OS Agent, use the MGRADDR command. The MGRADDR command enables the scheduling manager to know which agent receiver to connect. The address is always sent automatically.

Type: OPER command

Authority: To issue the MGRADDR command, you require OPER authority.

This command has the following format:

```
MGRADDR {MANAGER|MANAGER(managername)[PORT(port)] [PERSISTENT(YES|NO)]}
        {DISPLAY}
        {HELP}
        {LIST}
```

MANAGER

Sends a notification to all the scheduling managers.

MANAGER(*managername*)

Indicates the unique name of the scheduling manager, as defined in the scheduling manager topology. To specify multiple scheduling managers, use the hyphen wildcard.

PORT(*port*)

Notifies a scheduling manager of the agent receiver port number to connect.

PERSISTENT(YES|NO)

Indicates a permanent or temporary change on the scheduling manager.

YES

Specifies to save (permanently) z/OS Agent address and receiver port number on the scheduling manager. The YES operand enables the scheduling manager to know the receiver to connect when required.

NO

Indicates z/OS Agent address and receiver port number only change temporarily on the scheduling manager. The address and port number are reset to the values defined in the scheduling manager topology the next time the scheduling manager restarts. The NO operand is the default.

DISPLAY

Displays the address and home TCP/IP address of the local z/OS Agent. They are the same if the TCPIP operand is specified in the ZOSAGENT initialization parameter.

HELP

Displays the MGRADDR command options.

LIST

Displays the address and home TCP/IP address of the local z/OS Agent. They are the same if the TCPIP operand is specified in the ZOSAGENT parameter of the agent definition file.

Note: The MGRADDR command is useful when z/OS Agent is stopped on one system and started on another because the TCP/IP address changes. MGRADDR tells the scheduling manager to respond to messages from z/OS Agent on a new TCP/IP address and port. You can issue the MGRADDR command only from the new system.

Use the MGRADDR command when you move z/OS Agent to a new system. This change is temporary until the agent restarts. If the change is permanent, issue the MGRADDR command with the PORT(*port*) and PERSISTENT(YES) operands.

Note: You can also issue the AGENTMSG command with the MGRADDR verb from the original system or from the new system.

Example: Display z/OS Agent Address

To display the address and home TCP/IP address of the local z/OS Agent, enter one of the following commands:

- MGRADDR
- MGRADDR DISPLAY
- MGRADDR LIST

Example: Notify All Scheduling Managers

These two examples show how to send a notification to all the scheduling managers.

```
MGRADDR MANAGER
MGRADDR MANAGER(-)
```

Example: Notify Multiple Scheduling Managers

This example shows how to send a notification to scheduling managers with the name prefix AUTO.

```
MGRADDR MANAGER(AUTO-)
```

Example: Specify Port Number

This example shows how to send a notification to the scheduling manager AUTOSYS1 and inform the scheduling manager to connect to agent receiver TCP/IP port 5451.

```
MGRADDR MANAGER(AUTOSYS1) PORT(5451)
```

Example: Permanently Change z/OS Agent's Address and Port Number on the Scheduling Manager

This example shows how to send a notification to a scheduling manager called AUTOSYS1, informing it to connect to agent receiver TCP/IP port 5451. The information is permanently saved in the AUTOSYS1 scheduling manager so the scheduling manager knows which receiver to connect when required.

```
MGRADDR MANAGER(AUTOSYS1) PORT(5451) PERSISTENT(YES)
```

PREALLOC Command: Preallocate Data Sets

To preallocate and deallocate a data set, use the PREALLOC command.

Type

OPER command

Authority

To issue the PREALLOC command, you require OPER authority.

This command has the following format:

```
PREALLOC [dataset]
         [ALLOC|UNALLOC]
```

dataset

Indicates the data set name.

ALLOC

Requests that the data set be allocated. ALLOC is the default.

UNALLOC

Requests that the data set be unallocated.

When access to the data set is needed, z/OS Agent bypasses the dynamic allocation for each preallocated data set.

Use one PREALLOC command for each data set you wish to preallocate.

You can issue the PREALLOC command without any operands to display the current preallocation list of data sets. Issue the PREALLOC command with the UNALLOC operand to de-allocate a previously allocated data set.

Related information

For information on defining z/OS Agent data sets, see the *Installation and Configuration Guide*.

