System i

Programming
i5/OS commands
Starting with ADDACC (Add Access Code)

Version 6 Release 1

System i

Programming
i5/OS commands
Starting with ADDACC (Add Access Code)

Version 6 Release 1

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 639.

This edition applies to version 6, release 1, modification 0 of IBM i5/OS (product number 5761-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CICS models.

Contents

Add Access Code (ADDACC)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Access Code (ADDACC) command allows you to define a new access code to the system for later use when securing a document or folder, or on commands dealing with object authority or access authority. The Add Access Code (ADDACC) command identifies both the access code added and the descriptive text that is associated with the access code.

Restrictions:

To use this command, you must have all object (*ALLOBJ) authority.

Top

Parameters

Keyword	Description	Choices	Notes
ACC	Document access code	1-2047, *AVAIL	Required, Positional 1
TEXT	Text 'description'	*Character value*	Required, Positional 2

Top

Document access code (ACC)

Specifies the access code that is added to the system. The access code must not be currently defined on the system.

This is a required parameter.

***AVAIL**
The system chooses the next free access code and adds that access code to the system; you must specify the text. The access code selected is returned to you in a completion message.

1-2047 Specify an access code to be added to the system.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the object.

This is a required parameter.

character
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Examples

Example 1: Adding an Access Code

```
ADDACC   ACC(700)  TEXT('programmers')
```

This command adds access code 700 to the system. This access code is authorized to all programmers (after the GRTACCAUT command is run). It is used when filing documents to which all programmers may have access.

Example 2: Adding Next Available Access Code

```
ADDACC   ACC(*AVAIL)  TEXT('department managers')
```

This command adds the next available access code to the system. This access code is authorized to all department managers (after the GRTACCAUT command is run). It is used when placing in the document library objects to which all department managers may have access. The system returns a message containing the access code that was being used.

Top

Error messages

***ESCAPE Messages**

CPF897B
Mail Log Conversion failed.

CPF9001
Add access code request failed.

CPF9009
System requires file &1 in &2 be journaled.

CPF9845
Error occurred while opening file &1.

CPF9846
Error while processing file &1 in library &2.

CPF9847
Error occurred while closing file &1 in library &2.

Top

Add Autostart Job Entry (ADDAJE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Autostart Job Entry (ADDAJE) command adds an autostart job entry to the specified subsystem description. The entry identifies the job name and the job description to be used to automatically start a job.

Restrictions:

1. To use this command, you must have:
   - object operational (*OBJOPR), object management (*OBJMGT), and read (*READ) authority to the specified subsystem description and execute (*EXECUTE) authority to the library containing that subsystem description.
   - object operational (*OBJOPR) and read (*READ) authority to the job description and execute (*EXECUTE) authority to the library containing that job description.
2. Only a user with all object (*ALLOBJ) special authority is allowed to add an entry for which the job description does not exist.

Top

Parameters

Keyword	Description	Choices	Notes
SBSD	Subsystem description	*Qualified object name*	Required, Positional 1
	Qualifier 1: Subsystem description	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
JOB	Job name	*Name*	Required, Positional 2
JOBD	Job description	Single values: ***SBSD** Other values: *Qualified object name*	Optional
	Qualifier 1: Job description	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

Subsystem description (SBSD)

Specifies the name and library of the subsystem description where the autostart job entry is being added.

This is a required parameter.

Qualifier 1: Subsystem description

name Specify the name of the subsystem description where the autostart job entry is being added.

Note: The IBM-supplied object QSYSSBSD is not valid on this parameter.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the library where the subsystem description is located.

Top

Job name (JOB)

Specifies the simple name of the job that is automatically started when a subsystem is started using the subsystem description specified on the **Subsystem description (SBSD)** parameter.

This is a required parameter.

name Specify the simple name of the job that is automatically started.

Top

Job description (JOBD)

Specifies the name of the job description used for the job that is started by this autostart job entry. If the job description does not exist when the entry is added or changed, a library qualifier must be specified because the qualified job description name is kept in the subsystem description.

Note: Only a user with all object (*ALLOBJ) special authority is allowed to add or change an entry for which the job description does not exist.

Single values

***SBSD**
The job description having the same name and library as the subsystem description, specified on the **Subsystem description (SBSD)** parameter, is used for the job that is being started.

Qualifier 1: Job description

name Specify the name of the job description that is used for the job being started by this autostart job entry.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the library where the job description is located.

Top

Examples

```
ADDAJE   SBSD(ACCTLIB/ACCTINT) JOB(ACCTINIT) JOBD(ACCTLIB/INITSBS)
```

This command adds an autostart job entry for the job ACCTINIT subsystem description ACCTINT in the library ACCTLIB. In this case, the job that starts automatically might be used to perform certain routines whenever the subsystem ACCTINT is started. When the subsystem is started, the job description INITSBS in ACCTLIB is used to obtain the attributes for this job and a job named ACCTINIT is automatically started in the subsystem.

Top

Error messages

***ESCAPE Messages**

CPF1619
Subsystem description &1 in library &2 damaged.

CPF1697
Subsystem description &1 not changed.

Top

Add Alert Action Entry (ADDALRACNE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Alert Action Entry (ADDALRACNE) command allows the user to add an action entry to the specified alert filter. This entry describes the actions that should be taken for an alert that has been assigned to the specified group. More information on alerts is in the Alerts Support book, SC41-5413.

Top

Parameters

Keyword	Description	Choices	Notes
FILTER	Filter	*Qualified object name*	Required, Positional 1
	Qualifier 1: Filter	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
GROUP	Group	*Name*	Required, Positional 2
LOG	Log alert	*YES, *NO, ***NETATR**	Optional
ASNUSER	User assigned	*Character value*, ***NONE**	Optional
SEND	Send to system	Single values: ***NONE** Other values (up to 5 repetitions): *Element list*	Optional
	Element 1: Network identifier	*Communications name*, *NETATR, *FOCALPT	
	Element 2: Control point	*Communications name*	
SNDDTAQ	Send to data queue	Single values: ***NONE** Other values (up to 5 repetitions): *Element list*	Optional
	Element 1: Data queue	*Qualified object name*	
	Qualifier 1: Data queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 2: Data queue key	*Character value*, ***NONE**, X''	
GENTRAP	Generate SNMP trap	***NO**, *YES	Optional

Top

Filter (FILTER)

Specifies the filter to which the action entry is to be added.

This is a required parameter.

Qualifier 1: Filter

name Specify the name of the filter.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the filter. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the filter is located.

Top

Group (GROUP)

Specifies the group name to which the defined actions are to be applied. The group name is assigned from the selection criteria in the filter.

This is a required parameter.

name Specify the group name.

Top

Log alert (LOG)

Specifies whether the alert is logged.

***NETATR**
The ALRLOGSTS network attribute controls the logging of this alert.

***YES** The alert is logged.

***NO** The alert is not logged.

Top

User assigned (ASNUSER)

Specifies the user to be assigned to the alert.

***NONE**
No user is specified.

character-value
Specify a user name.

Top

Send to system (SEND)

Specifies the destination to which the alert is to be sent. An alert cannot be sent to the local system. The system checks for this action when the alert is sent.

Single values

***NONE**
The alert is not sent.

Other values (up to 5 repetitions)

Element 1: Network identifier

***FOCALPT**
Sends the alert to the system focal point. The focal point system is determined at send time.

***NETATR**
The LCLNETID value specified in the system network attributes is used.

communications-name
Specify the network ID of the destination node.

Element 2: Control point

communications-name
Specify the control point name of the destination system.

Top

Send to data queue (SNDDTAQ)

Specifies the data queue in which an alert notification record is placed. Keyed data queues are supported.

The data queue name can be qualified by one of the following library values:

Single values

***NONE**
No data queue is used.

Other values (up to 5 repetitions)

Element 1: Data queue

Qualifier 1: Data queue

name Specify the name of the data queue.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the data queue. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the data queue is located.

Element 2: Data queue key

***NONE**
No key is used on the data queue.

character-value
Specify the data queue key.

Top

Generate SNMP trap (GENTRAP)

Specifies whether the alert generates an SNMP trap.

***NO** An SNMP trap is not generated from this alert.

***YES** An SNMP trap is generated from this alert.

Top

Examples

```
ADDALRACNE   FILTER(MYLIB/MYFILTER)  GROUP(CHICAGO)
             LOG(*NETATR)  ASNUSER(CHICAGOOPR)
             SEND((*FOCALPT) (*NETATR.MILWKEE))
             SNDDTAQ(*LIBL/ALERTDTAQ)
```

This command defines the following actions for group CHICAGO:

1. Log the alert based on the ALRLOGSTS network attribute.
2. Send the alert to this system's focal point.
3. Send the alert to the system with control point name MILWKEE and a network id based on the LCLNETID value specified in the system network attributes.
4. Place an alert notification on data queue ALERTDTAQ.
5. Assign the alert to user CHICAGOOPR.

Top

Error messages

***ESCAPE Messages**

CPF2150
Object information function failed.

CPF2151
Operation failed for &2 in &1 type *&3.

CPF812F
Filter damaged.

CPF91DB
Group &4 already exists.

CPF91DE
Filter &1/&2 at maximum size.

CPF91EB
Filter type &3 not correct for this operation.

CPF91EC
Internal processing error occurred.

CPF91E8
Internal processing error occurred.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9830
Cannot assign library &1.

Top

Add Alert Description (ADDALRD)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Alert Description (ADDALRD) command adds an alert description to the specified alert table.

Top

Parameters

Keyword	Description	Choices	Notes
MSGID	Message identifier	*Name*	Required, Positional 1
ALRTBL	Alert table	*Qualified object name*	Required, Positional 2
	Qualifier 1: Alert table	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
ALRTYPE	Alert type	00-F0, ***NONE**	Optional
ALRD	Alert description code point	0000-FFF0, ***NONE**	Optional
PBLCAUSE	Probable cause code point	Single values: ***NONE** Other values (up to 99 repetitions): 0000-FFF0	Optional
CAUSE	Cause	Single values: ***NONE** Other values (up to 99 repetitions): *Element list*	Optional
	Element 1: Cause type	*USER, *INSTALL, *FAILURE	
	Element 2: Cause code point	*Hexadecimal value*	
	Element 3: Detail data ID 1	*Hexadecimal value*, ***NONE**	
	Element 4: Detail data 1	*Character value*, ***NODATA**	
	Element 5: Detail data ID 2	*Hexadecimal value*, ***NONE**	
	Element 6: Detail data 2	*Character value*, ***NODATA**	
	Element 7: Detail data ID 3	*Hexadecimal value*, ***NONE**	
	Element 8: Detail data 3	*Character value*, ***NODATA**	
	Element 9: Product identifier	***NONE**, *SNDHDW, *SNDSFW, *RSCHDW	
ACTION	Recommended action	Single values: ***NONE** Other values (up to 99 repetitions): *Element list*	Optional
	Element 1: Action type	*USER, *INSTALL, *FAILURE, *UNKNOWN	
	Element 2: Action code point	*Hexadecimal value*	
	Element 3: Detail data ID 1	*Hexadecimal value*, ***NONE**	
	Element 4: Detail data 1	*Character value*, ***NODATA**	
	Element 5: Detail data ID 2	*Hexadecimal value*, ***NONE**	
	Element 6: Detail data 2	*Character value*, ***NODATA**	
	Element 7: Detail data ID 3	*Hexadecimal value*, ***NONE**	
	Element 8: Detail data 3	*Character value*, ***NODATA**	
	Element 9: Product identifier	***NONE**, *SNDHDW, *SNDSFW, *RSCHDW	

Top

Message identifier (MSGID)

Specifies the message ID to which this alert description corresponds.

This is a required parameter.

name Specify the message identifier.

Top

Alert table (ALRTBL)

Specifies the alert table in which this alert description is created.

This is a required parameter.

Qualifier 1: Alert table

name Specify the name of the alert table to be used.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the alert table name. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the library where the alert table name is located. Only the library named is searched. The user must have *USE authority for the specified library.

Top

Alert type (ALRTYPE)

Specifies the code point for the alert type.

***NONE**
There is no alert type code point for this alert description.

X'00'-X'F0'
Specify the alert type code point.

Top

Alert description code point (ALRD)

Specifies the code point for the alert description.

***NONE**
There is no alert description code point for this alert description.

X'0000'-X'FFF0'
Specify the alert description code point to be used.

Top

Probable cause code point (PBLCAUSE)

Specifies a maximum of 99 code points for probable causes, which are listed in order of decreasing probability.

Single values

***NONE**
There are no probable cause code points for this alert description.

Other values (up to 99 repetitions)

X'0000'-X'FFF0'
Specify the probable cause code point.

Top

Cause (CAUSE)

Specifies user, install, or failure causes. A maximum of 99 causes can be specified.

Up to three detailed data qualifiers or one product identifier qualifier can be specified for a code point, depending on the code point. For a detailed data qualifier, specify the detailed data ID and the detailed data.

Single values

***NONE**
There are no cause code points for this alert description.

Other values (up to 99 repetitions)

Element 1: Cause type

***USER**
A user cause code point follows.

***INSTALL**
An install cause code point follows.

***FAILURE**
A failure cause code point follows.

Element 2: Cause code point

hexadecimal-value
Specify the cause code point.

Element 3: Detail data ID 1

***NONE**
There is no detailed-data-ID code point for this code point.

hexadecimal-value
Specify the detailed data identifier code point used to identify the data. Detailed data identifiers can be specified up to three times in each session.

Element 4: Detail data 1

***NODATA**
There is no detailed data for this code point.

character-value
Specify up to 40 characters of detailed data. A substitution variable from the corresponding message description can be specified, in which case the message data is substituted into the alert description when the alert is created.

Element 5: Detail data ID 2

***NONE**
There is no detailed-data-ID code point for this code point.

hexadecimal-value
Specify the detailed data identifier code point used to identify the data. Detailed data identifiers can be specified up to three times in each session.

Element 6: Detail data 2

***NODATA**
There is no detailed data for this code point.

character-value
Specify up to 40 characters of detailed data. A substitution variable from the corresponding message description can be specified, in which case the message data is substituted into the alert description when the alert is created.

Element 7: Detail data ID 3

***NONE**
There is no detailed-data-ID code point for this code point.

hexadecimal-value
Specify the detailed data identifier code point used to identify the data. Detailed data identifiers can be specified up to three times in each session.

Element 8: Detail data 3

***NODATA**
There is no detailed data for this code point.

character-value
Specify up to 40 characters of detailed data. A substitution variable from the corresponding message description can be specified, in which case the message data is substituted into the alert description when the alert is created.

Element 9: Product identifier

***NONE**
There is no product identifier for this code point.

***SNDHDW**
Indicates the sender hardware (always System i5).

***SNDSFW**
Indicates the sender software code responsible for the alert. This is from the LICPGM value from the Create Alert Table (CRTALRTBL) command.

***RSCHDW**
Indicates the failing resource hardware, which is determined by the resource hierarchy in the message description.

Recommended action (ACTION)

Specifies a recommended action for a user, install, or failure cause. A maximum of 99 actions can be listed.

Single values

***NONE**
There are no recommended action code points for this alert description.

Other values (up to 99 repetitions)

Element 1: Action type

***USER**
A user cause code point follows.

***INSTALL**
An install cause recommended action code point follows.

***FAILURE**
A failure cause recommended action code point follows.

***UNKNOWN**
A code point for a recommended action for a 'cause undetermined' error follows.

Element 2: Action code point

hexadecimal-value
Specify the recommended action code point.

Up to three detailed data qualifiers or one product identifier qualifier can be specified for a code point, depending on the code point. For a detailed data qualifier, specify the detailed data ID and the detailed data.

Element 3: Detail data ID 1

***NONE**
There is no detailed-data-ID code point for this code point.

hexadecimal-value
Specify the detailed data identifier code point used to identify the data. Detailed data identifiers can be specified up to three times in each session.

Element 4: Detail data 1

***NODATA**
There is no detailed data for this code point.

character-value
Specify up to 40 characters of detailed data. A substitution variable from the corresponding message description can be specified, in which case the message data is substituted into the alert description when the alert is created.

Element 5: Detail data ID 2

***NONE**
There is no detailed-data-ID code point for this code point.

hexadecimal-value
Specify the detailed data identifier code point used to identify the data. Detailed data identifiers can be specified up to three times in each session.

Element 6: Detail data 2

***NODATA**
There is no detailed data for this code point.

character-value
Specify up to 40 characters of detailed data. A substitution variable from the corresponding message description can be specified, in which case the message data is substituted into the alert description when the alert is created.

Element 7: Detail data ID 3

***NONE**
There is no detailed-data-ID code point for this code point.

hexadecimal-value
Specify the detailed data identifier code point used to identify the data. Detailed data identifiers can be specified up to three times in each session.

Element 8: Detail data 3

***NODATA**
There is no detailed data for this code point.

character-value
Specify up to 40 characters of detailed data. A substitution variable from the corresponding message description can be specified, in which case the message data is substituted into the alert description when the alert is created.

Element 9: Product identifier

***NONE**
There is no product identifier for this code point.

***SNDHDW**
Indicates the sender hardware (always System i5).

***SNDSFW**
Indicates the sender software code responsible for the alert. This is from the LICPGM value from the Create Alert Table (CRTALRTBL) command.

***RSCHDW**
Indicates the failing resource hardware, which is determined by the resource hierarchy in the message description.

Top

Examples

```
ADDALRD   MSGID(USR1234)  ALRTBL(USER/USRMSGS)
          ALRTYPE(01)  ALRD(3100)  PBLCAUSE(1000 3121)
          CAUSE((*USER 6001)(*FAILURE 1000)(*FAILURE 3121))
          ACTION((*USER 1000)
                 (*FAILURE 00B0 00A5 'DSPMSG QSYSOPR')
                 (*FAILURE F0A0 22 '&5')
                 (*FAILURE 00E1 *NONE *NODATA *NONE
                  *NODATA *NONE *NODATA *SNDHDW))
```

This command defines three recommended failure actions:

- '00B0', which requires a detailed qualifier. One detailed data qualifier is provided.
- The detailed data identifier code point is '00A5', which identifies the text Command and
- The detailed data 'DSPMSG QSYSOPR'.

Failure recommended action 'F0A0' specifies a message substitution variable ('&5') as the detailed data. When the message 'USR1234' is sent, the message data for variable '&5' is put into the alert for the detailed data.

Failure action X'00E1' references a product identifier; in this case, it is the sending hardware System i5. Place holders are needed for the detailed data qualifiers.

Top

Error messages

***ESCAPE Messages**

CPF1A01
Alert table &1 in &2 cannot be extended.

CPF1A02
Alert code &1 already in alert table &2.

CPF1A03
Alert identifier &1 already in alert table &2.

CPF1A05
Alert table &1 in &2 damaged.

CPF2499
Message identifier &1 not valid.

CPF7BB0
Alert description already exists.

CPF7BB5
Alert description &1 could not be added to alert table &2 in library &3.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9811
Program &1 in library &2 not found.

CPF9812
File &1 in library &2 not found.

CPF9814
Device &1 not found.

CPF9820
Not authorized to use library &1.

CPF9821
Not authorized to program &1 in library &2.

CPF9822
Not authorized to file &1 in library &2.

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9831
Cannot assign device &1.

CPF9899
Error occurred during processing of command.

Top

Add Alert Selection Entry (ADDALRSLTE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Alert Selection Entry (ADDALRSLTE) command allows the user to add an alert selection entry to an alert filter. Selection entries are the criteria that categorize a group of alerts. More information on alerts is in the Alerts Support book, SC41-5413.

Top

Parameters

Keyword	Description	Choices	Notes
FILTER	Filter	*Qualified object name*	Required, Positional 1
	Qualifier 1: Filter	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SELECT	Selection data	Single values: *ANY Other values (up to 10 repetitions): *Element list*	Required, Positional 2
	Element 1: Relationship	*IF, *AND, *OR	
	Element 2: Attribute	*ORIGIN, *RSCNAME, *RSCTYPE, *HIERNAME, *HIERTYPE, *MSGID, *MSGSEV, *ALRID, *ALRTYPE, *ALRDSC, *PBLCSE, *USRCSE, *INSCSE, *FLRCSE, *RSCHDW, *SNDHDW, *RSCSFW, *SNDSFW	
	Element 3: Relational operator	*EQ, *GT, *LT, *NE, *GE, *LE, *CT	
	Element 4: Value	*Character value*	
SEQNBR	Sequence number	1-9999, ***GEN**	Optional
GROUP	Group	*Name*, ***DEFAULT**	Optional

Top

Filter (FILTER)

Specifies the filter to be added.

This is a required parameter.

Qualifier 1: Filter

name Specify the name of the filter.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the filter. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the filter is located.

Selection data (SELECT)

Specifies the comparisons to be made to determine if the alert belongs in the specified group. The selection entry results in a successful match with an alert when the data in the alert satisfies the relationships specified on the SELECT parameter. Up to 10 attribute values can be compared to the alert.

This is a required parameter.

Single values

***ANY** Any alert matches this selection record.

Other values (up to 10 repetitions)

Specify the conditions under which an alert matches the selection entry. Each condition must contain the following four elements:

1. One of the logical operators *IF, *AND, or *OR
2. The attribute compared
3. One of the relational operators
4. The attribute value

Element 1: Relationship

***IF** Identifies the first condition that must be satisfied.

***AND** The conditions on both sides of the *AND must be satisfied.

***OR** One of the conditions on each side of the *OR must be satisfied.

If there is one set or several sets of conditions, the *IF value must be specified as the first value in the first set of comparison values. If more than one set of conditions are specified, *AND or *OR must be specified as the first value in each set after the first. Each condition must be enclosed in parentheses. *AND is evaluated before *OR.

Element 2: Attribute

***ORIGIN**
Specifies whether the alert is generated or received. The valid values for this attribute are L (Locally generated) or R (Received).

***RSCNAME**
Specifies the name of the failing resource. The value for this attribute must be a 8-character name.

***RSCTYPE**
Specifies the type of the failing resource. The value for this attribute must be a 3-character resource type (for example, TAP or DKT).

***HIERNAME**
Specifies all of the resources in the alert resource hierarchy. The alert resource hierarchy is the list of resources, separated by blanks, displayed on the Work with Alerts (WRKALR) command detailed data displays. The value for this attribute can be a list of up to 5 resource names separated by a blank, unless the value is used with the *CT relational operator. If the *CT value is used, the selection relationship can test to see if the given resource name is found anywhere within the hierarchy. This attribute contains the resource names from the hierarchy only.

***HIERTYPE**
Specifies all of the resource types in the alert resource hierarchy. The resource types match the

resource names specified on the *HIERNAME attribute. The value for this attribute can be a list of up to 5 resource types 1 to 3 characters in length separated by a blank, unless the value is used with the *CT relational operator. If the *CT value is used, the selection relationship can test to see if the given resource type is found anywhere within the hierarchy.

***MSGID**
Specifies the message identifier.

***MSGSEV**
Specifies the message severity. This value, 00 through 99, represents the severity level of the message (99 is the highest severity level).

***ALRID**
Specifies the alert identifier. The alert identifier is displayed on the Work with Alerts (WRKALR) command detailed data display. The value for this attribute must be an 8-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 8. The alert ID may not be a valid comparison for alerts created after problem analysis.

***ALRTYPE**
Specifies the alert type code point that is in the alert. The value for this attribute is a 2 digit hexadecimal number.

***ALRDSC**
Specifies the alert description code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***PBLCSE**
Specifies the probable cause code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***USRCSE**
Specifies the first user cause code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***INSCSE**
Specifies the first install cause code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***FLRCSE**
Specifies the first failure cause code point that is in the alert. The value for this attribute must be an 4-digit hexadecimal number unless it is used with the *CT relational operator. If the *CT operator or a wildcard character is used, the attribute must have an even number of digits up to a maximum of 4.

***RSCHDW**
Specifies the failing hardware resource information in the alert. This information is displayed on the Work with Alerts (WRKALR) command detailed data displays. Specify a value for this attribute using the following form:

```
'tttt mmm ss-ssssssss'
'tttt mmm ss-sssss'
'tttt mmm sssssss'
'tttt mmm sssss'
```

where *tttt* is the machine type, *mmm* is the model number, and *sssssssss* is the serial number. Use this format to match a particular hardware resource or use a part of the hardware value with the *CT relational operator to provide a partial match.

***SNDHDW**
Specifies the sending hardware resource information in the alert. This information is displayed on the Work with Alerts (WRKALR) command detailed data displays. Specify a value for this attribute using the following form:

```
'tttt mmm ss-ssssssss'
'tttt mmm ss-sssss'
'tttt mmm ssssssss'
'tttt mmm sssss'
```

where *tttt* is the machine type, *mmm* is the model number, and *sssssssss* is the serial number. Use this format to match a particular hardware resource or use a part of the hardware value with the *CT relational operator to provide a partial match.

***RSCSFW**
Specifies the failing software resource information in the alert. This information is displayed on the Work with Alerts (WRKALR) command detailed data displays. Specify a value for this attribute using the following form:

```
'ppppppp vv rr mm'
```

where *ppppppp* is the licensed program identifier, *vv* is the version number, *rr* is the release number, and *mm* is the modification level. Use this format to match a particular software resource or use a part of the software value with the *CT relational operator to provide a partial match.

***SNDSFW**
Specifies the sending software resource information in the alert. This information is displayed on the Work with Alerts (WRKALR) command detailed data displays. Specify a value for this attribute using the following form:

```
'ppppppp vv rr mm'
```

where *ppppppp* is the licensed program identifier, *vv* is the version number, *rr* is the release number, and *mm* is the modification level. Use this format to match a particular software resource or use a part of the software value with the *CT relational operator to provide a partial match.