Example: Display the Current Preallocation List

This example shows how to display the current preallocation list of data sets.

```
OPER PREALLOC
```

Example: Preallocate a Data Set

This example shows how to preallocate data set CYBER.JCL.CNTL.

```
OPER PREALLOC CYBER.JCL.CNTL ALLOC
```

Example: Unallocate a Data Set

This example shows how to unallocate data set CYBER.ZAG.PROCS.

```
OPER PREALLOC CYBER.ZAG.PROCS UNALLOC
```

QUIESCE Command: Quiesce z/OS Agent

To put z/OS Agent into a quiesced state, use the QUIESCE command. While z/OS Agent is in a quiesced state, event scheduling and application processing are also quiesced. Jobs that are running when z/OS Agent is quiesced continue to run. No new jobs are submitted.

Type: OPER command

Authority: To issue the QUIESCE command, you require OPER authority.

This command has the following format:

```
QUIESCE
```

If z/OS Agent is being restarted, you can bring it up in a quiesced state by using the QUIESCE option on the start command.

Related Information

For information on taking z/OS Agent out of the quiesced state, see the RESTART Command (see page 61).

For information on displaying the processing status of z/OS Agent, see the STATUS Command (see page 64).

Example: Quiesce z/OS Agent

This example shows how to quiesce z/OS Agent.

```
OPER QUIESCE
```

RESTART Command: Restart z/OS Agent

To restart z/OS Agent out of a quiesced state, use the RESTART command. The RESTART command also restarts event scheduling and application processing.

Type: OPER command

Authority: To issue the RESTART command, you require OPER authority.

This command has the following format:

```
RESTART
```

Use the RESTART command after z/OS Agent has been quiesced.

z/OS Agent acts as if it is down for the entire time when it is in a quiesced state. Events that were scheduled to execute are considered to be overdue when the event scheduling is restarted. The overdue operand of each overdue event is examined to determine whether the event executes or is bypassed.

Related Information

For information on putting z/OS Agent into a quiesced state, see the QUIESCE Command (see page 61).

For information on displaying the processing status of z/OS Agent, see the STATUS Command (see page 64).

Example: Restart z/OS Agent

This example shows how to restart z/OS Agent. The RESTART command also restarts the event scheduling and application processing.

```
OPER RESTART
```

You can display the processing status with the STATUS command:

```
OPER STATUS
ESP1977I z/OS Agent RELEASE x.x.x.000 STATUS
ESP1978I EVENT SCHEDULING ACTIVE
ESP1979I DATASET TRIGGERING ACTIVE
```

ROUTING Command: Route Proxy Requests

The ROUTING command instructs the connected z/OS Agent where to process requests. The ROUTING command also displays the current routing status.

Type: CA WA High Availability ESP Edition or CA WA Service Governor ESP Edition command

This command has the following format:

```
ROUTING [LOCAL|MASTER|STATUS|HELP]
```

LOCAL

Instructs the connected z/OS Agent to process requests itself.

MASTER

Instructs the connected z/OS Agent to route requests to the master.

STATUS

Displays the current ROUTING status of the client. STATUS is the default.

HELP

Displays the ROUTING command options.

Note:

- When a TSO session is connected to the master, the ROUTING command is not meaningful, as the LOCAL and MASTER options mean the same thing.
- When a TSO session is connected to a proxy, entering ROUTING LOCAL instructs the proxy to process subsystem requests (for example, z/OS Agent commands) itself. Entering ROUTING MASTER instructs the proxy to route subsystem requests to the master.
- Routing is local by default.
- When a TSO session is connected to the local proxy, subsystem requests can only be forwarded to the master if an active XCF ROUTING connection exists between the local proxy and the master.

Example: Route Subsystem Requests to the Master

This example shows how to instruct the connected subsystem to route subsystem requests to the master.

```
ROUTING MASTER
```

Example: Process Subsystem Requests

This example shows how to instruct the connected subsystem to process subsystem requests itself.

```
ROUTING LOCAL
```

SPINLOG Command—Spin a Log

To spin a selected log to a sysout class, use the SPINLOG command.

Type: OPER command

Authority: To issue the SPINLOG command, you require OPER authority.

This command has the following format:

```
SPINLOG [AUDITLOG|ESPLOG|ENCORELG|CPEENQLG]
```

AUDITLOG

Spin the audit log.

ENCORELG

Spin the log of RQEN (ENCREQ) IARs.

CPEENQLG

Spin the log of RQEN (CPEENQ) IARs.

Audit Log

A SPINLOG command with no operands acts upon z/OS Agent audit log. The AUDITLOG initialization parameter identifies the sysout class, and optionally the form number for spinning the audit log. An automatic spinlog exists for the audit log that is set with the AUDITLOG initialization parameter.

Related Information

For information on setting the print characteristics of z/OS Agent audit log, see the AUDITLOG initialization parameter.

Example: Display the SPINLOG

This example shows how to display the SPINLOG. In the example, 262 records from the audit log are spun to sysout class E.

```
OPER SPINLOG
ESP1023I Auditlog dataset spun with 262 records to sysout class E
```

STATUS Command: Display the Processing Status

To display the processing status of z/OS Agent, use the STATUS command.

Type: General command

Authority: To issue the STATUS command, you require OPER authority.

This command has the following format:

```
{STATUS|ST}
```

The STATUS command displays the following information:

- The current z/OS Agent release number.
- Whether z/OS Agent is quiesced or active.
- Whether data set triggering is suspended or active.
- The names of any suspended Event data sets.

Example: Display the Processing Status

This example shows how to display the status of z/OS Agent, event scheduling, and data set triggering.

```
OPER STATUS
ESP1977I z/OS Agent RELEASE x.x.x.000 STATUS
ESP1978I EVENT SCHEDULING ACTIVE
ESP1979I DATASET TRIGGERING ACTIVE
```

TRACE Command: Activate Trace Facility

To activate the trace facility and set options, use the TRACE command.

Type: OPER command

Authority: To issue the TRACE command, you require OPER authority.

This command has the following format:

```
TRACE [SET(id[:id])]
      [RESET(id[:id])]
      [SWITCH|CLOSE|OPEN|STATUS|WRITE]
      [REUSE]
```

SET(*id*)

Indicates one or more record identifiers you want to trace. You can specify a list of identifiers and a range of identifiers.

RESET(*id*)

Indicates one or more record identifiers you no longer want to trace.

SWITCH

Requests the data set currently in use for the trace facility be freed. The next trace data set is activated automatically. With this option, data sets are activated in the sequence in which they were first defined. Once the last data set in the sequence is closed, the first trace data set to be defined is used.

CLOSE

Requests the closure of the trace data set currently in use, but prevents switching to the next trace data set. The buffers that were defined using TRACEDEF automatically continue to hold trace data in the core until a TRACE OPEN command is issued.

OPEN

Requests opening of the trace data set most recently active. Use this operand after TRACE CLOSE is issued to reactivate the same trace data set.

REUSE

Indicates to checkpoint the data set currently in use for the trace facility. A checkpoint enables subsequent writes to begin at the checkpoint rather than at the start of the data set.

WRITE

Writes to the trace data set to clear any buffered data.

STATUS

Displays information indicating where the trace facility is active. Display includes how many records were written since trace was activated, the current trace data sets in use, and which data set is active.

The TRACE command is useful as a problem-solving tool. CA Support can ask you to SET a specific trace record ID to provide information to help with troubleshooting.

If you want to capture only records relating to event processing (for example, type 601), use the ESPPARM AUDITLOG ddname in z/OS Agent started task procedure or the AUDITLOG initialization parameter. This allows the use of a preallocated SYSOUT for the event activity and eliminates the need to use the TRACEDEF and TRACE commands.

Related Information

- For information on printing trace data, see the TRACEPRT Command (see page 68).
- For information on displaying event log data collected in the trace data set, see the LOGPRT Command (see page 51).

Example: Activate the Trace Facility

This example shows how to activate a trace and specify that record IDs 602 through 604, and 607 be traced.

```
OPER TRACE SET(602:604,607)
```

Example: Checkpoint the Data Set Currently in Use

This example shows how to checkpoint the data set currently in use for the trace facility before switching to the next trace data set. When the current trace data set is later reused, records are written starting at the checkpoint.

```
OPER TRACE SWITCH REUSE
```

Example: Turn Off the Trace Facility for Specific Records

This example shows how to turn the trace off for record Ids 602 through 604.

```
OPER TRACE RESET(602:604)
```

Example: Write Out Buffered Data and Close the Trace Data Set

This example shows how to write out the buffered data and close the trace data set. Writing out the buffered data and closing the trace data set is done before printing the trace data set.

```
OPER TRACE WRITE
OPER TRACE CLOSE
```

TRACEDEF Command: Identify Data Sets

To identify data sets to record the information the trace facility collects, use the TRACEDEF command.