Element 3: Relational operator

***EQ** The attribute in **Attribute** must be equal to the value specified in **Attribute Value**.

***GT** The attribute in **Attribute** must be greater than the value specified in **Attribute Value**.

***LT** The attribute in **Attribute** must be less than the value specified in **Attribute Value**.

***NE** The attribute in **Attribute** must not be equal to the value specified in **Attribute Value**.

***GE** The attribute in **Attribute** must be greater than or equal to the value specified in **Attribute Value**.

***LE** The attribute in **Attribute** must be less than or equal to the value specified in **Attribute Value**.

***CT** The attribute in **Attribute** must contain the value specified in **Attribute Value**.

Element 4: Value

attribute-value
Specify the value (a maximum of 60 characters) to be compared with the contents of the specified attribute. The value must be specified in apostrophes if it contains blanks or special characters and must be in character format. If a CL variable is specified for the value, it must be a character variable.

generic*-attribute-value
Specify the generic attribute value. A generic value is a character string of one or more characters followed by an asterisk (*); for example, ABC*. If a generic name is specified, all values that begin with the generic value are matches. If an asterisk is not included with the generic (prefix) value, the system assumes it to be the complete value. Generic attribute values are only allowed with the *EQ and *NE operator.

Top

Sequence number (SEQNBR)

Specifies the sequence number of the alert selection entry. Selection entries are evaluated in order by sequence number.

***GEN** Allows the system to generate the sequence number. The sequence number will be greater than all previous selection entries.

1-9999 Specify the sequence number to be used.

Top

Group (GROUP)

Specifies the group that an alert is assigned to if the alert matches the criteria specified on the SELECT parameter.

***DEFAULT**
The alert is assigned to the *DEFAULT group. The *DEFAULT group is added automatically when a filter is created.

name Specify a group name to which the alert is assigned.

Top

Examples

```
ADDALRSLTE   FILTER(MYLIB/MYFILTER)
             SELECT((*IF *RSCNAME *EQ CHICAGO1)
                    (*AND *RSCTYPE *EQ CP))
             SEQNBR(*GEN)  GROUP(CHICAGO)
```

This command adds selection entry 0010 to the filter MYFILTER in library MYLIB (a 0010 is generated because no entries have been added to the filter). Any alerts that have a resource name of 'CHICAGO1' and a resource type of 'CP' (control point) are assigned to group CHICAGO.

Top

Error messages

***ESCAPE Messages**

CPD91CB
*CT not allowed with numeric values.

CPF2150
Object information function failed.

CPF2151
Operation failed for &2 in &1 type *&3.

CPF812F
Filter damaged.

CPF91DA
Sequence number &4 already exists.

CPF91DE
Filter &1/&2 at maximum size.

CPF91D9
Sequence number cannot be automatically created.

CPF91EA
*IF relationship not in correct position.

CPF91EB
Filter type &3 not correct for this operation.

CPF91EC
Internal processing error occurred.

CPF91E6
Generic values only allowed with *EQ or *NE.

CPF91E7
Character in position &4 not valid in value specified.

CPF91E8
Internal processing error occurred.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9830
Cannot assign library &1.

Top

Add Authorization List Entry (ADDAUTLE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Authorization List Entry (ADDAUTLE) command allows the user to add entries to an authorization list. An entry consists of a user's name and the authorities associated with that user on the authorization list. Both the authorization list and the user profile must exist. If the specified user is already on the list, a message is issued and the user's authorities on the list are not changed.

The users who can use this command to add users to an authorization list are: the owner of the authorization list, a user with authorization list management (*AUTLMGT) authority on the authorization list, or a user with all object (*ALLOBJ) special authority.

When the ADDAUTLE command is used to add a user to an authorization list, the user must specify the name of the authorization list, a list of authorized users, and a list of authorities specified for the list. Each user on the list is given the authorities specified on the command.

Restrictions:

- Authorization list management (*AUTLMGT) authority allows a user to manage the authorization list and, therefore, to manage the authorities for all objects secured by the list.
- Only the owner of the list or a user with all object (*ALLOBJ) special authority can add a user with authorization list management (*AUTLMGT) authority.
- A user with *AUTLMGT authority can add users and give specific authorities only to the *AUTLMGT level.

Top

Parameters

Keyword	Description	Choices	Notes
AUTL	Authorization list	*Generic name, name*	Required, Positional 1
USER	User	Values (up to 50 repetitions): *Name*	Required, Positional 2
AUT	Authority	Single values: *EXCLUDE Other values (up to 11 repetitions): ***CHANGE**, *ALL, *USE, *OBJALTER, *OBJEXIST, *OBJMGT, *OBJOPR, *OBJREF, *ADD, *DLT, *EXECUTE, *READ, *UPD, *AUTLMGT	Optional, Positional 3

Top

Authorization list (AUTL)

Specifies the authorization list to which the users are to be added. The authorization list must already exist.

This is a required parameter.

generic-name
Specify the generic name of the authorization lists to be changed.

A generic name is a character string of one or more characters followed by an asterisk (*); for example ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name.

name Specify the name of the authorization list to which the user profile name is added.

Top

User (USER)

Specifies one or more users to be added to the authorization list. Up to 50 user profile names can be specified. If a user profile name is already on the authorization list, a message is issued and the user's authorities are not changed.

This is a required parameter.

name Specify the name of the user profile to be added to the authorization list.

Top

Authority (AUT)

Specifies the authority to be given to the users specified for the **User (USER)** parameter. Users must have *AUTLMGT authority to manage the authorization list.

Single values

***EXCLUDE**
The user cannot access the object.

Other values (up to 11 repetitions)

***CHANGE**
The user can perform all operations on the object except those limited to the owner or controlled by object existence (*OBJEXIST) and object management (*OBJMGT) authorities. The user can change and perform basic functions on the object. *CHANGE authority provides object operational (*OBJOPR) authority and all data authority. If the object is an authorization list, the user cannot add, change, or remove users.

***ALL** The user can perform all operations except those limited to the owner or controlled by authorization list management (*AUTLMGT) authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the object.

***USE** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. Use (*USE) authority provides object operational (*OBJOPR), read (*READ), and execute (*EXECUTE) authorities.

***AUTLMGT**
Authorization list management authority provides the authority to add users to the authorization list, to change users' authorities on the authorization list, to remove user names from the authorization list, or to remove users from the authorization list, to rename an authorization list, or to create a duplicate authorization list.

***OBJALTER**
Object alter authority provides the authority needed to alter the attributes of an object. If the user has this authority on a database file, the user can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. If the user has this authority on an SQL package, the user can change the attributes of the SQL package. This authority is currently only used for database files and SQL packages.

***OBJEXIST**
Object existence authority provides the authority to control the object's existence and ownership. These authorities are necessary for users who want to delete an object, free storage for an object, perform save and restore operations for an object, or transfer ownership of an object. A user with special save system (*SAVSYS) authority does not need existence authority to save or restore objects. Object existence authority is required to create an object that has an existing authority holder.

***OBJMGT**
Object management authority provides the authority to The security for the object, move or rename the object, and add members to database files.

***OBJOPR**
Object operational authority provides authority to look at the description of an object and to use the object as determined by the user's data authority to the object.

***OBJREF**
Object reference authority provides the authority needed to reference an object from another object such that operations on that object may be restricted by the other object. If the user has this authority on a physical file, the user can add referential constraints in which the physical file is the parent. This authority is currently only used for database files.

Data authorities

***ADD** Add authority provides the authority to add entries to an object (for example, job entries to an queue or records to a file).

***DLT** Delete authority allows the user to remove entries from an object (for example, remove messages from a message queue or records from a file.)

***EXECUTE**
Execute authority provides the authority needed to run a program or locate an object in a library or directory.

***READ**
Read authority provides the authority needed to show the contents of an object.

***UPD** Update authority provides the authority to change the entries in an object.

Top

Examples

```
ADDAUTLE  AUTL(PAYROLL)  USER(TOM)  AUT(*ALL *AUTLMGT)
```

This command adds user TOM to the PAYROLL authorization list and gives him all authority to the objects secured by the authorization list. TOM also has authority to manage the authorization list.

Top

Error messages

***ESCAPE Messages**

CPF22AA
Only *AUTLMGT authority can be specified with *ALL authority.

CPF22AB
Only *AUTLMGT can be specified with *CHANGE authority.

CPF22AC
Only *AUTLMGT authority can be specified with *USE authority.

CPF2253
No objects found for &1 in library &2.

CPF2280
*PUBLIC is always on authorization list, cannot be added.

CPF2281
The users specified do not exist on the system.

CPF2282
&1 errors adding users, &2 authorization lists processed.

CPF2283
Authorization list &1 does not exist.

CPF2284
Not authorized to change authorization list &1.

CPF2289
Unable to allocate authorization list &1.

CPF2290
*EXCLUDE cannot be specified with another authority.

Top

Add Breakpoint (ADDBKP)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Breakpoint (ADDBKP) command sets up to ten breakpoints in a program. A breakpoint is a location in a program where processing stops and control is given to the user or to a specified program. The breakpoint is set when a statement number or label of a command or machine instruction is specified. The program is stopped just before processing begins on the statement (or machine instruction) on which the breakpoint is set.

This command shows the values of certain program variables when any breakpoint in the program is reached. As many as 10 variables per breakpoint can be specified, and as many as 10 breakpoints per command can be set. However, the same program variables apply to every breakpoint specified in the command. To specify different sets of variables for each breakpoint, you must use separate commands.

This command specifies conditional breakpoints in which the program is stopped when a condition is true. This condition involves two program variables or one program variable and a constant. When using conditional breakpoints, it is possible to stop the program when a program variable becomes a certain value.

A conditional breakpoint can also be specified by specifying a skip value. The program does not stop until the breakpoint statements have been processed as many times as the skip number indicates. After that, the breakpoint causes the program to stop.

When a breakpoint is reached in the interactive debugging environment, a display is shown to the user that identifies which breakpoint has been reached and (optionally) shows the values of the specified program variables when the program is stopped. This information is also written to the job log. From the display, the user can press the F10 key to show the command entry display, or press the F3 key to exit the display and cancel the program. The user can press the Enter key to allow the program to continue running.

When a breakpoint is reached in the batch debugging environment, the breakpoint information is written to a printer file and, optionally, another program can be called to take action on the breakpoint condition. The name of the called program is specified on the **Breakpoint program to call (BKPPGM)** parameter.

When an interactive job is debugging another job and a breakpoint is reached in the debugged job, a breakpoint display is shown. This display appears in the debugging job, interrupting what was previously being displayed. You must press the Enter key, allowing the stopped program to continue, before returning to the previous display.

Restrictions:

- You can use this command only in debug mode. To start debug mode, refer to the Start Debug (STRDBG) command.
- You cannot use this command if you are servicing another job, and that job is on a job queue, or is being held, suspended, or ended.
- This command cannot be used to add breakpoints to a bound program.
- To use this command, you must have all object (*ALLOBJ) special authority.

Top

Parameters

Keyword	Description	Choices	Notes
STMT	Statement identifier	Values (up to 10 repetitions): *Character value*	Required, Positional 1
PGMVAR	Program variables	Single values: ***NONE** Other values (up to 10 repetitions): *Element list*	Optional, Positional 2
	Element 1: Program variable	*Character value*, *CHAR	
	Element 2: Basing pointer variable	Values (up to 5 repetitions): *Character value*	
OUTFMT	Output format	***CHAR**, *HEX	Optional, Positional 5
PGM	Program	*Name*, ***DFTPGM**	Optional
START	Char output start position	*Integer*, **1**	Optional, Positional 3
LEN	Characters to display	*Integer*, ***DCL**	Optional, Positional 4
SKIP	Skip value	*Integer*, **0**	Optional
BKPCOND	Breakpoint condition	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Variable	*PGMVAR1, *PGMVAR2, *PGMVAR3, *PGMVAR4, *PGMVAR5, *PGMVAR6, *PGMVAR7, *PGMVAR8, *PGMVAR9, *PGMVAR10	
	Element 2: Operator	*EQ, *GT, *LT, *NE, *GE, *NL, *LE, *NG, *CT	
	Element 3: Compare value or variable	*Character value*	
BKPPGM	Breakpoint program to call	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Breakpoint program to call	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

Statement identifier (STMT)

Specifies the statement identifiers of one to ten statements or machine instructions in the program at which breakpoints are set. When run, the program breaks **before** processing a statement specified as a breakpoint.

This is a required parameter.

The list can contain a maximum of 10 identifiers (statement numbers, program labels, or machine instruction numbers) that are valid for the program specified by the PGM parameter. At least one identifier is needed. If a machine instruction number is specified, a slash must be placed in front of the number and both the slash and the number must be enclosed in apostrophes.

In high-level language programs, different statements, different labels, or both can be mapped to the same internal instruction. This happens when there are several statements that do not operate on variables directly (such as DO, END, and comments) following one another in a program. To determine which statements can be mapped to the same instruction, the intermediate representation of a program listing can be used.

Because different statements can be mapped to the same instruction, adding a breakpoint can redefine a previous breakpoint that was added for a different statement. When this occurs, the new breakpoint replaces the previously added breakpoint.

Top

Program variables (PGMVAR)

Specifies the names of up to ten program variables shown that are in a high-level language or machine instruction program. The name and the value of each program variable is shown when any of the breakpoints specified on the **Statement identifier (STMT)** parameter are reached. When run, the program stops **before** processing a statement specified as a breakpoint.

Note: In some high-level languages such as RPG, variables that are declared but not referred to in the program cannot be specified on the PGMVAR parameter.

Single values

***NONE**
No program variables are shown for any of the breakpoints specified.

Other values (up to 10 repetitions)

Element 1: Program variable

***CHAR**
This special value is specified instead of a variable name if a basing pointer is also specified. This special value displays a character view of a pointer to be shown without the use of a based variable.

character-value
Specify the names of one to ten program variables, separated by blanks, shown when a breakpoint is reached. The name must be enclosed in apostrophes if it contains special characters.

If the program variable is an array, the subscripts representing the element in the array can be specified. If an array name is specified without any subscripts, all of the array elements are recorded. A single-dimensional cross-section can also be specified. Up to 132 characters may be specified for this program variable entry. This includes any qualifiers, subscripts, embedded blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, a machine-interface object-definition-table-vector (MI ODV) number, asterisk (single-dimensional cross-section), or a numeric variable name can be specified for a subscript. For more information on testing and debugging at machine interface level and on the program variable value, refer to the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Element 2: Basing pointer variable

character-value
Specify up to five basing pointers for each program variable to be shown. In some languages, the program variable may be based on a pointer variable. This set of values allows you to explicitly specify the basing pointers for the variable being recorded. Each basing pointer name must be enclosed in apostrophes if it contains special characters.

If the basing pointer is an array, the subscripts representing an element in the array must be specified. Up to 132 characters can be specified for a basing pointer name. This includes any qualification, subscripts, embedded blanks, parentheses, and commas. It does not include the

enclosing apostrophes when special characters are used. An integer, a machine-interface object-definition-table-vector (MI ODV) number, or a numeric variable name can be specified for a subscript.

For more information, refer to ″Parameter values used for testing and debugging″ in ″CL concepts and reference″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Top

Output format (OUTFMT)

Specifies the format used to show the variables.

***CHAR**
Variables are shown in character form.

***HEX** Variables are shown in hexadecimal form.

Top

Program (PGM)

Specifies the program to which the breakpoints are added.

***DFTPGM**
The breakpoints are added to the program currently specified as the default program in debug mode.

name Specify the name of the program to which the breakpoints are added. The program must already be in debug mode.

Top

Char output start position (START)

Specifies, for string variables only, the starting position in the string from which its value is shown when the breakpoint is reached. If more than one string variable is specified for the **Program variables (PGMVAR)** parameter, the same starting position value is used for each one. For a bit string, the value specifies the starting bit position, and for a character string, the value specifies the starting character position.

For conditional breakpoints, the **Char output start position (START)** parameter also specifies the starting point in the string where the comparison is made.

1 The variable is shown from the first position onward through the length specified by the value supplied for the **Characters to display (LEN)** parameter.

integer
Specify the first position of the program variable to be shown.

The value supplied for the START parameter must not be larger than the maximum string length for any variable specified, except that a value of **1** for the START parameter is allowed if the maximum length for a string is zero. The value supplied for the LEN parameter plus the value supplied for the START parameter minus one, must not be greater than the maximum string length. These checks are made for each string variable specified for the PGMVAR parameter.

Top

Characters to display (LEN)

Specifies, for string variables only, the length of the string shown when the breakpoint is reached, starting at the position specified by the value supplied for the **Char output start position (START)** parameter. If more than one string variable is specified by the value supplied for the **Program variables (PGMVAR)** parameter, the same value is used for each one. For a bit string, the value specifies the number of bits shown, and for a character string, the value specifies the number of characters shown.

For conditional breakpoints, the **Characters to display (LEN)** parameter also specifies the length of the string where the comparison is made.

***DCL** The string variable is shown to the end of the string or for a value of 200 bytes, whichever is less. If the string variable has a maximum length of zero, the only allowable value for the LEN parameter is *DCL.

integer
Specify the length of the data shown. The length, as well as the combination of values supplied for the START parameter and the LEN parameter must be no greater than the length of the shortest string specified by the value supplied for the PGMVAR parameter.

Top

Skip value (SKIP)

Specifies the number of times the statement or statements on the **Statement identifier (STMT)** parameter must be processed before the program is stopped.

0 The program stops immediately when the statement or statements are processed. No skipping of breakpoints is done.

integer
Specify the number of times the statements must be processed before the program is stopped. If more than one statement is specified, each statement will have its own independent skip value. There is a separate skip count for each statement.

Top

Breakpoint condition (BKPCOND)

Specifies a condition, defined by an expression, that must be true before the program is stopped. The expression is tested before any statement on the **Statement identifier (STMT)** parameter is processed. If the expression is false, the breakpoint does not stop the program. If the expression is true, the program is stopped.

***NONE**
No breakpoint expression is specified.

breakpoint-expression
Specify a breakpoint expression indicating a conditional breakpoint. A breakpoint expression consists of these three parts:

- A variable (*PGMVAR1 to *PGMVAR10)
- An operator
- A compare value

variable
Specify the variable to be used in the breakpoint expression. For example, *PGMVAR1 indicates the first variable of the **Program variables (PGMVAR)** parameter, *PGMVAR2 indicates the second, and so on. Only numeric, character, or bit variables may be specified.

operator

Specify what type of comparison is to be done for a conditional breakpoint. The following comparisons are allowed:

***EQ** equal to

***NE** not equal to

***GT** greater than

***LT** less than

***GE** greater than or equal to

***NL** not less than (same as *GE)

***LE** less than or equal to

***NG** not greater than (same as *LE)

***CT** contains

Note: The *CT operator compares whether one character string contains one or more occurrences of another character string. This comparison is for an exact match, and it is case sensitive.

compare-value

Specify a constant or another variable to compare with the first variable. If a constant is specified, it must be the same type as the variable. If the variable is numeric, the constant must be a number. If the variable is a bit, the constant must be a string containing only '1's and '0's. If the variable is a character, the compare value is treated as a character string, even if a number is specified.

If another program variable is specified, it is compared with the first variable. The variables must be of the same type. If the variables are numeric, they must both be floating point or not floating point. For example, a packed number may not be compared with a floating point number.

When comparing two non-floating point variables, or a non-floating point variable and a constant, the total number of digits needed to represent them must not exceed 31. For example, a PACKED(24,2) and a PACKED(24,20) cannot be compared. The first variable requires 22 digits to the left of the decimal point and two digits to the right. The second variable requires four digits to the left of the decimal and 20 digits to the right. To compare them requires a variable with 22 digits to the left of the decimal and 20 to the right. This exceeds the maximum number of allowed digits, 31.

When comparing two character strings, the shorter of the two will be padded on the right with blanks. When comparing two bit strings, they must both be of the same length. The **Skip value (SKIP)** parameter and the **Breakpoint condition (BKPCOND)** parameter may be used together. In this case, the breakpoint expression is not evaluated until the breakpoint has skipped the number of times specified by SKIP. After that, the breakpoint expression is evaluated and the program stops if the expression is true.

Top

Breakpoint program to call (BKPPGM)

Specifies the name of the user-supplied program (if any) to call when a breakpoint is reached in the program specified by the **Program (PGM)** parameter. When the program specified by the BKPPGM parameter is called, it is passed four parameters that identify the program, the recursion level, the HLL statement identifier, and machine instruction number at which the breakpoint occurred. The parameters have the following format:

1. Program name (10 bytes). The name of the program in which the breakpoint was reached.

2. Recursion level (5 bytes). The recursion level number of the program in which the breakpoint was reached. This value is a 1-digit to 5-digit number that is padded on the right with blanks.
3. Statement identifier (10 bytes). The high-level language program statement identifier that was reached. This statement identifier is the statement identifier specified in the Add Breakpoint (ADDBKP) command that defined the breakpoint. If a machine instruction number was used to specify the breakpoint, this parameter contains a slash (/) followed by a 4-digit hexadecimal machine instruction number.
4. Instruction number (5 bytes). The machine instruction number that corresponds to the high-level language statement at which the breakpoint was reached. No slash appears in front of this machine instruction number. It consists of 1 to 4 hexadecimal characters that represent the MI instruction number, followed by one or more blanks. If a machine instruction number is passed in the third parameter, the numbers in the third and fourth parameters are the same.

All the parameter values are left-adjusted and padded with blanks. When the called program returns, the program being debugged continues processing, starting with the statement that has the breakpoint on it.

Single values

***NONE**
No breakpoint-handling program is called when any breakpoint specified in this command is reached in the batch environment. The stopped program continues processing.

Qualifier 1: Breakpoint program to call

name Specify the name of the user-supplied program to be called if any of the breakpoints on this command are reached while debugging in a batch environment. The program specified here should not be the same as the program specified on the **Program (PGM)** parameter. If they are the same, the results are unpredictable. After the called program runs, it returns control to the stopped program, which continues processing.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the program. If no library is specified as the current library for the job, QGPL is used.

name Specify the library where the program is located.

Top

Examples

Example 1: Adding Breakpoints in Debug Mode

```
ADDBKP  STMT(150 RTN1 205) PGMVAR('&TEMP' '&INREC')
```

This command establishes breakpoints at CL statement numbers 150 and 205 and at the label RTN1 for the default program in debug mode. When any of these breakpoints is reached, the CL variables &TEMP and &INREC are automatically shown. Note that the CL variables must include a leading ampersand (&) and be specified within apostrophes.

Example 2: Adding Breakpoints to HLL Program

```
ADDBKP  STMT(100)  PGMVAR('AMOUNT(200)') PGM(MYPROG)
```

Assume in this example that MYPROG is a high-level language program being debugged in an interactive environment and that the program variable AMOUNT is a 250-element array in MYPROG. This command adds a breakpoint to statement 100 in MYPROG. When MYPROG is started, the program stops processing at statement 100, and the value of the 200th element of the AMOUNT array is shown. If AMOUNT had been specified without a subscript, all of the array elements would have been shown.

Example 3: Program Stops After Processing Statement 10 Times

```
ADDBKP   STMT(10)  SKIP(1000)
```

This command causes the default program to stop when statement 10 is processed 1000 times (the breakpoint is skipped 1000 times).

Example 4: Program Stops After Processing Multiple Statements

```
ADDBKP   STMT(10 20 30)  SKIP(50)
```

This command causes the default program to stop when statements 10, 20, and 30 are processed 50 times.

Example 5: Conditional Breakpoint

```
ADDBKP   STMT(10)  PGMVAR(X)  BKPCOND(PGMVAR1 *EQ 5)
```

This command stops the default program at statement 10 when variable X is equal to five.

Example 6: Conditional Breakpoint

```
ADDBKP   STMT(20)  PGMVAR((S1) (S2))  SKIP(100)
         BKPCOND(*PGMVAR1 *CT *PGMVAR2)
```

This command stops after statement 20 has been processed 100 times, and then only if the character string S2 occurs in the character string S1.

Top

Error messages

***ESCAPE Messages**

CPF1999

Errors occurred on command.

Top

Add Binding Directory Entry (ADDBNDDIRE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Binding Directory Entry (ADDBNDDIRE) command adds an entry to a binding directory.

Restrictions:

- You must have use (*USE) authority for the library where the binding directory to be changed is located.
- You must have object operational (*OBJOPR) and add (*ADD) authorities to the binding directory to be changed.
- You must have execute (*EXECUTE) authority to the specified library when a generic name is specified for the **Object specifications (OBJ)** parameter.

Top

Parameters

Keyword	Description	Choices	Notes
BNDDIR	Binding directory	*Qualified object name*	Required, Positional 1
	Qualifier 1: Binding directory	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB, *USRLIBL	
OBJ	Object specifications	Values (up to 50 repetitions): *Element list*	Required, Positional 2
	Element 1: Object	*Qualified object name*	
	Qualifier 1: Object	*Generic name, name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**	
	Element 2: Object type	***SRVPGM**, *MODULE	
	Element 3: Activation	***IMMED**, *DEFER	
POSITION	Position specifications	Single values: ***LAST**, *FIRST Other values: *Element list*	Optional, Positional 3
	Element 1: Object position	*AFTER, *BEFORE, *REPLACE	
	Element 2: Object	*Qualified object name*	
	Qualifier 1: Object	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**	
	Element 3: Object type	***SRVPGM**, *MODULE	

Top

Binding directory (BNDDIR)

Specifies the binding directory to which an entry is added.

This is a required parameter.

Qualifier 1: Binding directory

name Specify the name of the binding directory to be updated.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

***USRLIBL**
Only the libraries in the user portion of the job's library list are searched.

name Specify the name of the library to be searched.

Top

Object specifications (OBJ)

Specifies one or more object names to be added to the binding directory.

This is a required parameter.

You can specify 50 values for this parameter.

You can control the activation of each service program. You can specify whether the referenced service program is activated at the same time as the program or service program being created from the binding directory, or is deferred until a procedure exported from the referenced service program is called. Deferring activation may improve your application's performance.