Type: OPER command

Authority: To issue the TRACEDEF command, you require OPER authority.

This command has the following format:

```
TRACEDEF DSN(dsname[,dsname]...)
         [BUF(size,count)]
```

dsname

Indicates the name of one or more data sets used as trace data sets. Separate the data set names with a blank or comma. Specifying multiple trace data sets allows you to use the SWITCH operand with TRACE to free up one trace data set and switch to another.

size, count

Indicates the buffers that you want to use for the trace data sets you define. Size specifies the buffer size and count identifies how many buffers are required. This defaults to (4096,4)—(four buffers, each one 4096 bytes).

Before you use the trace facility, allocate one or more trace data sets. Identify these data sets to z/OS Agent with the TRACEDEF command.

You do not have to specify any DCB attributes when you initially allocate the data sets, because z/OS Agent does this task automatically.

Trace data sets use the following attributes: DCB=(RECFM=VB, LRECL=4096, BLKSIZE=4100).

The buffers that you define continue to hold trace data until each one becomes full. Once full, the data is written automatically to the trace data set and another buffer is used.

Related Information

For information on defining a trace data set at the initialization parameter level, see the *Installation and Configuration Guide.*

For information on activating the trace facility, see the TRACE Command. (see page 65)

Example: Define Two Trace Data Sets

This example shows how to define two trace data sets that each have three 23400-byte buffers.

```
TRACEDEF DSN('ZAGCYB.TRACE1', 'ZAGCYB.TRACE2') BUF(23400,3)
```

TRACEPRT Command: Print Trace Data

To print trace data, use the TRACEPRT command.

Type: General command

Authority: To issue the TRACEPRT command, you require OPER authority.

This command has the following format:

```
TRACEPRT DSN(dsn[,dsn])
         [FROM(criteria)]
         [SELECT(id[:id])]
         [APPL(appl[,appl...])]
         [JOB(job[,job...])]
```

dsn

Indicates the name of one or more trace data sets.

criteria

Indicates any valid schedule criteria that represent a time range.

id

Indicates a trace ID to print. Ranges are supported.

appl

Allows the selection by application name. Wildcards are supported.

job

Allows the selection by jobname. Wildcards are supported.

Issue the TRACEPRT command in batch or from line mode, after collecting trace data from a TRACE command.

Note: For diagnostic purposes, the contents of the raw trace data set are required.

Related Information

For information on using the TRACE command, see the TRACE Command (see page 65).

Example: Print a Trace Data Set

This example shows how to print CYBER.TRACE1 from 6 am yesterday to 6 am today.

```
TRACEPRT DSN('CYBER.TRACE1') FROM('6AM YESTERDAY UNTIL 6AM TODAY')
```

Example: Print Trace Data Set Based on an Application

This example shows how to print trace data set information for the PAYROLL application.

```
TRACEPRT DSN('CYBER.TRACE1') APPL(PAYROLL)
```

Example: Print a Range of Trace IDs

This example shows how to print trace IDs 601 - 603.

```
TRACEPRT DSN('CYBER.TRACE1') SELECT(601:603)
```

XCF DISPLAY Command: View XCF Attributes

To view the system attributes of your XCF group, use the XCF DISPLAY command.

Type: XCF command

Entering Commands

You can issue all XCF commands using the following methods:

- Line mode
- z/OS MODIFY command

This command has the following format:

```
XCF {DISPLAY|D} {ACTIVE|A}
                {GROUP|G}
                {SERVICE|S}
                {SYSTEM|SYS}
                {TRACE|TR}
                {MC|M}
```

ACTIVE

Displays the active XCF service listener on the master and the connection.

GROUP

Displays the status of your XCF group and its members.

SERVICE

Displays the status of the XCF services in the group.

SYSTEM

Displays all systems in the sysplex.

TRACE

Displays the status of the trace log.

MC

Displays two additional fields, MC (must-complete count) and HMK (must-complete high mark), in the response for the XCF connections.

Example: Issue the DISPLAY ACTIVE Command

The following example shows the results of the DISPLAY ACTIVE command when entered from a master. The example uses the short form of the DISPLAY ACTIVE command.

```
XCF D A
```

```
Group=QS10, Member=QS10M
Partner          Service  Status  Connection
Listener         DSTRIG   Active
Listener         ROUTING  Active
Listener         TRACKING Active
QS10S            DSTRIG   Active  1,49
QS10S            ROUTING  Active  3,50
QS10S            TRACKING Active  7,52
QS10S2           DSTRIG   Active  2,48
QS10S2           ROUTING  Active  4,49
QS10S2           TRACKING Active  8,51
```

The following example shows the results of the DISPLAY ACTIVE command when entered from a proxy. The example uses the short form of the DISPLAY ACTIVE command.

```
XCF D A
```

```
Group=QS10, Member=QS10S
Partner          Service  Status  Connection
QS10M3           DSTRIG   Active  49,1
QS10M3           ROUTING  Active  50,3
QS10M3           TRACKING Active  52,7
```

Example: Issue the DISPLAY GROUP Command

The following example shows the results of the DISPLAY GROUP command when entered from a master. The example uses the short form of the DISPLAY GROUP command.

```
XCF D G
```

```
Group=QS10 Member=QS10M3 Interval=60 TermOpt=Quiesce
Group Member System ASID Jobname  SSID ESP    Status
QS10  QS10M1  SYSC  0056 QS10     QS10 Master  Active
QS10  QS10M3  SYSA  009B QS10SHAD QS10 Shadow  Active
QS10  QS10M5  SYSC  0041 QS10SMC  QS10 Shadow  Active
QS10  QS10S   SYSA  009F QS10S    QS1S Proxy   Active
QS10  QS10S2  SYSC  0046 QS10S2   QS12 Proxy   Active
```

The following example shows the results of the DISPLAY GROUP command when entered from a proxy. The example uses the short form of the DISPLAY GROUP command.

```
XCF D G

Group=QS10 Member=QS10S Interval=45 TermOpt=Quiesce
Group Member System ASID Jobname  SSID ESP    Status
QS10  QS10M1  SYSC  0056 QS10     QS10 Master  Active
QS10  QS10M3  SYSA  009B QS10SHAD QS10 Shadow  Active
QS10  QS10M5  SYSC  0041 QS10SMC  QS10 Shadow  Active
QS10  QS10S   SYSA  009F QS10S    QS1S Proxy   Active
QS10  QS10S2  SYSC  0046 QS10S2   QS12 Proxy   Active
```

Example: Issue the DISPLAY SERVICE Command

The following example shows the results of the DISPLAY SERVICE command when entered from a master. The example uses the short form of the DISPLAY SERVICE command.

```
XCF D S

Service   Status   Group    Member     Susp
DSTRIG    Active   QS10     QS10M      120
ROUTING   Active   QS10     QS10M      120
TRACKING  Active   QS10     QS10M      120
```

The following example shows the results of the DISPLAY SERVICE command when entered from a proxy. The example uses the short form of the DISPLAY SERVICE command.

```
XCF D S

Service   Status   Group    Member   Susp
DSTRIG    Active   QS10     QS10S     120
ROUTING   Active   QS10     QS10S     100
TRACKING  Active   QS10     QS10S     100
```

Example: Issue the DISPLAY SYSTEM Command

The following example shows the results of the DISPLAY SYSTEM command. The example uses the short form of the DISPLAY SYSTEM command.

```
XCF D SYS

Sysplex=SYSAPLEX, System=SYSA
System   Status      Interval
SYSA     Active            85
SYSC     Active            85
```

Example: Issue the DISPLAY TRACE Command

The following example shows how to issue the short form of the DISPLAY TRACE command.

```
XCF D TR
```

Example: Display the Must-Complete Count

The following example shows the must-complete count (MC) column and the must-complete high mark (HMK) column in the response for the XCF connections.

```
 XCF D A M

 Group=ESP, Member=ESPMCAxx
 Partner          Service  Status  Connection  MC HMK
 Listener         DSTRIG   Active  1
 Listener         ROUTING  Active  2
 Listener         TRACKING Active  4
 ESPPCAxx         DSTRIG   Active  5,1          0   0
 ESPPCAxx         ROUTING  Active  6,2          0   0
 ESPPCAxx         TRACKING Active  8,4          0   1
```