Element 1: Object

Qualifier 1: Object

***ALL** All objects of the specified object type residing in the specified library are to be added.

generic-name
Specify the generic name of the object. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. If the complete object name is specified, and multiple libraries are searched, multiple objects can be added only if *ALL or *ALLUSR library values can be specified for the name.

name Specify the name of the object to be added to the binding directory.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

name Specify the name of the library to be searched.

Element 2: Object type

***SRVPGM**
The object to be added is a service program.

***MODULE**
The object to be added is a module.

Element 3: Activation

***IMMED**
Activation of the bound service program takes place immediately when the program or service program being created from the binding directory is activated.

***DEFER**
Activation of the bound service program may be deferred until a function it exports is called.

Top

Position specifications (POSITION)

Specifies the position in the binding directory where the list of objects is added.

Single values

***LAST**
The list of objects is to be added to the end of the binding directory entries.

***FIRST**
The list of objects is to be inserted prior to the first binding directory entry.

Element 1: Object position

***AFTER**
The list of objects is added to the binding directory after the binding directory entry specified for this parameter. The entry specified must currently exist in the binding directory.

***BEFORE**
The list of objects is added to the binding directory before the binding directory entry specified for this parameter. The entry specified must currently exist in the binding directory.

***REPLACE**
The object specified on the OBJ parameter replaces the binding directory entry specified for this parameter. The entry specified must currently exist in the binding directory. Only one entry can be specified on the OBJ parameter.

Element 2: Object

Qualifier 1: Object

name Specify the object name of an existing binding directory entry.

Qualifier 2: Library

***LIBL** The existing binding directory entry has a library qualifier of *LIBL.

name Specify the name of the library qualifier for the existing entry.

Element 3: Object type

***SRVPGM**
The directory entry is a service program.

***MODULE**
The directory entry is a module.

Top

Examples

Example 1: Add Entry to End of List

```
ADDBNDDIRE   BNDDIR(TESTBNDDIR)  OBJ((TESTOBJ))
             POSITION(*LAST)
```

This command adds a binding directory entry for service program TESTOBJ in library *LIBL to the end of the list of binding directory entries found in the binding directory TESTBNDDIR.

Example 2: Add Entry to Beginning of List

```
ADDBNDDIRE   BNDDIR(TESTBNDDIR)
             OBJ((TESTLIB/TESTOBJ *MODULE))  POSITION(*FIRST)
```

This command adds a binding directory entry for module TESTOBJ in library TESLIB to the beginning of the binding directory entries found in the binding directory TESTBNDDIR.

Example 3: Add Multiple Entries

```
ADDBNDDIRE   BNDDIR(TESTBNDDIR)
             OBJ((TESTLIB/TESTOBJ *MODULE) (TESTOBJ2))
             POSITION(*FIRST)
```

This command adds a binding directory entry for module TESTOBJ in library TESTLIB followed by an entry for service program TESTOBJ2 in the library list to the beginning of the binding directory entries found in the binding directory TESTBNDDIR.

Example 4: Add Entries Before an Existing Entry

```
ADDBNDDIRE   BNDDIR(TESTBNDDIR)
             OBJ((TESTLIB/TESTOBJ *MODULE) (TESTOBJ2))
             POSITION(*BEFORE  *LIBL/TESTMOD *MODULE)
```

This command adds a binding directory entry for module TESTOBJ in library TESTLIB, followed by an entry for service program TESTOBJ2 in library *LIBL prior to the binding directory entry for module TESTMOD in library *LIBL found in the binding directory TESTBNDDIR.

The binding directory entry for module TESTMOD in library *LIBL must be found in the binding directory TESTBNDDIR for this operation to be successful.

Top

Error messages

***ESCAPE Messages**

CPF5D01
Binding directory &1 in library &2 is not usable.

CPF5D09
Object &2/&1 type &3 was not found in binding directory &4 in library &5.

CPF980F
Binding directory &1 in library &2 not found.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9830
Cannot assign library &1.

Top

Add Configuration List Entries (ADDCFGLE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Configuration List Entries (ADDCFGLE) command adds entries to a configuration list.

Note: You can also use an option on the full screen entry display of the Work With Configuration Lists (WRKCFGL) command to add, remove, or change entries in an existing list except for the configuration lists of type *APPNDIR, *APPNSSN, and *SNAPASTHR. TYPE(*SNAPASTHR).

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Configuration list type	*APPNDIR, *APPNLCL, *APPNRMT, *APPNSSN, *ASYNCADR, *ASYNCLOC, *RTLPASTHR, *SNAPASTHR	Required, Positional 1
CFGL	Configuration list	*Name*	Optional
APPNLCLE	APPN local location entry	Values (up to 50 repetitions): *Element list*	Optional
	Element 1: Local location name	*Communications name*	
	Element 2: Entry 'description'	*Character value*, ***BLANK**	
APPNRMTE	APPN remote location entry	Values (up to 50 repetitions): *Element list*	Optional
	Element 1: Remote location name	*Generic name, name*, *ANY	
	Element 2: Remote network identifier	*Communications name*, ***NETATR**, *NONE	
	Element 3: Local location name	*Communications name*, ***NETATR**	
	Element 4: Remote control point	*Communications name*, ***NONE**	
	Element 5: Control point net ID	*Communications name*, ***NETATR**, *NONE	
	Element 6: Location password	*Character value*, ***NONE**	
	Element 7: Secure location	*YES, ***NO**, *VFYENCPWD	
	Element 8: Single session	*YES, ***NO**	
	Element 9: Locally controlled session	*YES, ***NO**	
	Element 10: Pre-established session	*YES, ***NO**	
	Element 11: Entry 'description'	*Character value*, ***BLANK**	
	Element 12: Number of conversations	1-512, **10**	

Keyword	Description	Choices	Notes
ASYNCADRE	Async network address entry	Values (up to 50 repetitions): *Element list*	Optional
	Element 1: Network address	*Character value*	
	Element 2: Dial retry	1-255, **2**	
	Element 3: Entry 'description'	*Character value*, ***BLANK**	
ASYNCLOCE	Async remote location entry	Values (up to 50 repetitions): *Element list*	Optional
	Element 1: Remote location name	*Communications name*	
	Element 2: Remote location identifier	*Name*	
	Element 3: Entry 'description'	*Character value*, ***BLANK**	
RTLPASTHRE	Retail pass-through entry	Values (up to 50 repetitions): *Element list*	Optional
	Element 1: Retail device	*Name*	
	Element 2: SNUF device	*Name*	
	Element 3: Default host program	*Name*	
	Element 4: Entry 'description'	*Character value*, ***BLANK**	
FTRCPNAME	Filtered control point name	*Generic name, name*, ***ANY**	Optional
FTRCPNETID	Filtered CP network identifier	*Communications name*, ***NETATR**	Optional
LCLLOCNAME	Local location name	*Generic name, name*, ***ANY**	Optional
FTRACN	Filter action	*Character value*, ***ACCEPT**, *REJECT	Optional
APPNDIRE	APPN directory search entries	Values (up to 300 repetitions): *Element list*	Optional
	Element 1: Filtered CP location name	*Generic name, name*, ***ANY**	
	Element 2: Filtered CP location NETID	*Communications name*, ***NETATR**	
	Element 3: Partner location name	*Generic name, name*, ***ANY**	
	Element 4: Partner location network ID	*Communications name*, ***NETATR**	
	Element 5: Entry 'description'	*Character value*, ***BLANK**	
APPNSSNE	APPN session endpoint entries	Values (up to 300 repetitions): *Element list*	Optional
	Element 1: Remote location name	*Generic name, name*, ***ANY**	
	Element 2: Remote network identifier	*Communications name*, ***NETATR**	
	Element 3: Entry 'description'	*Character value*, ***BLANK**	
GRPNAME	SNA pass-through group name	*Name*	Optional
DEV	SNA pass-through device desc	Values (up to 254 repetitions): *Name*	Optional
TEXT	Entry 'description'	*Character value*, *BLANK	Optional

Keyword	Description	Choices	Notes
SNAPASTHRE	SNA pass-through entry	*Element list*	Optional
	Element 1: SNA pass-through group name	*Communications name*	
	Element 2: SNA pass-through device desc	Values (up to 254 repetitions): *Communications name*	
	Element 3: Entry 'description'	*Character value*, ***BLANK**	

Top

Configuration list type (TYPE)

Specifies the type of configuration list entry being added.

***APPNDIR**
An advanced peer-to-peer networking* (APPN*) directory search filter configuration list is used.

***APPNLCL**
An advanced peer-to-peer networking (APPN) local location list is used. Up to 476 APPN local location entries are allowed in the configuration list (using the CHGCFGL and CRTCFGL commands).

***APPNRMT**
An APPN remote location list is used. Up to 1898 APPN remote location entries are allowed in the configuration list (using the CHGCFGL and CRTCFGL commands).

***APPNSSN**
An APPN session end point filter configuration list is used.

***ASYNCADR**
An asynchronous network address list is used. Up to 294 asynchronous network address entries are allowed in the configuration list.

***ASYNCLOC**
An asynchronous remote location list is used. Up to 32000 asynchronous remote location entries are allowed in the configuration list.

***RTLPASTHR**
A retail pass-through list is used. Up to 450 retail pass-through entries can be specified in the configuration list.

***SNAPASTHR**
An SNA pass-through list is used. Up to 254 SNA pass-through entries can be specified in the configuration list.

Top

Configuration list (CFGL)

Specifies the name of the configuration list. This value is required and valid only when the configuration list is an asynchronous network address list (*ASYNCADR is specified for the **Configuration list type (TYPE)** parameter). The list types have system-supplied names: QAPPNLCL, QAPPNRMT, QASYNCADR, QASYNCLOC, QRTLPASTHR, and QSNAPASSTHR.

Top

APPN local location entry (APPNLCLE)

Specifies the APPN local location entry. This value is required if *APPNLCL is specified for the **Configuration list type** prompt (TYPE parameter).

You can enter multiple values for this parameter.

A maximum of 50 entries can be specified directly for this parameter. An entry consists of a value from each of the following elements.

local-location-name
Specify the location name residing on the local system. This name is used by APPN to determine if the request coming in is for this system or another system in the network. The local location name must be unique and cannot already exist as a remote location name used by configuration list QAPPNRMT, or be specified on another system as a local location in the same APPN network.

entry-description
Specify a short description of 20 characters or less for each local entry.

Top

APPN remote location entry (APPNRMTE)

Specifies the APPN remote location entry. This value is required if *APPNRMT is specified for the **Configuration list type** prompt (TYPE parameter).

You can enter multiple values for this parameter.

A maximum of 50 entries can be specified directly for this parameter. An entry consists of a value from each of the following elements.

remote-location-name
Specify the full name of a remote location or a generic name ending with an asterisk (*). The generic location name is used to allow one directory entry to be defined for all locations, on a single control point, whose name matches the characters preceding the asterisk. You can also specify *ANY so the system will accept all requests sent through it. Generic entries are only allowed from network nodes.

remote-network-identifier
Specify the network identifier of the network in which the remote location resides. The default of *NETATR uses the LCLNETID value from the system network attributes.

local-location-name
Specify the location name residing on the local system. This name is used by APPN to match a local/remote location pair entry. The default of *NETATR uses the LCLLOCNAME value from the system network attributes.

control-point-name
Specify the control point providing network functions for the remote location. By using this control point name (directory entry for the remote location), the network is searched more efficiently to find the location. This field is required if the remote location name is generic. The default is *NONE.

control-point-network-identifier
Specify the network identifier of the network in which the control point resides. The default of *NETATR uses the LCLNETID value from the system network attributes.

location-password
Specify the password that is used when establishing sessions on the local location/remote location name pair. This value must contain an even number of hexadecimal characters. The default is *NONE.

secure-location
Specifies how security information is handled for program start requests received from remote systems. The value is sent to the remote system when sessions are established. It is used in determining how allocate or evoke requests should be built. The value only applies to conversations started with the SECURITY(SAME) level of security.

***NO** The remote system is not a secure location. Security validation done by the remote system is not accepted. SECURITY(SAME) conversations are treated as SECURITY(NONE). No security information will be sent with allocate or evoke requests.

***YES** The remote system is a secure location and the local system will accept security validation done by remote systems. For SECURITY(SAME) conversations, the local system allows the remote system to verify user passwords. On the remote system, user IDs are retrieved from the operating system. The user IDs are then sent with an already verified indicator in the allocate or evoke requests.

***VFYENCPWD**
The remote system is not a secure location. For SECURITY(SAME) conversations, the remote system is not allowed to send the already verified indicator. On the remote system, user IDs and passwords are retrieved from the operating system. Passwords are then encrypted and sent with the user IDs in the allocate or evoke requests, to be verified by the local system. This value should only be used if the remote system is using i5/OS V3R2M0 or later. If the remote system does not support password protection then session establishment will not be allowed. For remote systems that support password protection, but do not support verification of encrypted passwords (VFYENCPWD), conversations will be treated as SECURITY(NONE).

number-of-conversations
Specify the number of conversations for a single session connection. The default number of conversations is 10. The default value must be used if single session is *NO. The valid range for the number of conversations is 1 through 512.

locally-controlled-session
Specify YES or NO to indicate whether a locally controlled session is allowed for this local location/remote location name pair. The default is *NO.

pre-established-session
Specify YES or NO to indicate whether the session is automatically bound when a connection is made between the local and remote location. The default is *NO.

entry-description
Specify a short description for each remote entry. The default is *BLANK.

Note: The combination of remote location name, network identifier, and local location name must be unique. Also, the remote location name can not already exist as a local location in configuration list QAPPNLCL, or as the current value for LCLLOCNAME or LCLCPNAME network attribute.

Top

Async network address entry (ASYNCADRE)

Specifies the asynchronous network address entry. This value is required if *ASYNCADR is specified for the **Configuration list type** prompt (TYPE parameter).

You can enter multiple values for this parameter.

A maximum of 50 entries can be specified directly for this parameter. An entry consists of a value from each of the following elements.

network-address
Specify the X.25 network address. This value must contain only digits 0-9.

dial-retry
Specify the number of times that dialing will be tried again when errors occur while dialing, before attempting to dial the next number on the list. The valid range of dial retries is 1-255.

entry-description
Specify a short description for each network address entry.

Top

Async remote location entry (ASYNCLOCE)

Specifies the asynchronous remote location entry. This value is required if *ASYNCLOC is specified for the **Configuration list type** prompt (TYPE parameter).

You can enter multiple values for this parameter.

A maximum of 50 entries can be specified directly for this parameter. An entry consists of a value from each of the following elements.

remote-location-name
Specify the name that, when combined with the remote location identifier, determines whether to accept an incoming call. It is the same as the name used in the remote system as it's local name. This value must be unique.

remote-location-identifier
Specify an identifier that, when combined with the remote location name, determines if an incoming call will be accepted. This identifier must be the same as the remote system has for its local identifier.

entry-description
Specify a short description for each remote location entry. The default is *BLANK.

Top

Retail pass-through entry (RTLPASTHRE)

Specifies the retail pass-through entry. This value is required if *RTLPASTHR is specified for the **Configuration list type** prompt (TYPE parameter).

You can enter multiple values for this parameter.

A maximum of 50 entries can be specified directly for this parameter. An entry consists of a value from each of the following elements.

retail-device-name
Specify the name of the retail device that communicates with the host. This value must be unique.

SNUF-device-name
Specify the name of the SNUF device through which the retail device communicates with the host. This value must be unique.

Default-host-program-name
Specify the name of the program to be started on the host if the program name is not present in the SNA command (INIT-SELF) that requests a session to be started.

entry-description
Specify a short description for each retail pass-through entry. The default is *BLANK.

Top

Filtered control point name (FTRCPNAME)

Specifies the control point name of the adjacent control point that is being filtered by the local system when a directory search request is made.

Note: This parameter is valid only if TYPE(*APPNDIR) is specified.

***ANY** Any control point name is filtered.

generic-filtered-CP-name*
Specify the generic control point name (part of a name followed by an asterisk) of the adjacent control point(s) being filtered. The generic control point name allows one directory entry to be defined for all control points, in a single network, with a name that matches the characters preceding an *.

filtered-CP-name
Specify the control point name of the adjacent control point being filtered.

Top

Filtered CP network identifier (FTRCPNETID)

Specifies the control point network identifier of the adjacent control point being filtered by the local system when a directory search request is made.

Note: This parameter is valid only if TYPE(*APPNDIR) is specified.

***NETATR**
The LCLNETID value specified in the system network attributes is used.

filtered-CP-network-ID
Specify the control point network identifier of the adjacent control point being filtered by the local system.

Top

Local location name (LCLLOCNAME)

Specifies the local location name being supplied by the caller that is being filtered by the local system. When the local system is initiating a session, this is the local location name being used. When a bind is received from another system, this is the Secondary Logical Unit (SLU) name being used.

Note: This parameter is valid only if TYPE(*APPNSSN) is specified.

***ANY** Any local location name will be filtered by the local system.

generic*-local-location-name
Specify the generic local location name (part of a name followed by an asterisk) of the local location(s) being filtered. The generic local location name allows one entry to be defined for all local location names, on the system, with a name that matches the characters preceding an *.

local-location-name
Specify the local location name that is being filtered by the local system.

Top

Filter action (FTRACN)

Specifies the filter action for APPN requests being handled by the local system.

Note: This parameter is valid only if TYPE(*APPNDIR) or TYPE(*APPNSSN) is specified.

***ACCEPT**
The request is accepted.

***REJECT**
The request is rejected.

Top

APPN directory search entries (APPNDIRE)

Specifies the APPN directory search entry being filtered by the local system. This parameter may be specified when *APPNDIR is specified for the TYPE parameter. Up to 300 entries may be specified at a time.

The possible Filtered Location Name values are:

***ANY** Any control point location will be filtered.

generic*-filtered-CP-loc-name
Specify the generic name (part of a name followed by an asterisk) of the control point location(s) to be filtered. The generic name allows one name to be specified for all control point locations with a name that matches the characters preceding an asterisk (*).

filtered-CP-location-name
Specify the control point location name to be filtered. This is the name of the location that is owned by the adjacent control point being filtered if the adjacent CP is an end node or LEN node. Or, the name of some location that accesses the local network via the adjacent control point (a non-native network node). This location name represents the name of the session partner attempting to establish a session with the remote location name (the location that exists in the local system's network).

The possible Filtered CP Location Network ID values are:

***NETATR**
The LCLNETID value specified in the system network attributes is used.

filtered-CP-location-network-ID
Specify the network identifier associated with the CP location name to be filtered.

The possible Partner Location Name values are:

***ANY** Any remote location will be filtered.

generic-partner-location-name*
Specify the generic name (part of a name followed by an asterisk) of the partner location(s) to be filtered. The generic name allows one name to be specified for all partner locations with a name that matches the characters preceding an asterisk (*).

partner-location-name
Specify the name of the partner location to be filtered.

The possible Partner Network Identifier values are:

***NETATR**
The LCLNETID value specified in the system network attributes is used.

partner-network-identifier
Specify the network identifier associated with the partner location to be filtered.

The possible Entry Description values are:

***BLANK**
Text is not specified.

'entry-description'
Specify a short description of 20 characters or less for each entry.

Top

APPN session endpoint entries (APPNSSNE)

Specifies the APPN session endpoint entry being filtered by the local system. This parameter may be specified when *APPNSSN is specified for the TYPE parameter. Up to 300 entries may be specified at a time.

The possible Remote Location Name values are:

***ANY** Any remote location will be filtered.

generic-remote-location-name*
Specify the generic name (part of a name followed by an asterisk) of the remote location(s) to be filtered. The generic name allows one name to be specified for all remote locations with a name that matches the characters preceding an asterisk (*).

The possible Remote Network Identifier values are:

***NETATR**
The LCLNETID value specified in the system network attributes is used.

Remote-network-identifier
Specify the remote network identifier associated with the remote location to be filtered.

The possible Text Description values are:

***BLANK**
Text is not specified.

'entry-description'
Specify a short description of 20 characters or less for each entry.

Top

SNA pass-through group name (GRPNAME)

Specifies the SNA pass-through group name of the configuration list entry being added. The group name has upstream SNA pass-through device names associated with it (DEV parameter) and must exist in the configuration list.

Note: This parameter is valid only if TYPE(*SNAPASTHR) is specified.

Top

SNA pass-through device desc (DEV)

Specifies the names of the upstream devices associated with the SNA pass-through group (GRPNAME parameter).

Note: This parameter is valid only if TYPE(*SNAPASTHR) is specified.

Top

Entry 'description' (TEXT)

Specifies the text that briefly describes the SNA pass-through group.

Note: This parameter is valid only if *APPNDIR, *APPNSSN, or *SNAPASTHR is specified for the TYPE parameter.

***BLANK**
Text is not specified.

'entry-description'
Specify a description of up to 50 characters for the SNA pass-through entry being added.

Top

SNA pass-through entry (SNAPASTHRE)

Specifies the SNA pass-through entry. This parameter can be specified if TYPE(*SNAPASTHR) is specified. However, because this parameter may be removed in a later release, whenever possible use GRPNAME, DEV, and TEXT parameters.

Top

Examples

```
ADDCFGLE  TYPE(*APPNLCL)
          APPNLCLE((LOC1 'location one') (LOC2 'location two'))
```

This command adds local locations LOC1 and LOC2 to configuration list QAPPNLCL.

Top

Error messages

***ESCAPE Messages**

CPF260F
Configuration list &1 not found.

CPF261C
Index for configuration list &1 not changed.

CPF261D
Index for configuration list &1 not changed.

CPF2613
Too many entries were added.

CPF2625
Not able to allocate object &1.

CPF263A
CFGL type &1 does not match existing type &2.

CPF2634
Not authorized to object &1.

CPF2663
Configuration list &1 previously deleted.

CPF9838
User profile storage limit exceeded.

Top

Add Keystore File Entry (ADDCKMKSFE)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The Add Keystore File Entry (ADDCKMKSFE) command stores the specified key value in a keystore file.

For more information on keystore, refer to the Cryptographic Services Keystore article in the Cryptographic Services section of the APIs topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Restrictions:

- You must have object operational (*OBJOPR), read (*READ) and add (*ADD) authorities to the keystore file.
- You must have execute (*X) authority to the directories in the path name prefix of the stream file if specified.
- You must have read (*R) authority to the stream file if specified.

Top

Parameters

Keyword	Description	Choices	Notes
KEYSTORE	Keystore file	*Qualified object name*	Required, Positional 1
	Qualifier 1: Keystore file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
RCDLBL	Record label	*Character value*	Required, Positional 2
KEYTYPE	Key type	*MD5, *SHA1, *SHA256, *SHA384, *SHA512, *DES, *TDES, *AES, *RC2, *RC4, *RSAPUB, *RSAPRV	Required, Positional 3
FORMAT	Key format	*BIN, *BER, *PEM	Required, Positional 4
STRING	Key string	*Element list*	Optional, Positional 5
	Element 1: Hexadecimal key value	*Character value*	
	Element 2: Length of key string	*Unsigned integer*	
STMF	Stream file	*Path name*	Optional
DISALLOW	Disallowed function	Values (up to 3 repetitions): ***NONE**, *ENCRYPT, *DECRYPT, *MAC, *SIGN	Optional

Top

Keystore file (KEYSTORE)

Specifies the keystore file to use.

This is a required parameter.

Qualifier 1: Keystore file

name Specify the name of the keystore file.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched.

name Specify the name of the library to search for the file.

Top

Record label (RCDLBL)

Specifies the label of a key record in the specified keystore file.

This is a required parameter.

character-value
Specify the key record label. The label can be up to 32 characters and contain any alphanumeric characters.

Top

Key type (KEYTYPE)

Specifies the algorithm type of the key.

This is a required parameter.

***MD5** An MD5 key is used for hash message authentication code (HMAC) operations. Because of weaknesses in the algorithm, MD5 should not be used except for compatibility purposes. The minimum length for an MD5 HMAC key is 16 bytes. A key longer than 16 bytes does not significantly increase the function strength unless the randomness of the key is considered weak. A key longer than 64 bytes will be hashed before it is used.

***SHA1**
An SHA-1 key is used for HMAC operations. Because of weaknesses in the algorithm, SHA-1 should not be used except for compatibility purposes. The minimum length for an SHA-1 HMAC key is 20 bytes. A key longer than 20 bytes does not significantly increase the function strength unless the randomness of the key is considered weak. A key longer than 64 bytes will be hashed before it is used.

***SHA256**
An SHA-256 key is used for HMAC operations. The minimum length for an SHA-256 HMAC key is 32 bytes. A key longer than 32 bytes does not significantly increase the function strength unless the randomness of the key is considered weak. A key longer than 64 bytes will be hashed before it is used.

***SHA384**
An SHA-384 key is used for HMAC operations. The minimum length for an SHA-384 HMAC key is 48 bytes. A key longer than 48 bytes does not significantly increase the function strength unless the randomness of the key is considered weak. A key longer than 128 bytes will be hashed before it is used.

***SHA512**
An SHA-512 key is used for HMAC operations. The minimum length for an SHA-512 HMAC key is 64 bytes. A key longer than 64 bytes does not significantly increase the function strength unless the randomness of the key is considered weak. A key longer than 128 bytes will be hashed before it is used.

***DES** An older, widely used symmetric encryption algorithm. DES should not be used except for compatibility purposes. Only 7 bits of each byte are used as the actual key. The last bit is sometimes used as a parity bit. Some cryptographic service providers require that a DES key have odd parity in every byte. The key size parameter must specify 8.

***TDES**
A symmetric encryption algorithm that improves the security of DES by performing the DES algorithm three times. Only 7 bits of each byte are used as the actual key. The last bit is sometimes used as a parity bit. Some cryptographic service providers require that a DES key have odd parity in every byte. The key size can be 8, 16, or 24. Triple DES operates on an encryption block by doing a DES encrypt, followed by a DES decrypt, and then another DES encrypt. Therefore, it actually uses three 8-byte DES keys. If the key is 24 bytes in length, the first 8 bytes are used for key 1, the second 8 bytes for key 2, and the third 8 bytes for key 3. If the key is 16 bytes in length, the first 8 bytes are used for key 1 and key 3, and the second 8 bytes for key 2. If the key is only 8 bytes in length, it will be used for all 3 keys (essentially making the operation equivalent to a single DES operation).

***AES** A newly developed symmetric encryption algorithm designed to replace DES. AES offers faster and stronger encryption than TDES. The key size can be 16, 24, or 32.

***RC2** A variable-key-size symmetric encryption algorithm. The key size can be 1 - 128.

***RC4** A variable-key-size symmetric stream encryption algorithm. The key size can be 1 - 256. Because of the nature of the RC4 operation, using the same key for more than one message will severely compromise security.

***RSAPUB**
An asymmetric encryption algorithm that uses a public/private key pair. The public key part can be used for encryption, or for verifying a digital signature. The FORMAT parameter must specify *BER or *PEM.

***RSAPRV**
An asymmetric encryption algorithm that uses a public/private key pair. The private key part can be used for decryption, or digital signature generation. The FORMAT parameter must specify *BER.

Top

Key format (FORMAT)

Specifies the format of the key string parameter.

This is a required parameter.

***BIN** The key is specified as a binary value. All key types, except RSA, must use this format.

***BER** If the key TYPE parameter specifies *RSAPUB, the key STRING parameter may specify the key in ASN.1 Basic Encoding Rules (BER) X.509 Certificate or SubjectPublicKeyInfo format. For

specifications of these formats, refer to RFC 3280. If the key TYPE parameter specifies *RSAPRV, the key STRING parameter must specify the key in BER encoded PKCS #8 format. For specifications of this format, refer to RSA Security Inc. Public-Key Cryptography Standards.

***PEM** If the key TYPE parameter specifies *RSAPUB, the key STRING parameter may specify the key in a Privacy Enhanced Mail (PEM) based certificate.

Top

Key string (STRING)

Specifies the key string. Either this parameter or the STMF parameter can be used to specify the key.

hexadecimal-value
Specify in hexadecimal (0-1, A-F) the key string. (Every two characters represent a byte.) For all but RSA keys, the key string is a simple binary value. For RSA keys, the key string must be in a special format. See the FORMAT parameter for more information.

Top

Stream file (STMF)

Specifies the path name for the stream file that contains the key string. Either this parameter or the STRING parameter can be used to specify the key.

path-name
Specify the path name of the stream file.

Top

Disallowed function (DISALLOW)

Specifies the functions that cannot be used with this key record. Multiple functions can be disallowed.

Single values

***NONE**
This key is allowed to be used in all cryptographic functions.

Other values (up to 3 repetitions)

***ENCRYPT**
This key is not allowed to be used in encryption operations.

***DECRYPT**
This key is not allowed to be used in decryption operations.

***MAC** This key is not allowed to be used in message authentication code (MAC) operations.

***SIGN**
This key is not allowed to be used in digital signing operations.

Top

Examples

Example 1: Add a Keystore File Entry Using a Key String

```
ADDCKMKSFE   KEYSTORE(MYLIB/MYKEYSTORE)  RCDLBL('Byllesby')
             KEYTYPE(*AES) FORMAT(*BIN)
             LEN(16) STRING(8276B09145C1324AC300D267F5D26694)
```

This command adds a 16-byte (128-bit) AES key to keystore file MYKEYSTORE in library MYLIB with label Byllesby.

Example 2: Add a Keystore File Entry Using a Stream File

```
ADDCKMKSFE   KEYSTORE(MYLIB/MYKEYSTORE)  RCDLBL('Pepin')
             KEYTYPE(*RSAPRV) FORMAT(*BER)
             LEN(*EOF) STMF('/myKeys/key1.SIGN')
             DISALLOW(*ENCRYPT *DECRYPT *MAC)
```

This command adds an RSA public/private key pair to keystore file MYKEYSTORE in library MYLIB with a label Pepin. The key is specified in BER-encoded format in stream file myKeys/key.SIGN. The length of keystring is the total length of data in the file. The resulting key record can only be used in digital signing and verification operations.

Top

Error messages

***ESCAPE Messages**

CPF3CF2
Error(s) occurred during running of &1 API.

CPF9872
Program or service program &1 in library &2 ended. Reason code &3.

CPF9D94
A pending value exists for a master key.

CPF9D9E
Record label already exists.

CPF9D9F
User not authorized to key store file.

CPF9DA0
Error opening key store file.

CPF9DA5
Key store file not found.

CPF9DA6
Key store file is not available.

CPF9DA7
File is corrupt or not a valid key store file.

CPF9DA9
The format of the PEM certificate is not valid.

CPF9DB3
Qualified keystore file name is not valid.

CPF9DB6
Record label not valid.

CPF9DB7
Error occured writing to the key store file.

CPF9DB8
Error occured reading record from key store.

CPF9DDA
Unexpected return code &1 from cryptographic service provider &2.

CPF9DDD
The key string length is not valid.

Top

Add Communications Entry (ADDCMNE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Communications Entry (ADDCMNE) command adds a communications entry to an existing subsystem description. Each communications entry describes one or more communications devices, device types, or remote locations for which the subsystem will start jobs when program start requests are received. The subsystem can allocate a communications device if the device is not currently allocated to another subsystem or job. A communications device that is currently allocated may eventually be de-allocated, making it available to other subsystems.

Restrictions:

1. To use this command, you must have:
   - object operational (*OBJOPR), object management (*OBJMGT), and read (*READ) authority to the specified subsystem description and execute (*EXECUTE) authority to the library containing that subsystem description.
   - object operational (*OBJOPR) and read (*READ) authority to the job description and execute (*EXECUTE) authority to the library containing that job description.
   - use (*USE) authority to the user profile.
2. Only a user with all object (*ALLOBJ) special authority is allowed to add an entry for which the job description does not exist.
3. A subsystem cannot allocate a communications device that is currently allocated to another subsystem or job.

Top

Parameters

Keyword	Description	Choices	Notes
SBSD	Subsystem description	*Qualified object name*	Required, Positional 1
	Qualifier 1: Subsystem description	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DEV	Device	*Generic name, name*, *ALL, *APPC, *ASYNC, *BSCEL, *FINANCE, *INTRA, *RETAIL, *SNUF	Optional, Positional 2
RMTLOCNAME	Remote location	*Communications name*	Optional, Positional 3
JOBD	Job description	Single values: ***USRPRF**, *SBSD Other values: *Qualified object name*	Optional, Positional 4
	Qualifier 1: Job description	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DFTUSR	Default user profile	*Name*, ***NONE**, *SYS	Optional
MODE	Mode	*Communications name*, ***ANY**	Optional
MAXACT	Maximum active jobs	0-1000, ***NOMAX**	Optional

Top

Subsystem description (SBSD)

Specifies the name and library of the subsystem description to which the communications entry is being added or in which it is being changed.

This is a required parameter.

Qualifier 1: Subsystem description

name Specify the name of the subsystem to which the communications entry is being added.

Note: The IBM-supplied object QSYSSBSD is not valid on this parameter.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the name of the subsystem description's library to which the communications entry is being added.

Top

Device (DEV)

Specifies the name of the device description or the type of the device being used with this communications entry.

Note: You must specify a value on either this parameter or the **Remote location (RMTLOCNAME)** parameter, but not both.

***ALL** All communications device types or names can be used with this communications entry.

***APPC**
All advanced program-to-program communications devices can be used with this communications entry. The devices created with the Create Device Desc (APPC) (CRTDEVAPPC) command can be used.

***ASYNC**
All asynchronous communications devices can be used with this communications entry. The devices created with the Create Device Desc (Async) (CRTDEVASC) command can be used. This value is valid only when *ANY is specified on the **Mode (MODE)** parameter.

***BSCEL**
All bisynchronous equivalency link communications devices can be used with this communications entry. The devices created with the Create Device Desc (BSC) (CRTDEVBSC) command can be used. This value is valid only when *ANY is specified on the **Mode (MODE)** parameter.

***FINANCE**
Specifies that all finance communications devices can be used with this communication entry. The devices created with the Create Device Desc (Finance) (CRTDEVFNC) command can be used by this communications entry. This value is valid only when *ANY is specified on the **Mode (MODE)** parameter.

***INTRA**
All INTRA communications devices can be used with this communications entry. The devices

created with the Create Device Create Device Desc (Intra) (CRTDEVINTR) command can be used. This value is valid only when *ANY is specified on the **Mode (MODE)** parameter.

***RETAIL**
All retail communications devices can be used with this communications entry. The devices created with the Create Device Desc (Retail) (CRTDEVRTL) command can be used by this communications entry. This value is valid only when *ANY is specified on the **Mode (MODE)** parameter.

***SNUF**
All SNA upline facility communications devices can be used with this communications entry. The devices created with the Create Device Desc (SNUF) (CRTDEVSNUF) command are can be used. This value is valid only when *ANY is specified on the **Mode (MODE)** parameter.

generic-name
Specify the generic name of the device description used with this communications entry.

name Specify the device description used with this communications entry.

Top

Remote location (RMTLOCNAME)

Specifies the name of the remote location used with this communications entry. The remote location name specified in the associated device description can be used here. No validity checking is done on the remote location name.

Note: You must specify a value for either this parameter or the **Device (DEV)** parameter, but not for both.

communications-name
Specify the name of the remote location used with this communications entry.

Top

Job description (JOBD)

Specifies the name and library of the job description used for jobs that are started as a result of receiving a program start request, and that are processed through this communications entry. If the job description does not exist when the entry is added or changed, a library qualifier must be specified because the qualified job description name is kept in the subsystem description.

Note: Only a user with all object (*ALLOBJ) special authority is allowed to add or change an entry for which the job description does not exist.

Single values

***USRPRF**
The job description name that is specified in the user profile of the user that made the program start request is used for jobs that are started through this communications entry.

***SBSD**
The job description having the same name as the subsystem description, specified on the **Subsystem description (SBSD)** parameter, is used for jobs started through this communications entry.

Qualifier 1: Job description

name Specify the name of the job description that is used for jobs started through this communications entry.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the library where the job description is located.

Top

Default user profile (DFTUSR)

Specifies the default user profile used for a program start request that contains no security information. This user profile is not used for program start requests that contain a password or specify a user profile (either valid or not valid).

***NONE**
No user profile is specified as the default.

***SYS** All user program start requests will be treated the same as *NONE. For program start requests sent by system functions, the request will run under a predetermined user profile if a user profile is not specified on the program start request.

name Specify the name of the user profile that is used for all program start requests that enter the system through this communications entry and that contain no password or user profile name.

Note: The names QDFTOWN, QLPINSTALL, QSECOFR, QSPL, QDOC, QDBSHR, QRJE, QTSTRQS, and QSYS are not valid entries for this parameter.

Top

Mode (MODE)

Specifies the mode name of the communications device or remote location name whose communications entry is being added or changed.

***ANY** Any available modes defined to the communications device or remote location are allocated to the subsystem. If the communications device does not have defined modes associated with it, the communications device itself is allocated to the subsystem.

name Specify the mode name of the communications device or remote location name that is being added or changed.

Note: The names SNASVCMG and CPSVCMG are not valid for this parameter.

Top

Maximum active jobs (MAXACT)

Specifies the maximum number of jobs (received program start requests) that can be active at the same time through this communications entry.

***NOMAX**
There is no maximum number of jobs that can be active at the same time through this communications entry.

1-1000 Specify the maximum number of jobs that can be active at the same time through this communications entry.

Top

Examples

```
ADDCMNE   SBSD(ALIB/SBS1)  DEV(COMDEV)
```

This command adds a communications entry for the APPC device named COMDEV and mode *ANY to the subsystem description SBS1, which resides in library ALIB. The DFTUSR parameter defaults to *NONE, which means that no jobs may enter the system through this entry unless valid security information is supplied on the program start request.

Top

Error messages

***ESCAPE Messages**

CPF1619
Subsystem description &1 in library &2 damaged.

CPF1691
Active subsystem description may or may not have changed.

CPF1697
Subsystem description &1 not changed.

Top

Add Community for SNMP (ADDCOMSNMP)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Community for SNMP (ADDCOMSNMP) command defines an SNMP community profile and adds it to the SNMP agent community list. An SNMP agent uses a community profile to determine whether or not to honor a request sent by an SNMP manager. The community profile consists of a community name, an object access specification, and a list of the SNMP managers that are part of the community. The community name combined with the ASCII community (ASCIICOM) parameter defines a community.

Multiple community profiles, each having a unique community name may exist in the SNMP agent community list at one time. Similarly, the same internet address may appear in more than one community profile.

The i5/OS SNMP agent does not support community views. A view is a subset of the objects in the management information base (MIB). Each i5/OS community consists of all of the objects in the MIB.

Restrictions: An SNMP manager sends three types of requests: get, get-next, and set. Get and get-next requests are used to read management information base (MIB) variables, and a set request is used to modify MIB variables. For a request from an SNMP manager to be accepted by the i5/OS SNMP agent, all of the following must be true:

1. The community name in the SNMP manager request specifies a defined community.
2. The internet address of the manager that sent the request must be listed in the community profile.
3. For a set request, the community object access must allow write operations to occur. For a get request or get-next request, read operations must be allowed.
4. For a set request, the object specified in the request must be able to be changed. For a get request or get-next request, the object must be readable.

Top

Parameters

Keyword	Description	Choices	Notes
COM	Community name	*Character value*	Required, Positional 1
ASCIICOM	Translate community name	***YES**, *NO	Optional
INTNETADR	Manager internet address	Single values: ***ANY** Other values (up to 300 repetitions): *Character value*	Optional
OBJACC	Object access	***SNMPATR**, *READ, *WRITE, *NONE	Optional
LOGSET	Log set requests	***SNMPATR**, *YES, *NO	Optional
LOGGET	Log get requests	***SNMPATR**, *YES, *NO	Optional

Top

Community name (COM)

Specifies the name of the SNMP community being added. Each SNMP community name must be unique.

The possible values are:

community-name
Specify the name of the SNMP community being added. The name may contain characters that cannot be displayed.

Top

Translate community name (ASCIICOM)

Specifies whether the community name is translated to ASCII characters when the community profile is added to the SNMP agent community list.

The possible values are:

***YES** The community name is translated to ASCII characters when the community profile is added to the SNMP agent community list. This value should be specified if the SNMP manager system defines its community names entirely of ASCII characters. An error message is sent if the community name cannot be translated to ASCII characters.

***NO** The community name is not translated to ASCII characters when the community profile is added to the SNMP agent community list. This value should be specified if the SNMP manager system defines its community names using EBCDIC characters or characters that cannot be displayed.

Top

Manager internet address (INTNETADR)

Specifies the internet addresses of the SNMP managers that are part of this community.

The possible values are:

***ANY** Allow any SNMP manager to be part of this community.

manager-internet-address
Specify the internet address of the SNMP manager. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes. Up to 300 unique internet addresses may be specified. The same internet address may appear in more than one community profile.

Top

Object access (OBJACC)

Specifies the object access for the community.

The possible values are:

***SNMPATR**
The object access defined with the Change SNMP Attributes (CHGSNMPA) command is used for this community.

***READ**
Allow SNMP managers that are part of this community to read all management information base (MIB) objects with get or get-next requests. Modification of MIB objects by SNMP managers is not permitted.

***WRITE**
Allow SNMP managers that are part of this community to change all MIB objects that are able to change with set requests. Specifying *WRITE implies *READ access.

***NONE**
Do not allow SNMP managers that are part of this community any access to MIB objects.

Top

Log set requests (LOGSET)

Specifies whether set requests from SNMP managers in this community are logged in journal QSNMP in library QUSRSYS.

The possible values are:

***SNMPATR**
The value defined with the Change SNMP Attributes (CHGSNMPA) command is used for this community.

***YES** Set requests are logged.

***NO** Set requests are not logged.

Top

Log get requests (LOGGET)

Specifies whether get requests and get-next requests from SNMP managers in this community are logged in journal QSNMP in library QUSRSYS.

The possible values are:

***SNMPATR**
The value defined with the Change SNMP Attributes (CHGSNMPA) command is used for this community.

***YES** Get requests and get-next requests are logged.

***NO** Get requests and get-next requests are not logged.

Top

Examples

```
ADDCOMSNMP   COM(ROCHESTER)  INTNETADR('8.6.5.4' '8.6.5.3')
             OBJACC(*WRITE)
```

This command adds the community ROCHESTER to the SNMP agent community list. SNMP managers with internet addresses 8.6.5.4 and 8.6.5.3 are the only managers in the community and are able to change all MIB objects.

Top

Error messages

***ESCAPE Messages**

TCP4001
Error occurred accessing SNMP configuration information.

TCP4008
Community already exists. Reason code &3.

TCP8050
*IOSYSCFG authority required to use &1.

Top

Add Directory Entry (ADDDIRE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Directory Entry (ADDDIRE) command allows you to add new entries to the system distribution directory. The directory contains information about a user, such as the user ID and address, system name, user profile name, mailing address, telephone, number, and other user information. The ADDDIRE command provides a parameter for each of the fields contained in the directory.

The ADDDIRE command does not provide interactive display support. This is provided by the Work with Directory Entries (WRKDIRE) command.

An X.400 originator/recipient (O/R) name can be added to the directory with this command. X.400 is an international standard for communications and the O/R name is the addressing information used in X.400 communications. The X.400 O/R name must be in character set 1169 and code page 500. This set includes A through Z, 0 through 9, and some special characters. Additional information on characters allowed is in the Globalization information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

NOTES

1. To prevent the system from changing lowercase characters to uppercase, enclose the values in apostrophes. This does not apply to user ID/address, system name/group, department, or X.400 O/R name.
2. Only the user ID/address, system name/group, department, and X.400 O/R name are translated from the graphic character identifier (GCID) specified by the CMDCHRID parameter. All other parameters are stored exactly as they are entered and the GCID is stored with them. The default GCID value is taken from the QCHRID system value. The user can override the defaults by specifying a character set and code page or by specifying *DEVD for the display device description.
3. Double-byte character set (DBCS) characters can be entered for the following system directory entry parameters:

```
USRD       LOCATION
LSTNAM     BLDG
FSTNAM     OFC
MIDNAM     ADDR1
PREFNAM    ADDR2
FULNAM     ADDR3
DEPT       ADDR4
TITLE      TEXT
CMPNY      USRDFNFLD
```

Restriction: You must have security administrator authority or be a security officer to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
USRID	User identifier	*Element list*	Required, Positional 1
	Element 1: User ID	*Character value*	
	Element 2: Address	*Character value*	

Keyword	Description	Choices	Notes
USRD	User description	*Character value*	Required, Positional 2
USER	User profile	*Name*, *NONE	Optional, Positional 3
SYSNAME	System name	Single values: ***LCL**, *PC, *ERROR Other values: *Element list*	Optional
	Element 1: System name	*Character value*	
	Element 2: System group	*Character value*	
NETUSRID	Network user ID	*Character value*, ***USRID**	Optional
LSTNAM	Last name	*Character value*, ***NONE**	Optional
FSTNAM	First name	*Character value*, ***NONE**	Optional
MIDNAM	Middle name	*Character value*, ***NONE**	Optional
PREFNAM	Preferred name	*Character value*, ***NONE**	Optional
FULNAM	Full name	*Character value*, ***DFT**	Optional
DEPT	Department	*Character value*, ***NONE**	Optional
TITLE	Job title	*Character value*, ***NONE**	Optional
CMPNY	Company	*Character value*, ***NONE**	Optional
TELNBR1	Telephone number 1	*Character value*, ***NONE**	Optional
TELNBR2	Telephone number 2	*Character value*, ***NONE**	Optional
FAXTELNBR	FAX telephone number	*Character value*, ***NONE**	Optional
LOC	Location	*Character value*, ***NONE**	Optional
BLDG	Building	*Character value*, ***NONE**	Optional
OFC	Office	*Character value*, ***NONE**	Optional
ADDR1	Address line 1	*Character value*, ***NONE**	Optional
ADDR2	Address line 2	*Character value*, ***NONE**	Optional
ADDR3	Address line 3	*Character value*, ***NONE**	Optional
ADDR4	Address line 4	*Character value*, ***NONE**	Optional
INDUSR	Indirect user	***NO**, *YES	Optional
PRTPERS	Print private mail	***NO**, *YES	Optional
PRTCOVER	Print cover page	***YES**, *NO	Optional
NFYMAIL	Mail notification	***SPECIFIC**, *ALLMAIL, *NOMAIL	Optional
NFYPTYPERS	Priority, private, important	***YES**, *NO	Optional
NFYMSGS	Messages	***YES**, *NO	Optional
TEXT	Text	*Character value*, ***NONE**	Optional
CMDCHRID	Command character identifier	Single values: ***SYSVAL**, *DEVD Other values: *Element list*	Optional
	Element 1: Graphic character set	*Integer*	
	Element 2: Code page	*Integer*	
COUNTRY	Country or region ID	*Character value*, ***NONE**	Optional
ADMD	Administration domain	*Character value*, ***NONE**	Optional
PRMD	Private management domain	*Character value*, ***NONE**	Optional
SURNAM	Surname	*Character value*, ***NONE**, *LSTNAM	Optional
GIVENNAM	Given name	*Character value*, ***NONE**, *FSTNAM	Optional
INITIALS	Initials	*Character value*, ***NONE**	Optional
GENQUAL	Generation qualifier	*Character value*, ***NONE**	Optional
ORG	Organization	*Character value*, ***NONE**	Optional

Keyword	Description	Choices	Notes
ORGUNIT	Organizational units	Single values: ***NONE** Other values (up to 4 repetitions): *Character value*	Optional
DMNDFNATR	Domain-defined attributes	Single values: ***NONE** Other values (up to 4 repetitions): *Element list*	Optional
	Element 1: Type	*Character value*	
	Element 2: Value	*Character value*	
USRDFNFLD	User-defined fields	Single values: ***NONE** Other values (up to 100 repetitions): *Element list*	Optional
	Element 1: Field name	*Character value*	
	Element 2: Product ID	*Character value*, *NONE	
	Element 3: Value	*Character value*	
MSFSRVLVL	Mail service level	Single values: ***USRIDX**, *SYSMS, *DOMINO Other values: *Element list*	Optional
	Element 1: Field name	*Character value*	
	Element 2: Product ID	*Character value*, *NONE	
PREFADR	Preferred address	*Element list*	Optional
	Element 1: Field name	*Character value*, ***USRID**, *ORNAME, *SMTP	
	Element 2: Product ID	*Character value*, *NONE	
	Element 3: Address type	*Character value*	
CCMAILADR	cc:Mail address	*Character value*, ***NONE**	Optional
CCMAILCMT	cc:Mail comment	*Character value*, ***NONE**	Optional
ALWSYNC	Allow synchronization	***YES**, *NO	Optional
DLOOWN	DLO owner	***USRPRF**, *GRPPRF	Optional

Top

User identifier (USRID)

Specifies the user ID and address for the new user. Both parts must be provided. The user ID and address are changed to upper case characters and are translated to the common character set and code page '697 500'. More information about specifying the user ID and address can be found in the SNA Distribution Services book, SC41-5410.

*ANY can be entered for the user ID and address. *ANY is the default entry that is used in distributions when an exact match is not made on the user ID and address, but the address matches a *ANY address. For example, *ANY CHICAGO is valid. HURST *ANY is not valid. Only one *ANY *ANY default entry is allowed in the directory.

This is a required parameter.

user-id address
Specify the user ID and address. A maximum of 8 characters can be used in each field.

If any lowercase characters are specified, the system changes them to, and stores them as, uppercase characters.

Top

User description (USRD)

Specifies the description associated with the user ID and address. For example, the description can contain the user's name or job title. By using a consistent naming convention such as ″last name, first name″ for the description, entries can be easily located while sorting by description from the Work with Directory display.

This is a required parameter.

user-description
Specify a maximum of 50 characters for the description associated with the user ID and address. The directory can contain more than one description for the same user ID and address, but each description must be unique for the same user ID and address.

Top

User profile (USER)

Specifies the user profile of the user being added to the directory. If the user being added is a local user, a valid profile must exist on the local system (and it must be specified). If the user is a remote user, no profile exists on the local system and *NONE should be specified.

***NONE**
The user being added to the directory is a remote user and has no local profile.

user-profile-name
Specify a maximum of 10 characters for the valid system user profile name. The user profile name is required for all local users. If a profile name is specified for a remote user, the profile name must be valid. In the latter case, mail is sent to the user at the remote location, but the user has local library access.

Top

System name (SYSNAME)

Specifies the one- or two-part name of the system on which the user works. If a two-part system name is specified on the command line, up to 8 characters make up both the system name and the system group name. The parts should be separated by at least one space.

A remote user can be added to the directory before the system name and system group are defined in the network tables, but distributions cannot be sent to that user until the remote system name and system group are defined. The remote system name and system group name are defined by using the Configure Distribution Services (CFGDSTSRV) command. Additional information on defining a remote system name and group name is in the SNA Distribution Services book, SC41-5410.

***LCL** The system name defaults to the local name. All local users being added to the directory should have *LCL specified as the system name.

***PC** *PC is for a distributed systems node executive (DSNX) user with a personal computer (PC) attached to this system.

***ERROR**
Use this value if your network contains a central system that receives all unresolved distributions. In this type of network, you may encounter distribution looping when a distribution cannot find a specific user ID on the intended system and the intended system has a ***ANY *ANY** entry directing distributions to the central system. The central system also has a default ***ANY address** entry directing unresolved distributions to the intended system. To prevent distribution looping, specify *ERROR as the system name for the default entry you are adding or changing. When a

distribution cannot find a specific user ID, but matches this default entry, the distribution is handled as a user that is not valid, just as if no directory match were found.

*ERROR is valid only when ***ANY address** or ***ANY *ANY** is specified on the **User identifier (USRID)** parameter.

system-name-group
Specify the one- or two-part name of the system to which distributions are sent. If a two-part system name is specified, each part can have a maximum of 8 characters. The name and group are changed to upper case characters and are translated to the common character set and code page '697 500'.