XCF SET TRACE Command: Specify XCF Trace Parameters

The XCF SET TRACE command specifies the parameters for the XCF trace facility. XCF tracing is normally used at the request of CA Support to diagnose an XCF problem. XCF SET TRACE can be coded in the initialization parameter data set or entered on the fly when the trace data is required. We recommend you put XCF SET TRACE in your initialization parameter data set.

Type: XCF command

Note: You can issue all XCF commands using the following methods:

- Line mode
- z/OS MODIFY command

This command has the following format:

```
XCF {SET|T} {TRACE|TR} {SYSOUT|S}(class)
                       [{MEMBER|ME}(member)|{ALLMEMBERS|ALLM}]
                       [{SERVICE|SE}(service)|{ALLSERVICES|ALLS}]
                       [{REQUESTS|R}|{NOREQUESTS|NOR}]
                       [{MSGEVENTS|MS}|{NOMSGEVENTS|NOM}]
                       [{GRPEVENTS|G}|{NOGRPEVENTS|NOG}]
                       [{SYSEVENTS|SYSE}|{NOSYSEVENTS|NOS}]
                       [{DATA|D}|{NODATA|NOD}}]
                       [{MCTHRESHOLD|MC}(threshold)]
```

SYSOUT(*class*)

Indicates an output class for sysout.

MEMBER(*member*)

Requests tracing for the specified XCF group member only.

ALLMEMBERS

Requests tracing for all XCF group members.

SERVICE(*service*)

Requests tracing for the specified XCF service only. The only valid XCF services are: TRACKING, DSTRIG, and ROUTING. The ROUTING service is only valid when the following are enabled: CA WA High Availability ESP Edition or CA WA Service Governor ESP Edition.

ALLSERVICES

Requests tracing for all XCF services.

REQUESTS

Requests tracing of all XCF API calls (IXCJOIN, IXCLEAVE, IXCMSGI, IXCMSGO, IXCQUERY, IXCSETUS etc.) when the XCF TRACE DATA option is active. Also requests tracing of all XCF API calls except for data transfers (IXCMSGO Send, IXCMSGO Send Response, and IXCMSGI) when the XCF TRACE DATA option is not active.

NOREQUESTS

Requests no XCF tracing of "IXCMSGO Send", "IXCMSGO Send Response", and "IXCMSGI" XCF API calls.

MSGEVENTS

Requests tracing of XCF message events, provided the XCF SET TRACE DATA options is active.

NOMSGEVENTS

Requests no tracing of XCF message events.

GRPEVENTS

Requests tracing of XCF group events.

NOGRPEVENTS

Requests no tracing of XCF group events.

SYSEVENTS

Requests tracing of XCF system events.

NOSYSEVENTS

Requests no tracing of XCF system events.

DATA

Requests tracing of "IXCMSGO Send", "IXCMSGO Send Response", and "IXCMSGI" XCF API calls, provided the XCF SET TRACE "REQUESTS" options is active. Also requests tracing of XCF message events, provided the XCF SET TRACE "MSGEVENTS" option is active.

NODATA

Requests no tracing of the following XCF API calls: "IXCMSGO Send", "IXCMSGO Send Response", and "IXCMSGI". Also requests no tracing of XCF message events.

MCTHRESHOLD(*threshold*)

Requests XCF tracing of must-complete XCF Sends and XCF Receives that result in an XCF connection must-complete count being incremented to the specified threshold value or higher. Also requests XCF tracing of must-complete XCF Send responses and XCF Receive responses that result in the XCF connection must-complete count being decremented from the specified.

Example: Set trace data

The following example shows the short form of the SET TRACE command to set trace data to a sysout class of X:

```
XCF T TR S(X)
```

XCF START TRACE Command: Initialize/Activate Trace Files

The XCF START TRACE command initializes and activates the XCF trace file. You can code the XCF START TRACE initialization parameter after the XCF SET TRACE initialization parameter in the initialization parameter data set. Otherwise, the XCF START TRACE command is entered on the fly when you must capture data.