A remote user can be added to the directory before the system name and system group are defined in the network tables, but distributions cannot be sent to that user until the system name and system group are defined. The remote system name and system group are defined by using the Configure Distribution Services (CFGDSTSRV) command.

Top

Network user ID (NETUSRID)

Specifies the unique network user ID for this directory entry. This ID is used during directory shadowing to uniquely identify a user in a network.

***USRID**
Set the network user ID to the user ID and address associated with this directory entry.

network-user-ID
Specify the network user ID for the user. A maximum of 47 characters can be specified.

Top

Last name (LSTNAM)

Specifies the last name of the user. If no names are provided (last, first, middle, preferred, or full) but a value is specified on the **Department (DEPT)** parameter, the last name defaults to an asterisk (*).

***NONE**
No last name is specified for the user.

last-name
Specify a maximum of 40 characters for the last name of the user.

Top

First name (FSTNAM)

Specifies the first name of the user.

***NONE**
No first name is specified for the user.

first-name
Specify a maximum of 20 characters for the first name of the user.

Top

Middle name (MIDNAM)

Specifies the middle name of the user.

***NONE**
No middle name is specified for the user.

middle-name
Specify a maximum of 20 characters for the middle name of the user.

Top

Preferred name (PREFNAM)

Specifies the name by which the user likes to be addressed.

***NONE**
No preferred name is specified for the user.

preferred-name
Specify a maximum of 20 characters for the preferred name of the user.

Top

Full name (FULNAM)

Specifies the user's full name. This field can be entered by an administrator, or it can be built by the system. If the administrator enters data in this field, it is stored in the format in which it is entered. If the full name is not specified, it is created by the system as follows:

- Last, First Middle (Preferred)
- If parts of the full name are not provided, the missing parts are removed from this format.
- The preferred name, when it is specified, is always enclosed in parentheses.
- If a preferred name is specified but it does not fit in the 50 characters after the last, first, and middle names, the preferred name occupies the last positions of the full name field, replacing the characters in those positions. Therefore, if a preferred name is specified, it always appears.

Note: Only an administrator can change this field.

***DFT** The full name is built by the system from the values entered for the last name, first name, middle name, and preferred name.

full-name
Specify a maximum of 50 characters for the full name of the user.

Top

Department (DEPT)

Specifies the name or number of the department of which the user is a member.

Restriction: Only an administrator can change this field.

***NONE**
No department is specified for the user.

department-name
Specify a maximum of 10 characters for the name of the user's department.

Top

Job title (TITLE)

Specifies the user's job title, such as marketing director, account manager, or production engineer.

***NONE**
No job title is specified for the user.

job-title
Specify a maximum of 40 characters for the user's job title.

Top

Company (CMPNY)

Specifies the name of the company for which the user works.

***NONE**
No company name is specified for the user.

company
Specify a maximum of 50 characters for the company name.

Top

Telephone number 1 (TELNBR1)

Specifies the user's primary telephone number. The telephone number can be specified in any format appropriate to the user, including an international telephone number format.

***NONE**
No telephone number is specified.

telephone-number-1
Specify a maximum of 26 characters for the user's primary telephone number.

Top

Telephone number 2 (TELNBR2)

Specifies the user's secondary telephone number. The telephone number can be specified in any format appropriate to the user, including an international telephone number format.

***NONE**
A secondary telephone number is not specified.

telephone-number-2
Specify a maximum of 26 characters for the user's secondary telephone number.

Top

FAX telephone number (FAXTELNBR)

Specifies the user's facsimile telephone number. The facsimile telephone number can be specified in any format appropriate for the user, including an international telephone number format.

***NONE**
A facsimile telephone number is not specified.

facsimile-telephone-number
Specify a maximum of 32 characters for the user's facsimile telephone number.

Top

Location (LOC)

Specifies the location of the user's business or system.

***NONE**
No location is specified.

location
Specify a maximum of 40 characters for the location.

Top

Building (BLDG)

Specifies the name of the building in which the user works.

***NONE**
No building is specified.

building
Specify a maximum of 20 characters for the name of the building in which the user works.

Top

Office (OFC)

Specifies the name or number of the user's office.

***NONE**
No office is specified.

office Specify a maximum of 16 characters for the user's office.

Top

Address line 1 (ADDR1)

Specifies the user's mailing address. A maximum of 40 characters can be entered into each of these fields.

***NONE**
No address line is specified.

character-value
Specify the user's mailing address in any format.

Top

Address line 2 (ADDR2)

Specifies the user's mailing address. A maximum of 40 characters can be entered into each of these fields.

***NONE**
No address line is specified.

character-value
Specify the user's mailing address in any format.

Top

Address line 3 (ADDR3)

Specifies the user's mailing address. A maximum of 40 characters can be entered into each of these fields.

***NONE**
No address line is specified.

character-value
Specify the user's mailing address in any format.

Top

Address line 4 (ADDR4)

Specifies the user's mailing address. A maximum of 40 characters can be entered into each of these fields.

***NONE**
No address line is specified.

character-value
Specify the user's mailing address in any format.

Top

Indirect user (INDUSR)

Specifies whether the user being added is an indirect user. An indirect user is a local user who does not sign on the system to receive mail. Mail is automatically printed for the indirect user. Each indirect user must have a profile on the local system.

***NO** The user is not an indirect user.

***YES** The user is an indirect user.

Top

Print private mail (PRTPERS)

Specifies whether private mail for an indirect user is printed. Consideration should be given to restricting public access to the printer when private mail is printed. This parameter is ignored if the user is not an indirect user.

***NO** No private mail is printed for this indirect user.

***YES** Private mail is printed for this indirect user.

Top

Print cover page (PRTCOVER)

Specifies whether a cover page is printed when a mail item is printed. This parameter is ignored when a remote user is added to the directory because remote users do not receive mail on this system.

***YES** A cover page is printed when a mail item is printed.

***NO** A cover page is not printed when a mail item is printed.

Top

Mail notification (NFYMAIL)

Specifies whether the user is notified of the arrival of mail by receiving a message on the user's message queue. This parameter is ignored when a remote user is added to the directory because remote users do not receive mail on this system.

***SPECIFIC**
The user is notified only when the types of mail specified on the **Priority, private, important (NFYPTYPERS)** parameter and **Messages (NFYMSGS)** parameter arrive. These types of mail can include priority and personal mail, and messages.

***ALLMAIL**
The user is notified of the arrival of all types of mail.

***NOMAIL**
The user is not notified of the arrival of mail.

Top

Priority, private, important (NFYPTYPERS)

Specifies whether the user is notified of the arrival of priority, private, and important mail. This parameter is ignored if *NOMAIL or *ALLMAIL is specified on the **Mail notification (NFYMAIL)** parameter.

***YES** The user is notified of the arrival of priority, private, and important mail.

***NO** The user is not notified of the arrival of priority, private, and important mail.

Top

Messages (NFYMSGS)

Specifies whether the user is notified of the arrival of messages. This parameter is ignored if *NOMAIL or *ALLMAIL is specified on the **Mail notification (NFYMAIL)** parameter.

***YES** The user is notified of the arrival of messages.

***NO** The user is not notified of the arrival of messages.

Top

Text (TEXT)

Specifies the text that briefly describes the object.

***NONE**
No text is specified.

text Specify a maximum of 50 characters of text, enclosed in apostrophes, to describe additional information about the user.

Top

Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values. The value you specify on this parameter applies to the **User identifier (USRID)** parameter, **System name (SYSNAME)** parameter, **Department (DEPT)** parameter, and to all of the X.400 O/R name parameters.

Note:

- Only the user ID and address, system name and group, department, and the X.400 O/R parameters are translated to the graphic character set identifier (GCID) specified on this parameter. All other parameter values that you specify are stored exactly as they are entered; the GCID value is stored with them.
- If this command is run interactively, the default GCID value is taken from the display device description. If it is run in batch, the default GCID value is taken from the QCHRID system value. You can override these values by specifying a specific character set and code page on this parameter.

Note: Double byte character set (DBCS) characters can be entered on the following system directory entry parameters:

- **User description (USRD)**
- **Last name (LSTNAM)**
- **First name (FSTNAM)**
- **Middle name (MIDNAM)**
- **Preferred name (PREFNAM)**
- **Full name (FULNAM)**
- **Department (DEPT)**
- **Job title (TITLE)**
- **Company (CMPNY)**
- **Location (LOC)**
- **Building (BLDG)**
- **Office (OFC)**
- **Address line 1 (ADDR1)**
- **Address line 2 (ADDR2)**
- **Address line 3 (ADDR3)**
- **Address line 4 (ADDR4)**
- **Text (TEXT)**

Single values

***SYSVAL**
The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

***DEVD**
The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

Element 1: Graphic character set

1-32767
Specify the graphic character set to use.

Element 2: Code page

1-32767
Specify the code page to use.

Top

Country or region. (COUNTRY)

Specifies the country or region part of the X.400 Originator/Recipient (O/R) name.

***NONE**
No country or region is specified.

country or region-code
Specify an ISO 3166 Alpha-2 code or a CCITT country or region code from the ISO X.400 Code List Table. Refer to the COUNTRY parameter description in ″Expanded descriptions″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for the ISO X.400 Code List Table.

Top

Administration domain (ADMD)

Specifies the administration management domain part of the X.400 O/R name.

***NONE**
No administration management domain is specified.

administration-management-domain
Specify a maximum of 16 characters for the description of the administration management domain. An administration management domain is a public organization that handles a management domain, which is a set of message transfer agents and user agents that comprise a system capable of handling messages.

Top

Private management domain (PRMD)

Specifies the private management domain part of the X.400 O/R name.

***NONE**
No private management domain is specified.

private-management-domain
Specify a maximum of 16 characters for the description of the private management domain. A private management domain is a private company or noncommercial organization that handles a management domain, which is a set of message transfer agents and user agents that comprise a system capable of handling messages.

Top

Surname (SURNAM)

Specifies the X.400 user last name part of the personal name within the X.400 O/R name.

Note: This parameter is required when a value is specified on the GIVENNAM, INITIALS or GENQUAL, parameter.

***NONE**
No surname is specified.

***LSTNAM**
The user last name specified in the directory entry is used as the surname.

surname
Specify a maximum of 40 characters for the surname.

Top

Given name (GIVENNAM)

Specifies the X.400 user first name part of the personal name within the X.400 O/R name.

Note: The SURNAM parameter is required when a value is specified on this parameter.

***NONE**
No given name is specified.

***FSTNAM**
The user first name specified in the directory entry is used as the given name. It is truncated to 16 characters.

given-name
Specify a maximum of 16 characters for the given name.

Top

Initials (INITIALS)

Specifies the initials part of the personal name within the X.400 O/R name. For example, the initials for 'John Henry Smith' are 'JH.'

Note: The SURNAM parameter is required when a value is specified on this parameter.

***NONE**
No initials are specified.

initials
Specify a maximum of 5 characters for the initials.

Top

Generation qualifier (GENQUAL)

Specifies the generation qualifier part of the personal name within the X.400 O/R name. For example, the generation qualifier in the name 'John R. Smith, III' is 'III.'

Note: The SURNAM parameter is required when a value is specified on this parameter.

***NONE**
No generation qualifier is specified.

generation-qualifier
Specify a maximum of 3 characters for the generation qualifier.

Top

Organization (ORG)

Specifies the organization name part of the X.400 O/R name.

***NONE**
No organization name is specified.

organization
Specify a maximum of 64 characters for the organization name.

Top

Organizational units (ORGUNIT)

Specifies the organization-defined unit part of the X.400 O/R name.

***NONE**
No organizational unit is specified.

'organizational-unit'
Specify a maximum of 32 characters for the name of an organizational unit. Up to 4 organizational units can be listed in order of descending significance.

Top

Domain-defined attributes (DMNDFNATR)

Specifies the type and value of a domain-defined attribute not specified by X.400 standards but allowed in the X.400 O/R name to accommodate existing systems of sending messages. A maximum of 4 sets of attributes can be specified.

Note: To specify an X.121 address, type X.121 in the Type field, and then type the X.121 address in the Value field. A maximum of 15 numeric characters can be specified for the X.121 address.

If you are specifying other domain-defined attributes, the X.121 pair must be the last attribute specified. You must also specify a value for the Country or region field if you specify an X.121 address.

X.121 is a CCITT Recommendation that provides a method for the international numbering of X.25 packet-switching data networks.

The possible **type** values are:

***NONE**
No domain-defined type is specified.

type Specify a maximum of 8 characters for the type of the domain-defined attribute.

The possible **value** values are:

***NONE**
No domain-defined value is specified.

value Specify a maximum of 128 characters for the value of the domain-defined attribute.

Top

User-defined fields (USRDFNFLD)

Specifies the user-defined field names and values. A list of these user-defined field names can be displayed using the CHGSYSDIRA command and prompting with the F4 key. Up to 100 user-defined fields can be specified.

Note: The following SMTP user-defined fields are not always displayed when the CHGSYSDIRA command is prompted, but they can still be used in the user-defined field (USRDFNFLD) parameter to add SMTP information to the system distribution directory.

- SMTPAUSRID SMTP
- SMTPDMN SMTP
- SMTPRTE SMTP

The possible single value is:

***NONE**
No user-defined fields are specified.

The possible User-Defined Field Name value is:

field-name
Specify up to 10 characters for the user-defined field name.

The possible User-Defined Field Product ID values are:

***NONE**
No user-defined field product ID is specified.

product-ID
Specify up to 7 characters for the user-defined field product ID.

The possible User-Defined Field Value value is:

'value' Specify up to 512 characters for the value of the user-defined field value. Blanks are padded on the right.

Top

Mail service level (MSFSRVLVL)

Specifies the mail server framework service level for a local user. This value is ignored for a remote user. It indicates where mail is stored on the system.

***USRIDX**
The mail is stored in a user index.

***SYSMS**
The mail is stored in the Post Office Protocol (POP) mailbox, which can be accessed by POP clients on the personal computer through the System i5 POP server using the POP interface.

***DOMINO**
The mail is stored in the Lotus Domino mail database.

The possible Mail service level field-name value is:

field-name
Specify the field name when another mail service is used for this user. Specify up to 10 characters for the field name. This value should contain a user-defined field in the system directory that has been defined by the Change System Directory Attributes (CHGSYSDIRA) in the USRDFNFLD parameter with a field type of *MSFSRVLVL. The user-defined field specified here should then contain information needed by the mail server framework user exit program when the program is determining where to store the mail. The address resolution exit point name is QIBM_QZMFMSF_ADR_RSL. See the AnyMail/400 Mail Server Framework Support book, SC41-5411 for more information. This field could just be used as an indicator and the value does not have to be a user-defined field. When ever possible, the value specified here should be a user-defined field.

The possible mail service level product-ID values are:

***NONE**
No user-defined-field product ID is specified.

Product-ID
Specify up to 7 characters for the user-defined field product ID.

Top

Preferred address (PREFADR)

Specifies the preferred address for a user. This tells the mail server framework what fields to use in the system distribution directory for the preferred address of a user. Specify *USERID for SNADS. SNADS handles all the mail that goes to a user index including the gateway for X.400 O/R names and Simple Mail Transfer Protocol (SMTP) names.

The possible single values are:

***USRID**
The user ID/address is the preferred address for this user.

***ORNAME**
The X.400 O/R name is the preferred address for this user.

***SMTP**
The SMTP name is the preferred address for this user.

The possible Preferred address field-name value is:

field-name
Specify up to 10 characters for the field name. This value should contain an IBM-defined or a user-defined field in the system directory that has been defined by the Change System Directory Attributes (CHGSYSDIRA) command on the USRDFNFLD parameter with a field type of *ADDRESS. The field specified here should then contain information needed by the mail server framework user exit program when the program is determining where to store the mail. This field could just be used as an indicator and the value does not have to be an IBM-defined or user-defined field. Whenever possible, the value specified here should be an IBM-defined or a user-defined field.

The possible Preferred address product-ID values are:

***NONE**
No user-defined field product ID is specified.

***IBM** The field name is an IBM-defined field in the system distribution directory. Allowed IBM-defined field names are:

- USER (user profile)
- CCMAILADR (cc:Mail** address)
- FULNAM (full name)
- NETUSRID (network user ID)
- TELNBR1 (telephone number 1)
- TELNBR2 (telephone number 2)
- FAXTELNBR (facsimile telephone number)

Product-ID
Specify up to 7 characters for the user-defined field product ID.

The possible Preferred address address-type values are:

address-type
Specify up to 8 characters for the address type. The address type is a mail server framework type name that is specified on the Add Mail Framework Type Configuration (QzmfAddMailCfg) API. Whenever possible, this value should be one of the mail server framework configuration type names. When an address type is specified for a preferred address that a is special value, specify *N for the product ID.

Top

cc:Mail address (CCMAILADR)

Specifies the cc:Mail address for this user.

***NONE**
No cc:Mail address is specified.

'cc:Mail-address'
Specify the cc:Mail address. The address field has a maximum of 126 characters. If the address includes both a remote post office name and an alias name, the limit is 126 characters for each, with a space separating them (total 253). If the remote post office name contains spaces, the name needs to be put in quotation marks. This adds two characters to the limit for a total of 128 or 255 with the alias name.

Top

cc:Mail comment (CCMAILCMT)

Specifies the cc:Mail comment for this user.

***NONE**
No cc:Mail comment is specified.

'cc:Mail-comment'
Specify up to 126 characters for the cc:Mail comment field.

Top

Allow synchronization (ALWSYNC)

Specifies whether synchronization of this entry with other directories should be allowed.

***YES** Synchronization is allowed.

***NO** Synchronization is not allowed.

Top

DLO owner (DLOOWN)

Specifies if the user profile or the group profile will be assigned the ownership of the Document Library Objects (DLOs) for this directory entry.

Note: If this directory entry does not have a user profile in the User profile field, the value in the DLO owner field will be ignored.

***USRPRF**
The user profile associated with this directory entry is the owner of the newly created Document Library Objects (DLOs).

***GRPPRF**
The group profile specified in the user profile associated with this directory entry is made the owner of newly created DLOs and has all authority to the DLOs. If the group profile value is *NONE in the user profile, then the owner of the DLO is the user profile.

Top

Examples

Example 1: Adding a Local User

```
ADDDIRE   USRID(HURST PAYROLL)
          USRD('Manager of Payroll')  USER(ABHURST)
          LSTNAM(Hurst)  FSTNAM(Arthur)  PREFNAM(Art)  DEPT(55K)
          ADDR1('Dept55K/025-3')
          ADDR2('IBM Rochester')
          ADDR3('Highway 52 North')
          ADDR4('Rochester, MN 55904')
          LOC('Main Office')  BLDG(025-3)  OFC(E219)
          TELNBR1('435-422-2120')  TELNBR2('435-422-1012')
          FAXTELNBR('435-422-3296')  DLOOWN(*GRPPRF)
```

This command adds a local user to the distribution directory by allowing the system name parameter to default to *LCL. Since this is a local user, the user profile is specified. Address lines, location, and telephone numbers have been specified. Since the TEXT parameter is not used, it defaults to *NONE. This user is not an indirect user since the INDUSR parameter defaulted to *NO.

The user's last, first, and preferred names are specified. The full name was not specified, so FULNAM(*DFT) is used and will be created as, 'Hurst, Arthur (Art)'. This user has been added as a member of the department named 55K. If this department is searched, then 'Hurst, Arthur (Art)' will be included on the search list.

Any newly created DLOs associated with this directory entry, HURST PAYROLL, will be owned by the group profile specified in the Group Profile field in user profile ABHURST. The user entry is added to the directory if each one of the following is true:

1. A user ID and address HURST PAYROLL is not already in the directory.
2. The user profile name ABHURST is not already in the directory.

Example 2: Adding a Remote User

```
ADDDIRE   USERID(BYRD NEWYORK)  USRD('Arthur J. Byrd')
          USER(*NONE)  SYSNAME(BOCA)
          LOC('Boca Raton, Florida')  DEPT(61Q)
```

This command adds a remote user entry to the distribution directory. Since this is a remote user, the USER(*NONE) parameter is specified. The system name without the system group is specified. Except for the location, all of the parameters use default values. If the user-ID and address are unique, the user entry is added to the directory as a remote user.

If you are using directory shadowing, you do not need to add remote users as these users can be shadowed to your system.

If a department value is specified for this user, but no user name is specified, the last name is set to '*'. The full name is also '*' because it is created from the last name. This is done because the directory requires a non-blank name with department.

Top

Error messages

***ESCAPE Messages**

CPF8360
Not enough storage for commitment control operation.

CPF89A3
Operation not successful due to authority reasons.

CPF89A4
Operation not successful due to data validation reasons.

CPF8AA1
Library QUSRSYS not completely installed.

CPF90A8
*SECADM special authority required to do requested operation.

CPF9009
System requires file &1 in &2 be journaled.

CPF9024
System cannot get correct record to finish operation.

CPF905C
Error occurred trying to find a translation table.

CPF9082
User ID and address &1 &2 not added to directory.

CPF9096
Cannot use CMDCHRID(*DEVD), DOCCHRID(*DEVD) in batch job.

CPF9810
Library &1 not found.

CPF9838
User profile storage limit exceeded.

CPF9845
Error occurred while opening file &1.

CPF9846
Error while processing file &1 in library &2.

Top

Add Directory Shadow System (ADDDIRSHD)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Directory Shadow System (ADDDIRSHD) command adds a supplier system to supply system distribution directory data to your system through directory shadowing.

Restriction: To use this command, you must have security administrator (*SECADM) authority.

Top

Parameters

Keyword	Description	Choices	Notes
SYSNAME	System name	*Character value*	Required, Positional 1
SCD	Scheduled shadow	Single values: ***CURRENT** Other values: *Element list*	Optional
	Element 1: Date	*Date*	
	Element 2: Time	*Time*	
FRQ	Shadowing frequency	*DAILY, ***WEEKLY**, *BIWEEKLY, *MONTHLY, *MONTHLYREL, *HOURS	Optional
HOURS	Intervals of hours	1-999, **5**	Optional
SKIPDAY	Days to skip	Single values: ***NONE** Other values (up to 5 repetitions): *SUN, *MON, *TUE, *WED, *THU, *FRI, *SAT	Optional
MONTHWK	Week of the month	**4**, *LAST	Optional
INZ	Initial shadow	Single values: *NONAPPC, *COMPLETED Other values: *Element list*	Optional
	Element 1: Method	***APPC**	
	Element 2: Replace data	***NO**, *YES	
RMTLOCNAME	Remote location name	*Character value*, ***SYSNAME**	Optional
MODE	Mode	*Character value*, ***NETATR**	Optional
RMTNETID	Remote network identifier	*Character value*, ***LOC**, *NETATR, *NONE	Optional
LCLLOCNAME	Local location name	*Character value*, ***LOC**, *NETATR	Optional
TEXT	Text 'description'	*Character value*, ***SYSNAME**	Optional

Top

System name (SYSNAME)

Specifies a maximum of 8 characters for the name of the supplier system you are adding. You can specify uppercase letters A through Z, numbers 0 through 9, and special characters @, #, $, and embedded blanks. Embedded blanks must be enclosed in single quotation marks ('). Leading blanks are not allowed. The @, #, and $ characters are not recommended because they are not part of an invariant character set and are not available on all keyboards.

This is a required parameter.

Top

Scheduled shadow (SCD)

Specifies the date and time at which the system you are adding begins supplying data to your system.

***CURRENT**
The system begins shadowing data at the current date and time.

The possible **Shadow Date** values are:

scheduled-shadow-date
Specify the date on which the system begins supplying data to your system. The date must be specified in the job date format.

The possible **Shadow Time** values are:

scheduled-shadow-time
Specify the time at which the system begins supplying data to your system.

The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits where the time separator separates the hours, minutes, and seconds. If this command is entered from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

This is a required parameter.

Top

Shadowing frequency (FRQ)

Specifies the frequency with which the supplier system you are adding shadows data to your system, based on the value specified on the SCD parameter.

***WEEKLY**
Shadowing occurs once a week.

***DAILY**
Shadowing occurs once a day.

***BIWEEKLY**
Shadowing occurs every other week.

***MONTHLY**
Shadowing occurs on the same date every month.

***MONTHLYREL**
Shadowing occurs on the same relative day of the same relative week of every month, such as the first Monday of the month.

***HOURS**
Shadowing occurs in the interval specified on the HOURS parameter.

Top

Intervals of hours (HOURS)

Specifies the number of hours between shadows from the supplier system. This parameter is valid only when FRQ(*HOURS) is specified.

Top

Days to skip (SKIPDAY)

Specifies the days of the week when shadowing does not occur. A maximum of five values, other than *NONE, can be specified.

This parameter is valid only when FRQ(*DAILY) is specified.

***NONE**
No days are skipped.

***SUN** Sundays are skipped.

***MON**
Mondays are skipped.

***TUE** Tuesdays are skipped.

***WED** Wednesdays are skipped.

***THU** Thursdays are skipped.

***FRI** Fridays are skipped.

***SAT** Saturdays are skipped.

Top

Week of the month (MONTHWK)

Specifies whether shadowing that occurs on the same relative day of the month is scheduled to occur in the fourth week or the last week of the month.

This parameter is valid only when FRQ(*MONTHLYREL) is specified and when the SCD parameter date is the 22nd, 23rd, or 24th.

4 Shadowing occurs on the same relative day in the fourth week of the month.

***LAST**
Shadowing occurs on the same relative day in the last week of the month, whether or not the month has four or five weeks.

Top

Initial shadow (INZ)

Specifies the method used for the first shadow from the supplier system. The first shadow duplicates all of the local and shadowed data in the supplier system's distribution directory. Remote users are optionally supplied when the supplier specifies RMTSHD(*YES) on the Change Directory Attributes (CHGDIRA) command. Subsequent shadows include only data that has changed since the previous shadow.

The possible **Automatic Shadow** values are:

***APPC**
The first shadow occurs when this command is run using advanced program-to-program (APPC) communication. If you are adding a supplier system with a large directory, you may want to specify *NONAPPC to prevent the first shadow from tying up your communications lines.

When *APPC is specified, Element 2 allows you to specify whether the data in the fields of a directory entry on your system is replaced by shadowed data if the same entry also exists in the supplier system's directory.

The possible **Replace Data** values are:

***NO** The data in the fields of existing directory entries on your system is not replaced with data from the supplier system.

***YES** All shadowed data is added to your system distribution directory. The data in the fields of existing directory entries on your system is replaced with shadowed data if the same entry also exists in the supplier system's directory.

***NONAPPC**
The Copy to Directory (CPYTODIR) command is used for the first shadow. It is recommended that you run the CPYTODIR command before running this command. If shadowing from the supplier system you are adding starts before CPYTODIR is run, you may lose data.

***COMPLETED**
The initial shadow has already been done using the CPYTODIR command.

Top

Remote location name (RMTLOCNAME)

Specifies the remote location name of the supplier system you are adding.

***SYSNAME**
The value specified on the SYSNAME parameter is used for the remote location name.

remote-location-name
Specify the name of a remote location.

A maximum of 8 characters can be specified. The first character must be an uppercase letter A through Z, or special character $, #, or @. The name cannot contain a blank, plus sign (+), period (.), or an underscore (_). For more information, see the APPC Programmer's Guide.

Top

Mode (MODE)

Specifies the name of the mode that defines the sessions on the device used when shadowing data from the supplier system.

***NETATR**
The mode in the network attributes is used.

mode-name
Specify the mode name.

A maximum of 8 characters can be specified. The first character must be an uppercase letter A through Z, or special character $, #, or @. The name cannot contain a blank, plus sign (+), period (.), or an underscore (_). For more information, see the APPC Programmer's Guide.

Top

Remote network identifier (RMTNETID)

Specifies the supplier system's remote network identifier (ID).

***LOC** The remote network ID associated with the remote location is used. If several remote network IDs are associated with the remote location, the system determines which remote network ID is used.

***NETATR**
The remote network identifier specified in the network attributes is used.

***NONE**
No remote network identifier (ID) is used.

remote-network-ID
Specify a maximum of 8 characters for the remote network ID.

The first character must be an uppercase letter A through Z, or special character $, #, or @. The name cannot contain a blank, plus sign (+), period (.), or an underscore (_). For more information, see the APPC Programmer's Guide.

Top

Local location name (LCLLOCNAME)

Specifies the local location name. The local location name is used to identify your system to the supplier system you are adding.

***LOC** The local location name associated with the remote location is used.

***NETATR**
The LCLLOCNAME value specified in the system network attributes is used.

local-location-name
Specify a maximum of 8 characters for the local location name.

The first character must be an uppercase letter A through Z, or special character $, #, or @. The name cannot contain a blank, plus sign (+), period (.), or an underscore (_). For more information, see the APPC Programmer's Guide.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the object.

***SYSNAME**
The name specified on the SYSNAME parameter is used for the description.

'description'
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Examples

Example 1: Shadowing a System Weekly

```
ADDDIRSHD   SYSNAME(NYCITY)
            SCD( '92/05/01' '17:00:00' )
            FRQ(*WEEKLY)  TEXT('Shadow New York System')
```

This command adds the supplier system NYCITY, which starts shadowing directory data to the local system on May 1, 1992 at 5:00 p.m. The shadow frequency is once a week. The description of the shadow system is 'Shadow New York System'.

Example 2: Shadowing a Remote System Hourly

```
ADDDIRSHD   SYSNAME(CHICAGO)
            SCD('92/04/01' '20:00:00')
            FRQ(*HOURS)  HOURS(12)  RMTLOCNAME(CHIC01)
            LCLLOCNAME(CHICAGO1)  TEXT('Shadow Chicago System')
```

This command adds the supplier system CHICAGO, which starts shadowing directory data to the local system on April 1, 1992 at 8:00 p.m. The frequency of shadows is every 12 hours. The description of the shadow system is 'Shadow Chicago System'. The remote location name of the CHICAGO system is CHIC01 and the local location name is CHICAGO1.

Top

Error messages

***ESCAPE Messages**

CPF90A8
*SECADM special authority required to do requested operation.

CPF90FE
Add or change of shadow supplier &1 was not successful.

CPF905C
Error occurred trying to find a translation table.

CPF9838
User profile storage limit exceeded.

CPF9845
Error occurred while opening file &1.

CPF9846
Error while processing file &1 in library &2.

CPF9847
Error occurred while closing file &1 in library &2.

Top

Add DLO Authority (ADDDLOAUT)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Document Library Object Authority (ADDDLOAUT) command allows you to give a user access to a document or folder. It allows you to specify authority for users in the following ways:

- Give specific authority to a user.
- Give a set of users authority by specifying a previously defined authorization list.
- Give a group of users use (*USE) access by specifying an access code.

Restrictions:

The user of this command must have all (*ALL) authority to the objects, have all object (*ALLOBJ) special authority, or be the owner of the objects.

Top

Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	*Character value*, *ALL, *SYSOBJNAM	Required, Positional 1
FLR	Folder	*Character value*, ***NONE**	Optional
USRAUT	User authority	Single values: ***NONE** Other values (up to 50 repetitions): *Element list*	Optional
	Element 1: User profile	*Name*	
	Element 2: Authority level	*USE, *CHANGE, *EXCLUDE, *ALL	
AUTL	Authorization list	*Name*, ***NONE**	Optional
ACC	Document access code	Single values: ***NONE** Other values (up to 50 repetitions): 0-2047	Optional
SYSOBJNAM	System object name	*Name*	Optional

Top

Document library object (DLO)

Specifies the name of the document or folder to which authority is added.

***ALL** Authority is added to all objects in the specified folder. If *ALL is specified, a value must be specified on the **Folder (FLR)** parameter.

***SYSOBJNAM**
A system object name is used to identify the folder or document to which authority is added. You will be requested to enter the name.

name Specify the user-assigned name of the document or folder using a maximum of 12 characters.

Top

Folder (FLR)

Specifies the folder where the object specified for the **Document library object (DLO)** parameter is located.

***NONE**
A folder name is not specified. *NONE cannot be specified if either:

- A document or folder name is specified on the **Document library object (DLO)** parameter and the object is located in a folder
- *ALL is specified on the **Document library object (DLO)** parameter

name Specify the user-assigned name of the folder. The folder name can consist of a series of folder names if the object specified on the **Document library object (DLO)** parameter is located in a folder that is contained in another folder. A maximum of 63 characters can be specified.

Top

User authority (USRAUT)

Specifies the name of a specific user and the user's authority.

***NONE**
No additional user's authority is added.

name Specify the name of the user profile for whom specific authority is added.

***ALL** The user can perform all operations except those limited to the owner or controlled by authorization list management (*AUTLMGT) authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the object.

***CHANGE**
The user can perform all operations on the object except those limited to the owner or controlled by object existence (*OBJEXIST) and object management (*OBJMGT) authorities. The user can change and perform basic functions on the object. *CHANGE authority provides object operational (*OBJOPR) authority and all data authority. If the object is an authorization list, the user cannot add, change, or remove users.

***USE** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. Use (*USE) authority provides object operational (*OBJOPR), read (*READ), and execute (*EXECUTE) authorities.

***EXCLUDE**
The user cannot access the object.

Top

Authorization list (AUTL)

Specifies the name of the authorization list, if any, used to secure the document or folder specified on the **Document library object (DLO)** parameter.

***NONE**
An authorization list is not specified.

name Specify the name of a predefined authorization list.

Top

Document access code (ACC)

Specifies the access codes that are added. The access code must be defined to the system before it can be used. An access code of zero (0) allows all users use (*USE) authority for the object. Access code zero (0) cannot be assigned to an object if the security level is personal.

***NONE**
No access code is added.

0-2047 Specify the access code assigned to the object. All access codes except 0 must be predefined to the system using the Add Access Code (ADDACC) command.

Top

System object name (SYSOBJNAM)

Specifies the system object name of the document or folder. If a document or folder name or *ALL is specified on the **Document library object (DLO)** parameter, this parameter is ignored, and a folder name is required.

name Specify the system object name of the document or folder using the entire 10 characters.

Top

Examples

```
ADDDLOAUT   DLO(*ALL)  USER(MIKE (*CHANGE))  AUTL(*NONE)
            FLR(MYFLR)  ACC(1023)
```

This command adds *CHANGE authority for user MIKE to all objects in the folder MYFLR. An access code of 1023 was also added to the object.

Top

Error messages

***ESCAPE Messages**

CPF8A75
Not authorized to access folder &1.

CPF8A77
Folder &1 not found.

CPF8A78
Folder &1 in use.

CPF8A79
Folder &1 is logically damaged.

CPF8A80
Document &2 in use in folder &1.

CPF8A82
Document &2 not found in folder &1.

CPF8A83
Not authorized to access document &2 in folder &1.

CPF8A88
Operation not allowed on document &2 in folder &1.

CPF8A89
Document &2 in folder &1 is logically damaged.

CPF90BA
Authority request for document library object failed.

CPF901F
*AUTL was specified for a user other than *PUBLIC.

CPF9073
No authority to view or change the security of document library object &1.

CPF908A
Requester &1 not enrolled.

CPF908B
Document library object not found.

CPF908E
&1 objects changed; &2 objects not changed.

CPF909A
Document &2 in folder &1 is damaged.

CPF9095
Folder &1 is damaged.

Top

Add Distribution List Entry (ADDDSTLE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Distribution List Entry (ADDDSTLE) command allows you to add new entries to an existing distribution list. A distribution list can include local, remote, indirect, and programmable work station users. It can also include remote distribution lists, but not local distribution lists.

Up to 300 entries can be added to a distribution list at one time. In addition, up to 50 local distribution list IDs can be specified whose members are all to become part of this list.

The distribution list must exist before this command can be run. The Create Distribution List (CRTDSTL) command can be used to create a new distribution list.

Restriction: The user of this command must have security administrator authority to add entries to a distribution list owned by someone else. Users can add entries to a distribution list they have created without restrictions.

Top

Parameters

Keyword	Description	Choices	Notes
LSTID	List identifier	*Element list*	Required, Positional 1
	Element 1: List ID	*Character value*	
	Element 2: List ID qualifier	*Character value*	
USRID	User identifier	Single values: ***NONE** Other values (up to 300 repetitions): *Element list*	Optional
	Element 1: User ID	*Character value*	
	Element 2: Address	*Character value*	
	Element 3: User description	*Character value*, ***FIRST**	
FROMLSTID	From list identifier	Single values: ***NONE** Other values (up to 50 repetitions): *Element list*	Optional
	Element 1: List ID	*Character value*	
	Element 2: List ID qualifier	*Character value*	
CMDCHRID	Command character identifier	Single values: ***SYSVAL**, *DEVD Other values: *Element list*	Optional
	Element 1: Graphic character set	*Integer*	
	Element 2: Code page	*Integer*	

Top

List identifier (LSTID)

Specifies the two-part list identifier of the distribution list that is to have entries added.

This is a required parameter.

list-identification
Specify the list ID in two parts.

More information about specifying the list ID is in SNA Distribution Services book, SC41-5410.

If any lowercase characters are specified, the system changes them to, and stores them as, uppercase characters.

Top

User identifier (USRID)

Specifies the user ID, address, and description of the user that is to be added to the distribution list. Both the user ID and address must be provided. A list ID (both parts) is used in place of the user ID and address to identify a remote distribution list ID being added to the distribution list. A remote distribution list can be added to a local directory as a remote user, or can be valid because of a default *ANY entry. The description can be specified and it must exactly match the entry as it is known in the directory. If no description is specified, *FIRST is the default.

Up to 300 sets of user IDs, addresses, and descriptions can be specified. Each valid set is added to the distribution list.

***NONE**
No user ID is specified. If *NONE is specified, a value other than *NONE must be specified on the **From list identifier (FROMLSTID)** parameter.

user-id and address
Specify both the user ID and address.

If any lowercase characters are specified, the system changes them to, and stores them as, uppercase characters.

A remote list ID can also be typed in this field.

***FIRST**
The first description in the specified user ID and address is added. If only one entry exists, it is the one added to the list.

user-description
Specify a maximum of 50 characters for the description of the user. If a list ID is specified, enter the list description. You must enter the description exactly as it appears in the directory entry or an error message is returned.

Top

From list identifier (FROMLSTID)

Specifies the name of an existing distribution list whose entries are added to this list. A maximum of 50 list IDs can be specified. Duplicate entries are not removed.

***NONE**
No list ID is specified.

from-list-identification
Specify the two-part list ID of a distribution list whose entries are added to this list.

Top

Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data entered as command parameter values. The value you specify on this parameter applies to the **List identifier (LSTID)** parameter, the **User identifier (USRID)** parameter, and the **From list identifier (FROMLSTID)** parameter.

Note:

- Only the user ID and address, system name and group, department, and the X.400 O/R parameters are translated to the graphic character set identifier (GCID) specified on this parameter. All other parameter values that you specify are stored exactly as they are entered; the GCID value is stored with them.
- If this command is run interactively, the default GCID value is taken from the display device description. If it is run in batch, the default GCID value is taken from the QCHRID system value. You can override these values by specifying a specific character set and code page on this parameter.

Single values

***SYSVAL**
The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

***DEVD**
The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

Element 1: Graphic character set

1-32767
Specify the graphic character set to use.

Element 2: Code page

1-32767
Specify the code page to use.

Top

Examples

```
ADDDSTLE   LSTID(CHICAGO DLIST)
           USRID((HURST NEWYORK 'Manager of Payroll')
                 (LEE DEPT554 *FIRST)
                 (BOCA DLIST 'Remote Distribution list for Boca')
                 (ERIC WAREHSE))
           FROMLSTID((DEPT48K DLIST) (ALLMGRS DLIST))
```

This command specifies that four user IDs are added to the distribution list CHICAGO DLIST. The third user ID is in fact a remote distribution list. The fourth user ID (ERIC WAREHSE) defaults to the first description for that user ID. In addition, all of the entries in two distribution lists are added to this distribution list.

Top

Error messages

***ESCAPE Messages**

CPF9024
System cannot get correct record to finish operation.

CPF905C
Error occurred trying to find a translation table.

CPF9090
No entries added to distribution list &1 &2.

CPF9091
&1 entries added and &2 lists copied to list &3 &4. &5 entries not added and &6 lists not copied.

CPF9096
Cannot use CMDCHRID(*DEVD), DOCCHRID(*DEVD) in batch job.

CPF9838
User profile storage limit exceeded.

CPF9845
Error occurred while opening file &1.

CPF9846
Error while processing file &1 in library &2.

Top

Add Distribution Queue (ADDDSTQ)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Distribution Queue (ADDDSTQ) command allows you to add an entry to the distribution services queue table. Distribution queues are used to store distributions before they are sent or forwarded to other systems.

Interactive display support is provided by the Configure Distribution Services (CFGDSTSRV) command. More information about configuring a distribution network is in the SNA Distribution Services book, SC41-5410.

Distribution queue names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Restrictions:

- This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
- The combination of remote location name, mode, remote network identifier, and local location name must be unique within the type of distribution queue. This combination does not need to be unique within the system, for SNA distribution services (SNADS) distribution queues in the distribution services queue table (SNADS-type distribution queues), and for SystemView distribution services (SVDS) distribution queues (SVDS-type distribution queues). The default value *LOC, which can be specified on the RMTNETID parameter and the LCLLOCNAME parameter, and the default value *NETATR, which can be specified on the MODE parameter, represent any possible values that the system determines are not already configured for another SNADS or SVDS distribution queue of the same type.
- You must specify a unique remote location name for each RPDS-type distribution queue in the queue table. RPDS queues do not use modes, remote network identifiers, or local location names.
- Configuration of the routing table is not required for SVDS-type distribution queues. SVDS queues may be configured optionally in the SNADS routing table. However, normal SNADS mail can neither be routed to change management queues nor be received through change management connections, and change management connections can neither be routed to SNADS queues nor be received through SNADS connections.
- SVDS-type distribution queues can support only a single queue view (the queue is not divided into normal and priority halves). For configuration and operations purposes, only the normal queue is specified.
- Messages that report errors about distribution queues may display or print different characters than you entered for the distribution queue name because of internal system transformations. Similarly (depending on the language used for the work station), the internal value for a distribution queue name may differ from the characters shown for the Work with Distribution Queue (WRKDSTQ) command. An error may be reported if the character-string value specified for the **Distribution queue** prompt (DSTQ parameter) does not match the rules for an internal distribution queue value or if it does not match the internal value for any defined distribution queue (ignoring case differences).

Top

Parameters

Keyword	Description	Choices	Notes
DSTQ	Distribution queue	*Character value*	Required, Positional 1
RMTLOCNAME	Remote location	*Communications name*	Required, Positional 2
DSTQTYPE	Queue type	***SNADS**, *RPDS, *SVDS, *DLS	Optional
MODE	Mode	*Communications name*, ***NETATR**	Optional
RMTNETID	Remote network identifier	*Communications name*, ***LOC**, *NONE	Optional
LCLLOCNAME	Local location	*Communications name*, ***LOC**	Optional
NRMPTY	Normal priority	*Element list*	Optional
	Element 1: Send time	Single values: ***ALWAYS** Other values: *Element list*	
	Element 1: From time (HHMM)	0000-2359, ' '	
	Element 2: To time (HHMM)	0000-2359, ' '	
	Element 2: Force time (HHMM)	0000-2359, ***NONE**	
	Element 3: Send depth	1-999, **1**, *MANUAL	
HIGHPTY	High priority	*Element list*	Optional
	Element 1: Send time	Single values: ***ALWAYS** Other values: *Element list*	
	Element 1: From time (HHMM)	0000-2359, ' '	
	Element 2: To time (HHMM)	0000-2359, ' '	
	Element 2: Force time (HHMM)	0000-2359, ***NONE**	
	Element 3: Send depth	1-999, **1**, *MANUAL	
RTYNBR	Number of retries	0-9999, **3**	Optional
RTYITV	Retry interval	0-9999, **5**	Optional
SNDQ	Send while receiving	***NO**, *YES	Optional

Top

Distribution queue (DSTQ)

Specifies the name of the distribution queue being added to the distribution services queue table.

This is a required parameter.

Top

Remote location (RMTLOCNAME)

Specifies the name of the remote location to which distributions are sent from this distribution queue. The remote location name must be the same as the remote location name specified in the device description of the device used when sending distributions to another system from this distribution queue.

This is a required parameter.

Top

Queue type (DSTQTYPE)

Specifies the type of distribution queue being added.

The possible values are:

***SNADS**
SNADS is the distribution queue type. SNADS queues are used to send distributions within a SNADS network.

***DLS** DLS is the distribution queue type. DLS queues are used to communicate between your system and document library services on a remote system.

***RPDS**
RPDS is the distribution queue type. RPDS queues are used to communicate between your system and the System i5 VM/MVS bridge or JES (2,3) on System/370-type systems, and for the SNADS extended bridge function of the IBM Communications Utilities for i5/OS licensed program.

***SVDS**
SystemView distribution services (SVDS) is the distribution queue type. SVDS queues support the communications bridge between a SNADS network and IBM System Manager for i5/OS change management. An SVDS queue must be defined in order to receive from as well as send to a remote system using change management.

Top

Mode (MODE)

Specifies the name of the mode that defines the sessions on the device used by the distribution queue. This parameter is not valid if you specify *RPDS on the **Queue type** prompt (DSTQTYPE parameter).

The possible values are:

***NETATR**
The mode name defined in the network attributes is used.

mode-name
Specify a maximum of 8 characters for the name of the mode. Do not use CPSVCMG or SNASVCMG; these mode names are reserved for system use.

Top

Remote network identifier (RMTNETID)

Specifies the remote network identifier of the remote network to which this distribution queue sends distributions. This parameter is not applicable if you specify *RPDS on the **Queue type** prompt (DSTQTYPE parameter).

The possible values are:

***LOC** The remote network identifier defined in the device description used by this distribution queue is used.

***NONE**
No remote network identifier is specified.

remote-network-ID
Specify the remote network identifier.

Top

Local location (LCLLOCNAME)

Specifies the name used to identify your system to remote systems in the network. It is recommended that the name be the same as your system name.

The possible values are:

***LOC** The local location name defined in the device description used by this distribution queue is used.

local-location-name
Specify a maximum of 8 characters for the local location name.

Top

Normal priority (NRMPTY)

Specifies the queue sending conditions for distributions having a service level of data low. This parameter's three elements are:

Send time
The time period during which queued distributions of this priority are sent from this distribution queue. If you do not enter a time period, the transmissions are controlled by send depth and are not related to time.

Force time
A specific time during which distributions of this priority are sent regardless of queue depth. If *ALWAYS is specified for the send time, the force time can be set to any time of day. If you specify a specific to-time and from-time for the send time, the force time must occur within that time period.

Send depth
The number of distributions of this priority that must be on the queue before the system automatically begins sending them.

The possible **send time** values are:

***ALWAYS**
Distributions of this priority are sent from this distribution queue regardless of the time of day.

from-time to-time
Specify the time of day during which distributions of this priority can be sent from this distribution queue. The from-time and to-time must be specified in the 24-hour time format **hhmm** where **h** = hours and **m** = minutes. Both from-time and to-time must be specified.

The possible **force time** values are:

***NONE**
No force time is specified.

force-time
Specify the force time in the 24-hour format **hhmm** where **h** = hours and **m** = minutes.

The possible **send depth** values are:

1 Distributions are sent when they are put on the queue.

***MANUAL**
Distributions are sent only when an operator manually sends them using the Work with Distribution Queue (WRKDSTQ) command or the Send Distribution Queue (SNDDSTQ) command.

send depth
Specify the number of distributions of this priority that must be in this distribution queue before any are sent. Valid values range from 1 through 999.

Top

High priority (HIGHPTY)

Specifies the queue sending conditions for distributions having a service level of fast, status, or data high. This parameter's three elements are:

Send time
The time period during which queued distributions of this priority are sent from this distribution queue. If you do not enter a time period, the transmissions are controlled by send depth and are not related to time.

Force time
A specific time during which distributions of this priority are sent regardless of queue depth. If *ALWAYS is specified for the send time, the force time can be set to any time of day. If you specify a specific to-time and from-time for the send time, the force time must occur within that time period.

Send depth
The number of distributions of this priority that must be on the queue before the system automatically begins sending them.

The possible **send time** values are:

***ALWAYS**
Distributions of this priority are sent from this distribution queue regardless of the time of day.

from-time to-time
Specify the time of day during which distributions of this priority can be sent from this distribution queue. The from-time and to-time must be specified in the 24-hour time format **hhmm** where **h** = hours and **m** = minutes. Both from-time and to-time must be specified.

The possible **force time** values are:

***NONE**
No force time is specified.

force-time
Specify the force time in the 24-hour format **hhmm** where **h** = hours and **m** = minutes.

The possible **send depth** values are:

1 Distributions are sent when they are put on the queue.

***MANUAL**
Distributions are sent only when an operator manually sends them using the Work with Distribution Queue (WRKDSTQ) command or the Send Distribution Queue (SNDDSTQ) command.

send depth
Specify the number of distributions of this priority that must be in this distribution queue before any are sent. Valid values range from 1 through 999.

Top

Number of retries (RTYNBR)

Specifies the maximum number of times the system attempts to resend distributions from this distribution queue after a failure occurs. This parameter applies to communications line failures and recoverable distribution failures on a remote system. The SNADS job serving this distribution queue ends when the number of retries is exceeded.

The possible values are:

3 The system attempts to resend distributions a maximum of three times after a failure.

number
Specify the maximum number of times the system can attempt to resend distributions after a failure. Valid values range from 0 to 9999.

Top

Retry interval (RTYITV)

Specifies the interval (in minutes) between each retry attempt.

The possible values are:

5 The number of minutes between retries is 5.

minutes
Specify the interval (in minutes) between retries. Valid values range from 0 to 9999.

Top

Send while receiving (SNDQ)

Specifies whether this distribution queue ignores the send time and depth values specified on the **Normal priority** prompt (NRMPTY parameter) and **High priority** prompt (HIGHPTY parameter) and begins sending when a distribution is received from the SNADS system to which the queue sends its distributions.

This parameter is valid only if *SNADS is specified on the **Queue type** prompt (DSTQTYPE parameter).

The possible values are:

***NO** Distributions are sent from this queue only when the queue's sending conditions are met, or when the Send Distribution Queue (SNDDSTQ) or Work with Distribution Queue (WRKDSTQ) command is used to send the distributions.

***YES** This distribution queue begins sending when distributions are received from the SNADS system to which the queue sends its distributions regardless of the queue's sending conditions. Distributions are automatically sent for manual queues (queues that have no specified depth variable).

Top

Examples

Example 1: Adding a SNADS Distribution Queue

```
ADDDSTQ   DSTQ(CHICAGO)  RMTLOCNAME(CHICAGOLU)  MODE(NEWMODE)
```

This command adds a distribution queue named CHICAGO. The queue uses remote location name CHICAGOLU and mode NEWMODE when sending SNADS distributions.

Example 2: Adding a DLS Distribution Queue

```
ADDDSTQ   DSTQ(DLSQUEUE)  DSTQTYPE(*DLS)  RMTLOCNAME(DLSLU)
          MODE(DLSMODE)
```

This command adds a DLS type of distribution queue named DLSQUEUE. The queue uses remote location name DLSLU and mode DLSMODE when sending DLS requests.

Example 3: Adding an SVDS Distribution Queue

```
ADDDSTQ   DSTQ(CHICACM)  RMTLOCNAME(CHIGAGOLU)  DSTQTYPE(*SVDS)
```

This command adds an SVDS type of distribution queue named CHICACM. The queue uses remote location name CHICAGOLU when sending and receiving SVDS change management distributions.