Type

XCF command

Entering commands

You can issue all XCF commands using the following methods:

- Line mode
- z/OS MODIFY command

This command has the following format:

```
XCF {START|S} {TRACE|TR}
```

TRACE

Indicates the XCF Trace facility.

Example: Short form of the START TRACE command

The following example shows the short form of the START TRACE command:

```
XCF S TR
```

XCF STOP TRACE Command: Stop Trace File

The XCF STOP TRACE command stops the trace file after data has been captured and spools the captured data to the preset sysout class specified by the XCF SET TRACE command.

Type

XCF command

Entering commands

You can issue all XCF commands using the following methods:

- Line mode
- z/OS MODIFY command

This command has the following format:

```
XCF {STOP|P} {TRACE|TR}
```

TRACE

Indicates the XCF Trace facility.

Example: Short form of the STOP TRACE command

The following example shows the short form of the XCF STOP TRACE command:

```
XCF P TR
```

XDAB Command: Trace VSAM I/O Events

To trace VSAM I/O events and send the trace data to a sysout data set, use the XDAB command.

Note: XDAB is a diagnostic tool. Do not use under normal operating conditions.

Type: OPER command

Authority: To issue the XDAB command, you require OPER authority.

This command has the following format:

```
XDAB SET|T TRACE|TR
     [SYSOUT(class)]|[LOGTRACE]
     [ASYNC|ALL]
     [LOCK|NOLOCK]
     [RESERVE|NORESERVE]

XDAB {START|S}
     {STOP|P}
     {RESTART|RES|RS|SPIN|SP}
     {DISPLAY|D}
     TRACE|TR

XDAB HELP|H|?
```

SYSOUT(*class*)

Send VSAM I/O trace data to a sysout data set with DDNAME DABTRACE and to the class that *class* specifies.

LOGTRACE

Send VSAM I/O trace data to a sysout data set with DDNAME LOGTRACE. The trace data is consolidated with certain other trace data.

ASYNC

Include only asynchronous VSAM I/Os in the trace data.

ALL

Include asynchronous and synchronous VSAM I/Os in the trace data.

LOCK

Include internal VSAM data set locking and unlocking events in the trace data.

<u>NOLOCK</u>

Exclude internal VSAM data set locking and unlocking events from the trace data.

RESERVE

Include VSAM data set cross-system serialization events in the trace data.

<u>NORESERVE</u>

Exclude VSAM data set cross-system serialization events from the trace data.

START|S

Start the VSAM I/O tracing.

STOP|P

Stop the VSAM I/O tracing.

RESTART|RES|RS|SPIN|SP

Stop and restart the VSAM I/O tracing.

DISPLAY|D

Display the current XDAB status.

HELP|H|?

Display the XDAB command operands.

Set Up the Trace

Do this task before you start the trace. Set up the trace by issuing XDAB with at least the SYSOUT or LOGTRACE operands. If you issue XDAB with the LOGTRACE operand, set up the trace data consolidation.

Start Trace Data Consolidation

When you specify the LOGTRACE operand, data for the XDAB trace is not generated. Additionally, data for the XDAB trace is not consolidated with data from other traces until you start the XDAB trace and start the trace consolidation.

Displaying Trace Settings

Issue XDAB SET TRACE alone to display the current settings for the XDAB tracing.

Asynchronous I/Os

Some VSAM data set I/Os, such as the JOBINDEX file I/Os, are asynchronous for performance reasons. z/OS Agent main task is not suspended when a wait is required for an asynchronous I/O completion.

Related Information

- See the XDAB initialization parameter in the *Initialization and Configuration Guide*.
- See the LOGTRACE Command (see page 52).

Example: Display the Current Settings for XDAB Tracing

These two examples show how to display XDAB trace settings.

```
XDAB DISPLAY TRACE
XDAB D TR
```

Example: Set the Sysout Class for XDAB Trace Data

This example shows how to specify that trace data goes to DDNAME DABTRACE with the sysout class set to A.

```
XDAB SET TRACE SYSOUT(A)
```

Example: Consolidate XDAB Trace Data with Other Trace Data

This example shows how to specify that trace data goes to the DDNAME LOGTRACE, and the trace data is consolidated with certain other trace data.

```
XDAB SET TRACE LOGTRACE
```

Example: Filter XDAB Trace Data

This example shows how to limit the trace data to asynchronous VSAM I/Os and include VSAM data set cross-system serialization events in the trace.

```
XDAB SET TRACE ASYNC RESERVE
```

Example: Control the XDAB Trace

This example shows how to start the XDAB trace.

```
XDAB START TRACE
```

This example shows how to restart the XDAB trace.

```
XDAB SP TRACE
```

Example: Get Help

This example shows how to display the operands for the XDAB command.

```
XDAB ?
```

XFRB Command: Trace Slot I/O Events

To trace slot I/O events and send the trace data to a sysout data set, use the XFRB command.

Note: XFRB is a diagnostic tool. Do not use under normal operating conditions.

Type: OPER command

Authority: To issue the XFRB command, you require OPER authority.

This command has the following format:

```
XFRB SET|T TRACE|TR
     [SYSOUT(class)]|[LOGTRACE]
     [ASYNC|ALL]
     [LOCK|NOLOCK]
     [RESERVE|NORESERVE]

XFRB {START|S}
     {STOP|P}
     {RESTART|RES|RS|SPIN|SP}
     {DISPLAY|D}
     TRACE|TR

XFRB HELP|H|?
```

SYSOUT(*class*)

Send slot I/O trace data to a sysout data set with DDNAME FRBTRACE and to the class that *class* specifies.

LOGTRACE

Send slot I/O trace data to a sysout data set with DDNAME LOGTRACE. The trace data is consolidated with certain other trace data.

ASYNC

Include only asynchronous slot I/Os in the trace data.

ALL

Include asynchronous and synchronous slot I/Os in the trace data.

LOCK

Include internal slot data set locking and unlocking events in the trace data.

NOLOCK

Exclude internal slot data set locking and unlocking events from the trace data.

RESERVE

Include slot data set cross-system serialization events in the trace data.

NORESERVE

Exclude slot data set cross-system serialization events from the trace data.

START|S

Start the slot I/O tracing.

STOP|P

Stop the slot I/O tracing.

RESTART|RES|RS|SPIN|SP

Stop and restart the slot I/O tracing.

DISPLAY|D

Display the current XFRB status.

HELP|H|?

Display the XFRB command operands.

Set Up the Trace

Do this task before you start the trace. Set up the trace by issuing XFRB with at least the SYSOUT or LOGTRACE operands. If you issue XFRB with the LOGTRACE operand, set up the trace data consolidation.

Start Trace Data Consolidation

When you specify the LOGTRACE operand, data for the XFRB trace is not generated. Additionally, data for the XFRB trace is not consolidated with data from other traces until you start the XFRB trace and start trace consolidation.

Displaying Trace Settings

Issue XFRB SET TRACE alone to display the current settings for the XFRB tracing.

Asynchronous I/Os

Some slot data set I/Os, such as the TRAKFILE I/Os, are asynchronous for performance reasons. z/OS Agent main task is not suspended when a wait is required for an asynchronous I/O completion.

Related Information

- See the XFRB initialization parameter in the *Initialization and Configuration Guide*.
- See the LOGTRACE Command (see page 52).

Example: Display the Current Settings for XFRB Tracing

These two examples show how to display XFRB trace settings.

```
XFRB DISPLAY TRACE
XFRB D TR
```

Example: Set the Sysout Class for XFRB Trace Data

This example shows how to specify that trace data goes to DDNAME FRBTRACE with the sysout class set to A.

```
XFRB SET TRACE SYSOUT(A)
```

Example: Consolidate XFRB Trace Data with Other Trace Data

This example shows how to specify that trace data goes to the DDNAME LOGTRACE, and the trace data is consolidated with certain other trace data.

```
XFRB SET TRACE LOGTRACE
```

Example: Filter XFRB Trace Data

This example shows how to limit the trace data to the asynchronous slot I/Os and include slot data set cross-system serialization events in the trace.

```
XFRB SET TRACE ASYNC RESERVE
```

Example: Control the XFRB Trace

This example shows how to start the XFRB trace.

```
XFRB START TRACE
```

This example shows how to restart the XFRB trace.

```
XFRB SP TRACE
```

Example: Get Help

This example shows how to display the operands for the XFRB command.

```
XFRB  ?
```