Top

Error messages

***ESCAPE Messages**

CPF8802
Distribution queue &1 was not found.

CPF8807
Error occurred while using QSNADS journal.

CPF8809
Errors detected on SNADS internal queues.

CPF881D
High priority data not allowed for *SVDS distribution queues

CPF8826
Distribution queue entries exist for distribution queue &1.

CPF8827
Routing table entries exist for distribution queue &1.

CPF8828
Remote document library entries exist for *DLS distribution queue &1.

CPF8833
Distribution queue &1 already exists.

CPF8849
Queue &1 in use by another distribution services function.

CPF9845
Error occurred while opening file &1.

CPF9846
Error while processing file &1 in library &2.

CPF9847
Error occurred while closing file &1 in library &2.

CPF9899
Error occurred during processing of command.

Top

Add Distribution Route (ADDDSTRTE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Distribution Route (ADDDSTRTE) command allows you to add an entry to the distribution services routing table. The routing table determines which distribution queue receives a distribution on its way to a particular destination.

Distributions are routed to distribution queues based on service levels. One or more service levels must be specified for each routing table entry. Your system will not route distributions for service levels you have not configured. Normally, all service levels routed to the same destination use the same distribution queue. However, the user can configure several distribution queues for one destination based on distribution service levels.

System names, group names, and distribution queue names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Restrictions:

- This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
- An error occurs if a distribution route specifying a SystemView distribution services (SVDS) type of distribution queue includes another type of distribution queue (such as SNA distribution services (SNADS) or VM/MVS bridge (RPDS)).

Top

Parameters

Keyword	Description	Choices	Notes
SYSNAME	System name	*Element list*	Required, Positional 1
	Element 1: System name	*Character value*, *ANY	
	Element 2: System group	*Character value*, *ANY	
FAST	Fast service level	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Distribution queue name	*Character value*	
	Element 2: Hop count	1-255, ***NETATR**	
STATUS	Status service level	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Distribution queue name	*Character value*	
	Element 2: Hop count	1-255, ***NETATR**	
DATAHIGH	Data high service level	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Distribution queue name	*Character value*	
	Element 2: Hop count	1-255, ***NETATR**	

Keyword	Description	Choices	Notes
DATALOW	Data low service level	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Distribution queue name	*Character value*	
	Element 2: Hop count	1-255, ***NETATR**	
TEXT	Text	*Character value*, ***BLANK**	Optional

Top

System name (SYSNAME)

Specifies the system name and group name of the remote system that is the destination for this routing table entry.

You can specify a maximum of 8 characters for the system name and a maximum of 8 characters for the group name.

*ANY can be specified for the system name. When **SYSNAME(*ANY group)** is specified, any system in the group can be the destination for the routing table entry. Only one *ANY is allowed for each group in the routing table and is used to resolve a distribution destination that does not match a specific system name, but matches a group name.

*ANY can be specified for the group name only if *ANY is also specified for the system name. Only one **SYSNAME(*ANY *ANY)** entry is allowed in the routing table and is used to resolve a distribution destination that does not match any other routing table entries.

This is a required parameter.

Top

Fast service level (FAST)

Specifies the distribution queue and maximum hop count to the destination system for fast service level distributions. The fast service level is the highest priority service level.

The maximum hop count is the maximum number of times in a SNADS network that a distribution can be routed (hop) between systems at the SNADS level, including the hop to the final destination system. The maximum hop count does not include the hops made by advanced peer-to-peer networking (APPN) routing. If the maximum number of hops is exceeded, the distribution is ended and an error message is sent to the user who originally sent the distribution. Maximum hop count provides a means to prevent a distribution from looping indefinitely among systems.

The possible **distribution queue** values are:

***NONE**
: No distribution queue is specified for distributions requiring a fast service level. Distributions requiring fast service cannot be routed using this routing table entry.

distribution-queue-name
: Specify the name of the distribution queue to which distributions using this service level and routing entry are sent. The distribution queue must already exist and cannot be a DLS (document library services) type of queue.

The possible **maximum hop count** values are:

***NETATR**
The system network attributes value for the maximum hop count is used. You can display the current system value using the Display Network Attributes (DSPNETA) command.

hop-count
Specify the maximum hop count. Valid values range from 1 through 255.

Top

Status service level (STATUS)

Specifies the distribution queue and maximum hop count to the destination system for status service level distributions. The status service level is used for network status and other feedback information.

The maximum hop count is the maximum number of times in a SNADS network that a distribution can be routed (hop) between systems at the SNADS level, including the hop to the final destination system. The maximum hop count does not include the hops made by advanced peer-to-peer networking (APPN) routing. If the maximum number of hops is exceeded, the distribution is ended and an error message is sent to the user who originally sent the distribution. Maximum hop count provides a means to prevent a distribution from looping indefinitely among systems.

The possible **distribution queue** values are:

***NONE**
No distribution queue is specified for distributions requiring a status service level. Distributions requiring status service cannot be routed using this routing table entry.

distribution-queue-name
Specify the name of the distribution queue to which distributions using this service level and routing entry are sent. The distribution queue must already exist and cannot be a DLS (document library services) type of queue.

The possible **maximum hop count** values are:

***NETATR**
The system network attributes value for the maximum hop count is used. You can display the current system value using the Display Network Attributes (DSPNETA) command.

hop-count
Specify the maximum hop count. Valid values range from 1 through 255.

Top

Data high service level (DATAHIGH)

Specifies the distribution queue and maximum hop count to the destination system for data high service level distributions. The data high service level is used for high priority data traffic.

The maximum hop count is the maximum number of times in a SNADS network that a distribution can be routed (hop) between systems at the SNADS level, including the hop to the final destination system. The maximum hop count does not include the hops made by advanced peer-to-peer networking (APPN) routing. If the maximum number of hops is exceeded, the distribution is ended and an error message is sent to the user who originally sent the distribution. Maximum hop count provides a means to prevent a distribution from looping indefinitely among systems.

The possible **distribution queue** values are:

***NONE**
No distribution queue is specified for distributions requiring a data high service level. Distributions requiring data high service cannot be routed using this routing table entry.

distribution-queue-name
Specify the name of the distribution queue to which distributions using this service level and routing entry are sent. The distribution queue must already exist and cannot be a DLS (document library services) type of queue.

The possible **maximum hop count** values are:

***NETATR**
The system network attributes value for the maximum hop count is used. You can display the current system value using the Display Network Attributes (DSPNETA) command.

hop-count
Specify the maximum hop count. Valid values range from 1 through 255.

Top

Data low service level (DATALOW)

Specifies the distribution queue and maximum hop count to the destination system for data low service level distributions. The data low service level is used for most data traffic.

The maximum hop count is the maximum number of times in a SNADS network that a distribution can be routed (hop) between systems at the SNADS level, including the hop to the final destination system. The maximum hop count does not include the hops made by advanced peer-to-peer networking (APPN) routing. If the maximum number of hops is exceeded, the distribution is ended and an error message is sent to the user who originally sent the distribution. Maximum hop count provides a means to prevent a distribution from looping indefinitely among systems.

The possible **distribution queue** values are:

***NONE**
No distribution queue is specified for distributions requiring a data low service level. Distributions requiring data low service cannot be routed using this routing table entry.

distribution-queue-name
Specify the name of the distribution queue to which distributions using this service level and routing entry are sent. The distribution queue must already exist and cannot be a DLS (document library services) type of queue.

The possible **maximum hop count** values are:

***NETATR**
The system network attributes value for the maximum hop count is used. You can display the current system value using the Display Network Attributes (DSPNETA) command.

hop-count
Specify the maximum hop count. Valid values range from 1 through 255.

Top

Text (TEXT)

Specifies the text that briefly describes the object.

The possible values are:

***BLANK**
No text is specified.

'description'
Specify a maximum of 50 characters of text enclosed in apostrophes.

Top

Examples

Example 1: Adding an Entry for an Adjacent System

```
ADDDSTRTE   SYSNAME(SYSTEMA GROUPA)  FAST(SYSTEMA)
            STATUS(SYSTEMA)
            DATAHIGH(SYSTEMA)  DATALOW(SYSTEMA)
```

This command adds a routing table entry for a system that is directly connected to this system (via a physical advanced program-to-program communications (APPC) connection or a logical APPN connection). The distribution queue is given the same name as the destination system. The hop count defaults to the system default value.

Example 2: Adding a Generic Routing Table Entry

```
ADDDSTRTE   SYSNAME(*ANY GROUPNM1)  FAST(SYSTEMA)
            STATUS(SYSTEMA)
            DATAHIGH(SYSTEMA)  DATALOW(SYSTEMA)
```

This command adds a routing table entry for all systems in system group GROUPNM1. The distribution queue SYSTEMA is used to route distributions to all systems in the group.

Top

Error messages

***ESCAPE Messages**

CPF8801
Document library services (*DLS) queue &1 not allowed in routing table.

CPF8802
Distribution queue &1 was not found.

CPF8807
Error occurred while using QSNADS journal.

CPF881E
Distribution route contains combination of distribution queues that is not allowed

CPF881F
System group name cannot be blank for distribution route

CPF8815
Routing table entry &1 &2 not found.

CPF8831
Entry &1 &2 already exists in routing table.

CPF8837
System name/Group &1 &2 in use by another distribution services function.

CPF8849
Queue &1 in use by another distribution services function.

CPF9845
Error occurred while opening file &1.

CPF9846
Error while processing file &1 in library &2.

CPF9847
Error occurred while closing file &1 in library &2.

CPF9899
Error occurred during processing of command.

Top

Add Secondary System Name (ADDDSTSYSN)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Distribution Secondary System Name (ADDDSTSYSN) command allows you to add an entry to the distribution services secondary system name table. The table contains the names of all of the alternate (or alias) system names for which this system receives and may redirect distributions. The SNA distribution services (SNADS) function automatically directs distributions with the local system as the destination system name, so the local system cannot be added to the secondary system name table.

System names and group names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Restrictions:

- This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
- The secondary system name table does not operate with SystemView distribution services (SVDS) types of distributions.

Top

Parameters

Keyword	Description	Choices	Notes
SYSNAME	Secondary system name	*Element list*	Required, Positional 1
	Element 1: System name	*Character value*	
	Element 2: System group	*Character value*	
TEXT	Text	*Character value*, ***BLANK**	Optional

Top

Secondary system name (SYSNAME)

Specifies the alternate system name and system group name being added to the distribution services secondary system name table. You cannot specify the current system name unless you specify a non-blank system group name.

You can specify a maximum of 8 characters for the system name and a maximum of 8 characters for the group name.

This is a required parameter.

Top

Text (TEXT)

Specifies the text that briefly describes the object.

The possible values are:

***BLANK**
No text is specified.

'description'
Specify a maximum of 50 characters of text enclosed in apostrophes.

Top

Examples

```
ADDDSTSYSN   SYSNAME(SYS2LAJ1 ROCHESTR)
```

This command adds the system named SYS2LAJ1 ROCHESTR to the distribution services secondary system name table. The local system will receive distributions that contain SYS2LAJ1 ROCHESTR as the destination system name.

If the local system is named SYS2LAJ1, this command allows the local system to participate in a network that requires a group name of ROCHESTR for each participating system.

Top

Error messages

***ESCAPE Messages**

CPF8807
Error occurred while using QSNADS journal.

CPF8818
Secondary system name table entry &1 &2 not found.

CPF8835
System name/Group &1 &2 already specified.

CPF8837
System name/Group &1 &2 in use by another distribution services function.

CPF9845
Error occurred while opening file &1.

CPF9846
Error while processing file &1 in library &2.

CPF9847
Error occurred while closing file &1 in library &2.

CPF9899
Error occurred during processing of command.

Top

Add Data Definition (ADDDTADFN)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Data Definition (ADDDTADFN) command copies file, record format, and field definitions from an externally described database file to a data dictionary.

When definitions are added to a dictionary, the system does a search to find out if the dictionary contains a definition with the same name as the one being added. If an exact match of the definition is found, the existing definition is used. If an exact match is not found, a new version of the definition is created.

Database files using the following functions are not added to a dictionary:

- Access path sharing
- Alternative collating sequence
- Program-described files
- Join logical files
- Logical files with more than one format
- Logical files with select/omit specifications

Only format and field definitions of database files using the following functions are added to a dictionary:

- Field default values
- Field validity check codes
- Key fields defined using names in based on physical files
- Derived fields

Note: When adding a file that is already linked, the current link is ended and then the definition is added and linked.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	Data base file	*Qualified object name*	Required, Positional 1
	Qualifier 1: Data base file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DTADCT	Data dictionary	*Name*	Required, Positional 2
DFN	File definition	*Name*, ***FILE**	Optional, Positional 3

Top

Data base file (FILE)

Specifies the name and library of the externally described database file from which the definition is copied to the specified dictionary.

This is a required parameter.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the thread is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Data dictionary (DTADCT)

Specifies the name of the dictionary to which the definitions are added. This is a required parameter.

Top

File definition (DFN)

Specifies the name the file definition is given when it is copied into the dictionary.

The possible values are:

***FILE** The name of the file definition is the same as the database file name specified in the **Data base file** prompt (FILE parameter).

file-definition
Specify the name the file definition is given when copied to the data dictionary.

Top

Examples

```
ADDDTADFN   FILE(MYLIB/MYFILE)  DTADCT(MINE)  DFN(*FILE)
```

This command copies the definitions from MYFILE located in library MYLIB to the dictionary MINE. The file definition has the same name as the database file specified in the FILE parameter.

Top

Error messages

***ESCAPE Messages**

CPF2E9E
Not enough space to add to dictionary &1.

CPF2FE0
Error occurred while opening dictionary &1.

CPF2FE1
Error occurred while closing dictionary &1.

CPF2FE2
Dictionary &1 currently in use.

CPF2F02
Not authorized to use dictionary &1.

CPF2F07
Dictionary &1 in error.

CPF2F08
Dictionary &1 not found.

CPF2F61
File &2 in &3 currently in use.

CPF2F72
File &2 in &3 not valid for ADDDTADFN.

CPF2F73
ADDDTADFN not allowed for SQL database files.

CPF2F74
Attributes of file &2 in &3 not supported.

CPF9812
File &1 in library &2 not found.

CPF9820
Not authorized to use library &1.

CPF9822
Not authorized to file &1 in library &2.

Top

Add Disk Watcher Definition (ADDDWDFN)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Disk Watcher Definition (ADDDWDFN) command adds a new Disk Watcher definition to the system. A Disk Watcher definition identifies the performance data that is to be collected during a Disk Watcher collection. A session can be started using the Start Disk Watcher (STRDW) command. A definition is required when starting a new Disk Watcher collection.

Restrictions:

- To use this command, you must have service (*SERVICE) special authority, or be authorized to the Disk Watcher function of the Operating System through System i5 Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_DISK_WATCHER, can also be used to change the list of users that are allowed to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
DFN	Definition	*Name*	Required, Positional 1
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional
COLITV	Collection interval	1-3600, ***DYN**	Optional
ASP	ASP number	Single values: ***ALL**, *ASPDEV Other values (up to 10 repetitions): 1-255, *SYS	Optional
ASPDEV	ASP device	*Name*	Optional
DSKUNIT	Disk unit identifier	Single values: ***ALL** Other values (up to 10 repetitions): 1-6047	Optional
POOLID	Storage pool identifier	1-64, ***ALL**	Optional
TYPE	Type	***STATS**, *ALL, *TRACE	Optional
IOTYPE	I/O type	***ALL**, *READ, *WRITE	Optional
COLCOND	Collection condition	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Data type	*RESPTIME, *QTIME, *SERVTIME	
	Element 2: Relational operator	*LT, *GT, *LE, *GE	
	Element 3: Value	*Integer*	
COLRNG	Collection range	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Data type	*RESPTIME, *QTIME, *SERVTIME	
	Element 2: Lower value	*Integer*	
	Element 3: Upper value	*Integer*	
OBJINF	Object information	***NONE**, *ALL, *RESIDENT	Optional
FRCRCD	Force record write	***ITVEND**, *CALC	Optional
TOASPTHLD	To file ASP threshold	1-99, ***SYSTEM**	Optional

Keyword	Description	Choices	Notes
SYSASPTHLD	System ASP threshold	1-99, ***SYSTEM**	Optional

Top

Definition (DFN)

Specifies the name of the Disk Watcher definition being added. If the specified definition already exists an error condition will occur. This is a required parameter.

name Specify the name of the new Disk Watcher definition.

Top

Text 'description' (TEXT)

The text description of the Disk Watcher definition to be added.

***BLANK**
The Disk Watcher definition will have no text description.

character-value
Specify the text description for this Disk Watcher definition. The description should be no more that 50 characters of text, enclosed in apostrophes.

Top

Collection interval (COLITV)

Specifies the interval between retrieval of disk I/O data.

Note: Disk data is collected from a wrapping flight recorder. If the value specified for this parameter is too large, I/O operations may be missed. The record format for Disk Watcher database file QAPYDWINTI includes a field ('Data missed') which indicates whether I/O operations were missed during the interval.

***DYN** The time interval between samples is dynamic. This indicates that the interval seconds will be adjusted to the optimal value and may change during the collection. The optimal value is the largest interval time that can be used without missing data.

1-3600 Specify the number of seconds between intervals.

Top

ASP number (ASP)

Specifies the auxiliary storage pools for which disk I/O information will be collected. Selected ASPs must be varied on in order for data collection to occur.

Single values

***ALL** Disk I/O information will be collected for all ASPs on the system.

***ASPDEV**
Disk I/O information will be collected for the independent ASP specified for the **ASP device (ASPDEV)** parameter.

Note: A value must be specified for the ASPDEV parameter if this value is specified.

Other values (up to 10 repetitions)

***SYS** Disk I/O information will be collected for only the system ASP (ASP 1).

1-255 Specify the ASP number for which disk I/O information will be collected.

Top

ASP device (ASPDEV)

Specifies the independent auxiliary storage pool (ASP) device for which disk I/O information should be collected.

Note: A value must be specified for this parameter if *ASPDEV is specified for the **ASP number (ASP)** parameter.

name Specify the name of the independent ASP device for which disk I/O information should be collected. The device description associated with the independent ASP must be varied on and have a status of 'Available' in order for data collection to occur.

Top

Disk unit identifier (DSKUNIT)

Specifies the disk units for which I/O information should be collected. Disk unit identifiers from 4001 through 6047 are reserved for independent auxiliary storage pools.

Single values

***ALL** Disk I/O information will be collected for all disk units in the auxiliary storage pools specified by the **ASP number (ASP)** or **ASP device (ASPDEV)** parameter.

Other values (up to 10 repetitions)

1-6047 Specify the numeric identifier of each disk unit for which disk I/O information will be collected.

Top

Storage pool identifier (POOLID)

Specifies the main memory storage pool for which disk I/O information should be collected as data is read from disk into memory or written from memory back to disk. This parameter is a value ranging from 1 through 64, where pool 1 is the machine pool, and pool 2 is the base pool. This value also corresponds to the identifier shown on the Work with System Status (WRKSYSSTS) display.

Single values

***ALL** Disk I/O information will be collected for all storage pools.

Other values

1-64 Specify the pool number of the storage pool for which disk I/O information will be collected.

Top

Type (TYPE)

Specifies the type of disk I/O collection to be performed. This determines the detail level of the data which will be collected.

Note: Disk data is collected from a wrapping flight recorder. If the value specified for the **Collection interval (COLITV)** parameter is too large, I/O operations may be missed. The record format for Disk Watcher database file QAPYDWINTI includes a field ('Data missed') which indicates whether I/O operations were missed during the interval.

***STATS**
A statistics collection will be performed. This type of collection will summarize the I/O data into a set of counters based on the type of I/O operation. A statistics collection will always include all I/O operations on the system. This means that the I/O operations included in the statistics data will not be limited by values specified for the **Collection condition (COLCOND)** or **Collection range (COLRNG)** parameters.

Data for a statistics collection is written to database files QAPYDWRUNI, QAPYDWINTI, and QAPYDWSTAT in the library specified for the **Library (LIB)** parameter.

***ALL** Both statistics and trace data will be collected and written to the Disk Watcher database files.

***TRACE**
A trace collection will be performed. This type of collection will gather data for every I/O which occurred on the specified ASP or disk unit(s). The trace data collected with this option may be limited to I/O operations which meet a certain condition. To collect a limited set of trace data, either specify a condition on the **Collection condition (COLCOND)** parameter or a range on the **Collection range (COLRNG)** parameter.

Data for a trace collection is written to the database files QAPYDWRUNI, QAPYDWINTI, QAPYDWTRC, and QAPYDWTDER in the library specified for the **Library (LIB)** parameter. Additionally, data will be written to the database files QAPYDWPGMR and QAPYDWOBJR if *ALL or *RESIDENT is specified for the **Object information (OBJINF)** parameter.

Top

I/O type (IOTYPE)

Specifies whether the I/O trace data collected should be limited to a particular type of I/O operation.

Note: This parameter will only limit the collection of I/O trace data. Statistics data will always include all I/O operations.

***ALL** All types of disk I/O operations will be included in the disk I/O trace data.

***READ**
Only operations that read data from disk storage will be included in the disk I/O trace data.

***WRITE**
Only operations that write data to disk storage will be included in the disk I/O trace data.

Top

Collection condition (COLCOND)

Specifies whether the I/O trace data collected should be limited to I/O operations which meet a specified condition.

Note: This parameter will only limit the collection of I/O trace data. Statistics data will always include all I/O operations.

Note: Either a single limiting value may be specified for this parameter or a range may be specified for the COLRNG parameter. You cannot specify values for both parameters.

Single values

***NONE**
No limiting condition is specified.

Element 1: Data type

***RESPTIME**
The conditional trace data collection will be based on the total response time of the I/O operation.

***QTIME**
The conditional trace data collection will be based on the queue (wait) time of the I/O operation.

***SERVTIME**
The conditional trace data collection will be based on the service time of the I/O operation.

Element 2: Relational operator

***LT** Collect trace records for disk I/O operations that took less than the number of microseconds specified for element 3 of this parameter.

***GT** Collect trace records for disk I/O operations that took more than the number of microseconds specified for element 3 of this parameter.

***LE** Collect trace records for disk I/O operations that took not more than the number of microseconds specified for element 3 of this parameter.

***GE** Collect trace records for disk I/O operations that took not less than the number of microseconds specified for element 3 of this parameter.

Element 3: Value

integer
Specify the number of microseconds to use to limit trace record collection.

Top

Collection range (COLRNG)

Specifies whether the I/O trace data collected should be limited to I/O operations which are within a range of time values.

Note: This parameter will only limit the collection of I/O trace data. Statistics data will always include all I/O operations.

Note: Either a single limiting value may be specified for the **Collection condition (COLCOND)** parameter or a range may be specified for this parameter. You cannot specify values for both parameters.

Single values

***NONE**
No limiting range is specified.

Element 1: Data type

***RESPTIME**
The conditional trace data collection will be based on the total response time of the I/O operation.

***QTIME**
The conditional trace data collection will be based on the queue (wait) time of the I/O operation.

***SERVTIME**
The conditional trace data collection will be based on the service time of the I/O operation.

Element 2: Lower value

integer
Specify the lower limit of a range to determine which I/O operations will be included in the trace collection.

Element 3: Upper value

integer
Specify the upper limit of a range to determine which I/O operations will be included in the trace collection.

Top

Object information (OBJINF)

Specifies whether Disk Watcher will gather object and program information, such as the object name and program name, and whether this data collection will be limited to objects and programs which are in resident storage.

During *TRACE collections, Disk Watcher collects information related to the object on which the I/O operation was performed. When this object is in resident storage, the time required to gather the data is relatively small. However, if the object is not in resident storage, there will be a more significant performance impact associated with gathering this information. The same is true when collecting information related to the program or procedure which initiated the I/O operation.

***NONE**
Object and program information will not be collected for any objects or programs.

***ALL** Object and program information will be collected for all objects and programs. This includes data for objects and programs which are not in resident storage.

***RESIDENT**
Object and program information will only be collected for objects and programs which are in resident storage.

Top

Force record write (FRCRCD)

Specifies when disk I/O data records will be written to the Disk Watcher database files.

***ITVEND**
Collected data will be written to the database files at the end of each interval.

***CALC**
Collected data will be written to the files at a time determined by the system. Data will always be available in the files after disk I/O data collection is ended.

Top

To file ASP threshold (TOASPTHLD)

Specifies the percentage of the auxiliary storage pool (ASP) that contains the Disk Watcher database files which can be used before the collection is forced to end. Because the amount of data collected can be very large, this parameter allows you to limit how much of the ASP is consumed. If the database files exist in the system ASP and values are specified in both this parameter and the **System ASP threshold (SYSASPTHLD)**, the SYSASPTHLD parameter will override the value specified here.

***SYSTEM**
The threshold which is configured on the system for this ASP. This value is a percentage which is configured using the Change Storage Threshold function of the Start System Service Tools (STRSST) command. Data collection will be forced to end if this percentage of the ASP storage is consumed.

1-99 Specify the percentage of the database files' ASP which may be used before disk I/O data collection will be forced to end. For example, if you specify 80, data collection will be forced to end if more than 80% of the ASP storage is consumed. A value specified on this parameter will override the threshold which is configured on the system.

Top

System ASP threshold (SYSASPTHLD)

Specifies the percentage of the system auxiliary storage pool (ASP) which can be used before the collection is forced to end. Because Disk Watcher allocates temporary storage and the amount of data collected can be very large, this parameter allows you to limit how much of the system ASP is consumed. If the database files exist in the system ASP and values are specified in both this parameter and the **To file ASP threshold (TOASPTHLD)** parameter, the value specified here will take precedence.

***SYSTEM**
The threshold which is configured on the system for the system ASP. This value is a percentage which is configured using the Change Storage Threshold function of the Start System Service Tools (STRSST) command. Data collection will be forced to end if this percentage of the system ASP storage is consumed.

1-99 Specify the percentage of the system ASP which may be used before disk I/O data collection will be forced to end. For example, if you specify 80, data collection will be forced to end if more than 80% of the system ASP storage is consumed. A value specified on this parameter will override the threshold which is configured on the system.

Top

Examples

Example 1: Add a Disk Watcher Definition Which Collects Data for Specific ASPs

```
ADDDWDFN   DFN(ASPNBR)  ASP(2 16)  TYPE(*STATS)  COLITV(5)
```

This command will create a Disk Watcher definition called ASPNBR. This definition will collect summarized statistics data for auxiliary storage pool (ASP) 2 and ASP 16. Data will be collected at an interval of 5 seconds and will include all I/O operations which occur in the specified ASPs. To collect Disk Watcher data using this definition use the **Start Disk Watcher (STRDW)** command with ASPNBR specified for the **Definition (DFN)** parameter.

Example 2: Add a Disk Watcher Definition Which Collects Data for a Specific Disk Unit

```
ADDDWDFN   DFN(DSKNBR)  ASP(*SYS)  DSKUNIT(12)  TYPE(*ALL)
```

This command will create a Disk Watcher definition called DSKNBR. This definition will collect data for disk unit 12 in the system auxiliary storage pool (ASP). Both detailed trace data and summarized statistics data will be collected. To collect Disk Watcher data using this definition use the **Start Disk Watcher (STRDW)** command with DSKNBR specified for the **Definition (DFN)** parameter.

Example 3: Add a Disk Watcher Definition Which Traces Only I/O Operations Which Meet a Collection Condition

```
ADDDWDFN   DFN(RESPTIME)  COLTYPE(*TRACE)
           COLCOND(*RESPTIME *GT 100)
```

In this example, a Disk Watcher definition called RESPTIME will be created. This definition will collect data for all ASPs on the system and only trace data will be collected. The I/O operation data collected for the trace will include only those disk operations whose total response time was greater than 100 microseconds. To collect Disk Watcher data using this definition use the **Start Disk Watcher (STRDW)** command with RESPTIME specified for the **Definition (DFN)** parameter.

Example 4: Add a Disk Watcher Definition Which Traces Only I/O Write Operations

```
ADDDWDFN   DFN(WRITEOPS)  TYPE(*TRACE)  IOTYPE(*WRITE)
```

In this example, a Disk Watcher definition called WRITEOPS will be created. This definition will collect data for all ASPs on the system. Only trace data for I/O operations which write disk data to storage will be collected. To collect Disk Watcher data using this definition use the **Start Disk Watcher (STRDW)** command with WRITEOPS specified for the **Definition (DFN)** parameter.

Top

Error messages

***ESCAPE Messages**

CPF0001
Error found on &1 command.

CPFAF10
Definition or filter already exists.

CPFB513
The user does not have the required authority.

CPFB51C
Error on Add Disk Watcher Definition (ADDDWDFN) command.

Top

Add Configuration Entry (ADDEMLCFGE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Emulation Configuration Entry (ADDEMLCFGE) command is used to add a configuration entry for a 3270 device emulation session to the configuration file. You can use this command to specify display and printer emulation options, such as setting the maximum image size for a display, or tracing the data stream for a printer.

The values you specify are used during an emulation session when the configuration entry is specified on the Start 3270 Display Emulation (STREML3270) or Start Printer Emulation (STRPRTEML) commands.

Top

Parameters

Keyword	Description	Choices	Notes
EMLCFGE	Configuration entry	*Name*, QEMDFTCFGE	Required, Positional 1
EMLDBGJOB	Debug job	***NOTRACE**, *TRACE	Optional
EMLSIG	Handle signals	***SAVE**, *IGNORE	Optional
EMLATR	Handle attributes	***IGNORE**, *REJECT	Optional
EMLMAXSCR	Maximum screen size	***DEVD**, *MOD2, *MOD5	Optional
EMLTRC	VLIC trace	***NOTRACE**, *TRACE	Optional
EMLSTR	Start printer TRCJOB	***NOTRACE**, *TRACE	Optional
EMLINLSCN	Initial screen	***YES**, *NO	Optional
EMLGRDLIN	Gridline	***NO**, *YES	Optional
EMLDBCS	Graphic DBCS	***NO**, *YES	Optional
EMLPRTFMT	Printout formatting	***NO**, *YES	Optional
EMLSNACLR	Clear processing	***RETRY**, *IGNORE	Optional
EMLBUF	Gridline Buffer	***FRONT**, *BACK	Optional
EMLVLG	Conditional vlog generated	***NO**, *YES	Optional
EMLSCS	Handle SCS command errors	***NO**, *YES	Optional
EMLNML	Enforce 5250 Numeric Lock	***NO**, *YES	Optional
EMLMSG	Send SNA Unbind message	***NO**, *YES	Optional

Top

Configuration entry (EMLCFGE)

Specifies the name of the emulation configuration entry that you are adding to the configuration file.

Note: The configuration file is shipped with the default configuration entry for 3270 emulation sessions, QEMDFTCFGE.

This is a required parameter.

Top

Debug job (EMLDBGJOB)

Specifies whether to trace the printer data stream being passed to the printer function manager when a job that is using printer emulation is being debugged with the Trace Job (TRCJOB) command.

***NOTRACE**
The printer data stream is not traced in the TRCJOB output.

***TRACE**
The printer data stream is traced in the TRCJOB output.

Top

Handle signals (EMLSIG)

Specifies how an emulation job that is not in send mode responds to a Systems Network Architecture (SNA) request for permission to send data (signal).

***SAVE**
The emulation job stores the SNA signal, and if appropriate grants the host permission to send when a piece of data with the change direction indicator (CD) on it is received.

***IGNORE**
The emulation job does not grant the host permission to send.

Top

Handle attributes (EMLATR)

Specifies how the 3270 emulation job responds to an incorrect character attribute or attribute value that is received during a single-byte character set (SBCS) session. In 3270 data stream terminology, a **character attribute** is a set attribute instruction (SA order) and an **attribute value** is a value on which the instruction operates (value).

***IGNORE**
The emulation job ignores the incorrect value.

***REJECT**
The emulation job sends a negative response to the data stream containing the incorrect character attribute or attribute value.

Top

Maximum screen size (EMLMAXSCR)

Specifies the maximum size of the image to be shown on the display screen.

***DEVD**
The maximum size is whatever the device can support.

***MOD2**
The maximum size is 24 rows by 80 columns.

***MOD5**
The maximum size is 27 rows by 132 columns.

Note: If the device does not support 27 rows by 132 columns, the maximum size defaults to 24 rows by 80 columns.

Top

VLIC trace (EMLTRC)

Specifies whether trace points are issued by the data stream translation component of the Licensed Internal Code (LIC) when the following are true:

- The job is using the data stream translation API (application program interface).
- The Trace Job (TRCJOB) command is running on the job using the data stream translation API.
- A LIC source/sink trace is also running.

***NOTRACE**
Data stream translation trace points are not issued.

***TRACE**
Data stream translation trace points are issued.

Top

Start printer TRCJOB (EMLSTR)

Specifies whether to call the Trace Job (TRCJOB) command when a printer emulation job is started.

***NOTRACE**
The TRCJOB command is not called.

***TRACE**
The TRCJOB command is called.

Top

Initial screen (EMLINLSCN)

Specifies whether to show the Emulation Initialization In Progress display when a 3270 device emulation session is starting.

***YES** The display is shown.

***NO** The display is not shown.

Top

Gridline (EMLGRDLIN)

Specifies whether to suppress field outlining on the display.

***NO** Field outlining is not suppressed.

***YES** Field outlining is suppressed.

Top

Graphic DBCS (EMLDBCS)

Specifies whether to support the DBCS-graphic character string for input fields.

Note: This parameter is valid only for customer applications that specify field attributes or that specify character attributes for the entire input field. If only part of a field is defined with the DBCS-graphic attribute, you will get unpredictable results.

***NO** The DBCS-graphic character string is not supported.

***YES** The DBCS-graphic character string is supported.

Top

Printout formatting (EMLPRTFMT)

Specifies whether to use the values supplied by the STRPRTEML command for the number of lines (NUMLIN) parameter and the number of columns (NUMCOL) parameter instead of using the values supplied by the printer file.

Note: This parameter is valid only for SNA character string (SCS) printer sessions.

***NO** The STRPRTEML command values for lines and columns are not used.

***YES** The STRPRTEML command values for lines and columns are used.

Note: The STRPRTEML command values remain in effect until a set horizontal format or a set vertical format command is received from the host.

Top

Clear processing (EMLSNACLR)

Specifies how the 3270 emulation job recovers when an attempt to get data sent by the host system fails after an SNA CLEAR command is received.

Note: This parameter is valid only for display emulation sessions.

***RETRY**
The emulation job tries again to get the data.

***IGNORE**
The emulation job does not try again to get the data.

Top

Gridline Buffer (EMLBUF)

Specifies whether the foreground or background buffer is used when base gridlines are displayed. This parameter does not apply when strategic gridlines are used.

***FRONT**
The foreground buffer is used.

***BACK**
The background buffer is used.

Top

Conditional vlog generated (EMLVLG)

Specifies whether to generate a VLIC log when the datastream translation routines send a negative response to the host because a command or order was not valid.

***NO** The VLIC log is not generated.

***YES** The VLIC log is generated. The emulation session continues.

Top

Handle SCS command errors (EMLSCS)

For SCS printer sessions only, this parameter says whether to follow architecture and default to a page size or page width of 1 line when a bad set horizontal format (SHF) or a bad set vertical format (SVF) is received.

***NO** The architecture is followed. Page size is one line.

***YES** The architecture is not followed, page size and width default to values entered on the start printer emulation (STRPRTEML) command.

Top

Enforce 5250 Numeric Lock (EMLNML)

Specifies whether to enforce 5250 Numeric Lock by the data stream translation component of the Licensed Internal Code (LIC) when the job is using the data stream translation API (application program interface).

***NO** 5250 Numeric Lock is not enforced.

***YES** 5250 Numeric Lock is enforced.

Top

Send SNA Unbind message (EMLMSG)

Specifies whether to display a message on the screen when an unbind has been received.

***NO** No message is displayed.

***YES** Message is displayed.

Top

Examples

Example 1: Adding a Configuration Entry

```
ADDEMLCFGE   EMLCFGE(ARTSDEPT)  EMLMAXSCR(*MOD5)
             EMLGRDLIN(*YES)
```

This command adds an emulation configuration entry named ARTSDEPT for a display with a maximum screen image of 27 rows by 132 columns. Field outlining does not show on the display.

Example 2: Adding a Configuration Entry for a Printer

```
ADDEMLCFGE   EMLCFGE(FASBPRINT)  EMLDBGJOB(*TRACE)
             EMLATR(*REJECT)
```

This command adds an emulation configuration entry named FASBPRINT to the configuration file. The FASBPRINT configuration entry traces the printer data stream when a job is running with a trace on it and sends a negative response when incorrect data is received.

Top

Error messages

***ESCAPE Messages**

CPF853A
Emulation entry &1 already exists.

CPF854B
Internal error in emulation configuration routines.

Top

Add Environment Variable (ADDENVVAR)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The Add Environment Variable (ADDENVVAR) command adds an environment variable consisting of a character string in the form 'environment variable name=environment variable value'. Environment variables can be used, for example, to specify configuration values to application programs on systems that are compliant with the Single UNIX Specification.

If you are not ready to set the environment variable value, you can use this command to add an environment variable with a null value. You can then use the Add Environment Variable (ADDENVVAR) or Change Environment Variable (CHGENVVAR) command to associate the environment value with the environment variable name.

Restriction: You must have *JOBCTL special authority to use this command to add system-level environment variables.

Top

Parameters

Keyword	Description	Choices	Notes
ENVVAR	Environment variable	*Character value*	Required, Positional 1
VALUE	Initial value	*Character value*, ***NULL**	Optional, Positional 2
CCSID	Coded character set ID	1-65535, ***JOB**, *HEX	Optional, Positional 3
LEVEL	Level	***JOB**, *SYS	Optional
REPLACE	Replace existing entry	***NO**, *YES	Optional

Top

Environment variable (ENVVAR)

Specifies the name of the environment variable to be added. If an environment variable by this name currently exists at the specified level (LEVEL parameter), error message CPFA980 is issued, unless the REPLACE(*YES) option is used, in which case the variable is set to the new value specified.

ADDENVVAR limits environment variable name to a maximum of 128 bytes in length. Valid values include all EBCDIC characters except the equal sign (=), the null-terminator (X'00') and blank (X'40'). The name must be enclosed in apostrophes if it contains any non-alphanumeric character. If an apostrophe is intended, two apostrophes must be used ('').

Note: The case is preserved when lowercase characters are specified.

This is a required parameter.

Top

Initial value (VALUE)

Specifies the environment variable value.

The possible values are:

***NULL**
The value of the environment variable is the null character (X'00').

environment-variable-value
Specify the value of the environment variable. ADDENVVAR limits value to a maximum of 1024 bytes in length. Valid values include all EBCDIC characters. The value must be enclosed in apostrophes if it contains any non-alphanumeric character or blanks. If an apostrophe is intended, two apostrophes must be used ('').

Note: The case is preserved when lowercase characters are specified.

Top

Coded character set ID (CCSID)

Specifies the coded character set identifier (CCSID) of the text supplied on the ENVVAR and the VALUE parameters. This value is stored with the environment variable.

The possible values are:

***JOB** The CCSID of the text is assumed to be the CCSID of the job running this command.

***HEX** The CCSID of 65535 is stored with this environment variable.

coded-character-set-identifier
Specify the CCSID to be stored with the environment variable. Valid values range from 1 through 65535.

Top

Level (LEVEL)

Specifies the level of the environment variable.

The possible values are:

***JOB** This is a job-level environment variable.

***SYS** This is a system-level environment variable.

Top

Replace existing entry (REPLACE)

Specifies whether the value of an existing environment variable should be reset to the new value.

The possible values are:

***NO** Do not replace. If an environment variable with the specified name (ENVVAR parameter) exists at the specified level (LEVEL parameter), error message CPFA980 is issued.

***YES** Replace. If an environment variable with the specified name (ENVVAR parameter) exists at the specified level (LEVEL parameter), its value will be replaced by the new value.

Top

Examples

Example 1: Add an Environment Variable with CCSID 37

```
ADDENVVAR   ENVVAR(altdir)  VALUE('/mydir/dir2') CCSID(37)
```

This command adds the environment variable named altdir with the value /mydir/dir2 to the environment variables for the job. The value 37 is stored with the environment variable to indicate its CCSID.

Example 2: Set an Environment Variable to Null

```
ADDENVVAR   ENVVAR(LIBPATH)  VALUE(*NULL)
```

This command adds the environment variable named LIBPATH with the null (x'00) character value to the environment variables for the job.

Example 3: Add a System-level Environment Variable

```
ADDENVVAR   ENVVAR(homedir)  VALUE('/home')  LEVEL(*SYS)
```

This command adds a system-level environment variable named homedir with value /home.

Example 4: Reset a Job-level Environment Variable

```
ADDENVVAR   ENVVAR(altdir)  VALUE('/mydir/dir3')  REPLACE(*YES)
```

This command replaces the existing value of the variable altdir with the new value of /mydir/dir3.

Top

Error messages

***ESCAPE Messages**

CPFA980
Environment variable name exists.

CPFA982
ENVVAR character not valid.

CPFA983
Unexpected error occurred.

CPFA984
Maximum number of environment variables exist.

CPFA98E
*JOBCTL special authority required to update system-level environment variables.

CPF3BCA
CCSID &1 not supported.

Top

Add EWC Barcode Entry (ADDEWCBCDE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Extended Wireless Controller Bar Code Entry (ADDEWCBCDE) command adds a set of bar code group parameters to an extended wireless controller source file member. The bar code group defines the parameters for scanning a particular bar code label. The Portable Transaction Computer (PTC) group specifies the bar code groups that are used to configure the bar code scanner.

Restriction: If the values specified for the INZFILE and INZMBR parameters of this command do not match the values specified for the corresponding parameters of the wireless controller description, extended wireless controller configuration data will not be downloaded to the wireless adapter.

Restriction: To execute this command, the user profile requires *IOSYSCFG special authority.

Note: You can use the Change Controller Description (Local Work Station) (CHGCTLLWS) command to view or change values specified for the INZFILE and INZMBR parameters in the wireless controller description.

Top

Parameters

Keyword	Description	Choices	Notes
BCDGRP	Barcode group	*Name*	Required, Positional 1
INZMBR	Initialization source member	*Name*	Required, Positional 2
INZFILE	Initialization source file	*Qualified object name*	Optional, Positional 3
	Qualifier 1: Initialization source file	*Name*, **QEWCSRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
BCDTYPE	Barcode type	***UPC**, *EAN, *PLESSEY, *ALPHAPLESSEY, *ISBNPLESSEY, *PUREPLESSEY, *SAINPLESSEY, *UPCA, *UPCE, *EAN8, *EAN13, *CODABAR, *CODE3OF9, *CODE2OF5, *DISCR2OF5, *INTERL2OF5, *INDUST2OF5, *CODE11, *CODE128, *CODE93	Optional
LBLLEN	Label length	0-64, **0**	Optional
CHK1DIGIT	First check digit	***NO**, *YES	Optional
CHK2DIGIT	Second check digit	***NO**, *YES	Optional
ALLZERO	All zeros	***NO**, *YES	Optional
ALPHADSP	Alpha display	***NO**, *YES	Optional
ADDON2	Add on 2	***NO**, *YES	Optional
ADDON5	Add on 5	***NO**, *YES	Optional
SYS1UPCE	System 1 UPC-E	***NO**, *YES	Optional
SYS0UPCE	System 0 UPC-E	***NO**, *YES	Optional
UPCE	UPC-E	***NO**, *YES	Optional
EAN13	EAN 13	***NO**, *YES	Optional

Keyword	Description	Choices	Notes
EXT3OF9	Extended Character Set	***NO**, *YES	Optional
ADDON	Add on	***BIDIRECTIONAL**, *FORWARD	Optional
DROPBEGIN	Drop begin	0-64, **0**	Optional
DROPEND	Drop end	0-64, **0**	Optional
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Top

Barcode group (BCDGRP)

Specifies the bar code group name to be added. This name is used to identify configuration data related to a bar code group. If the bar code group name has been previously added, this command will fail. The bar code group name is a unique alphanumeric character string of as many as 16 characters in length.

Top

Initialization source member (INZMBR)

Specifies the extended wireless controller source file member to add the bar code entry to. The bar code configuration data is added to this member.

Top

Initialization source file (INZFILE)

Specifies the name of the source physical file that contains the extended wireless controller source file member. If the source physical file does not exist, this command will fail.

The name of the source file can be qualified by one of the following library values:

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name
Specify the name of the library to be searched.

The possible values are:

QEWCSRC
The source file name QEWCSRC is used.

source-file-name
Specify the name of the physical file that contains the source member.

Top

Barcode type (BCDTYPE)

Specifies the bar code type defined by this bar code group.

The possible values are:

***UPC** UPC bar codes are used.

***EAN** EAN bar codes are used.

***PLESSEY**
Plessey bar codes are used.

***ALPHAPLESSEY**
Alpha Plessey bar codes are used.

***ISBNPLESSEY**
ISBN Plessey bar codes are used.

***PUREPLESSEY**
Pure Plessey bar codes are used.

***SAINPLESSEY**
Sainsbury Plessey bar codes are used.

***UPCA**
UPC-A bar codes are used.

***UPCE**
UPC-E bar codes are used.

***EAN8**
EAN-8 bar codes are used.

***EAN13**
EAN-13 bar codes are used.

***CODABAR**
CODABAR bar codes are used.

***CODE3OF9**
CODE 3 of 9 bar codes are used.

***CODE2OF5**
CODE 2 of 5 bar codes are used.

***DISCR2OF5**
Discrete 2 of 5 bar codes are used.

***INTERL2OF5**
Interleaved 2 of 5 bar codes are used.

***INDUST2OF5**
Industrial 2 of 5 bar codes are used.

***CODE11**
CODE 11 bar codes are used.

***CODE128**
CODE 128 bar codes are used.

***CODE93**
CODE 93 bar codes are used.

Top

Label length (LBLLEN)

Specifies the label length of a bar code label for the specified bar code group.

The possible values are:

00 The label length is variable from 1 to 64.

label-length
Specify the character length of a bar code label for the bar code group. If the label length is 00 then label length is variable from 1 to 64. The valid range of values is from 0 through 64.

Top

First check digit (CHK1DIGIT)

Specifies whether the check digit or the first check digit are checked on the bar code label. This is valid only when *PLESSEY, *CODE3OF9, *EXTCODE3OF9, *CODE11, or *CODE2OF5 are specified by the BCDTYPE parameter.

The possible values are:

***NO** The check digit or the first check digit are ignored.

***YES** The check digit or the first check digit are checked for a valid read.

Top

Second check digit (CHK2DIGIT)

Specifies whether the second check digit is checked on the bar code label. This is valid only when *PLESSEY or *CODE11 are specified by the BCDTYPE parameter.

The possible values are:

***NO** The second check digit is ignored.

***YES** The second check digit is checked for a valid read.

Top

All zeros (ALLZERO)

Specifies whether a bar code label of all 0's is a valid scan. This is valid only when BCDTYPE(*PLESSEY) is specified.

The possible values are:

***NO** A bar code label of all 0's is not a valid scan.

***YES** A bar code label of all 0's is a valid scan.

Top

Alpha display (ALPHADSP)

Specifies whether to display bar code label characters : ; < = > and ? as the alphabetic characters A, B, C, D, E and F, respectively. This is valid only when BCDTYPE(*PLESSEY) is specified.

The possible values are:

***NO** Display the characters normally.

***YES** Display the characters as alphabetic characters.

Top

Add on 2 (ADDON2)

Specifies whether a 2-digit add on is valid or ignored. This is valid only when *UPC or *EAN are specified by the BCDTYPE parameter.

The possible values are:

***NO** A 2-digit add on is ignored.

***YES** A 2-digit add on is valid.

Top

Add on 5 (ADDON5)

Specifies whether a 5-digit add on is valid or ignored. This is valid only when *UPC or *EAN are specified by the BCDTYPE parameter.

The possible values are:

***NO** A 5-digit add on is ignored.

***YES** A 5-digit add on is valid.

Top

System 1 UPC-E (SYS1UPCE)

Specifies whether a System 1 UPC-E is valid or ignored. This is valid only when *UPC or *EAN are specified by the BCDTYPE parameter.

The possible values are:

***NO** A System 1 UPC-E is ignored.

***YES** A System 1 UPC-E is valid.

Top

System 0 UPC-E (SYS0UPCE)

Specifies whether a System 0 UPC-E is valid or ignored. This is valid only when *UPC or *EAN are specified by the BCDTYPE parameter.

The possible values are:

***NO** A System 0 UPC-E is ignored.

***YES** A System 0 UPC-E is valid.

Top

UPC-E (UPCE)

Specifies whether a UPC-E should be expanded to UPC-A. This is valid only when *UPC or *EAN are specified by the BCDTYPE parameter.

The possible values are:

***NO** UPC-E bar codes are unaffected.

***YES** UPC-E bar codes are expanded to UPC-A.

Top

EAN 13 (EAN13)

Specifies whether UPC and EAN bar codes are expanded to EAN-13. This is valid only when *UPC or *EAN are specified by the BCDTYPE parameter.

The possible values are:

***NO** UPC and EAN bar codes are unaffected.

***YES** UPC and EAN bar codes are expanded to EAN-13.

Top

Extended Character Set (EXT3OF9)

Specifies whether code 3 of 9 bar code uses the extended character set. This is valid only when *CODE3OF9 is specified by the BCDTYPE parameter.

The possible values are:

***NO** Code 3 of 9 bar code is unaffected.

***YES** Code 3 of 9 bar code uses the extended character set.

Top

Add on (ADDON)

Specifies the direction of add on digits. This is valid only when *UPC or *EAN are specified by the BCDTYPE parameter.

The possible values are:

***BIDIRECTIONAL**
Add on digits are valid in both directions.

***FORWARD**
Add on digits are valid only in the forward direction.

Top

Drop begin (DROPBEGIN)

Specifies the number of characters to drop from the beginning of the bar code label. The valid range of values is from 0 through 64.

The possible values are:

0 No characters are dropped from the beginning of the bar code label.

drop-begin
Specify the number of characters to drop from the beginning of the bar code label. The valid range of values is from 0 through 64.

Top

Drop end (DROPEND)

Specifies the number of characters to drop from the end of the bar code label. The valid range of values is from 0 through 64.

The possible values are:

0 No characters are dropped from the end of the bar code label.

drop-end
Specify the number of characters to drop from the end of the bar code label. The valid range of values is from 0 through 64.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the PTC entry.

The possible values are:

***BLANK**
Text is not specified.

'description'
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Examples

```
ADDEWCBCDE   BCDGRP(BCD01) INZMBR(EWC01)
             INZFILE(QGPL/QEWCSRC)  BCDTYPE(*UPC)
```

This command adds a bar code group named BCD01 with a bar code type of *UPC in a extended wireless controller configuration source file member named EWC01 in source physical file QEWCSRC in QGPL.

Top

Error messages

None

Top

Add Wireless Ctl Member (ADDEWCM)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Extended Wireless Controller Member (ADDEWCM) command adds a source file member with extended wireless controller parameters to the specified source file. This data is downloaded to the wireless controller when the controller is varied on. Specific Portable Transaction Computer (PTC) and bar code configurations are added to this member using the Add Extended Wireless Controller PTC Entry (ADDEWCPTCE) and Add Extended Wireless Controller Bar Code Entry (ADDEWCBCDE) commands.

Restriction: If the values specified for the INZFILE and INZMBR parameters of this command do not match the values specified for the corresponding parameters of the wireless controller description, extended wireless controller configuration data will not be downloaded to the wireless adapter.

Restriction: To execute this command, the user profile requires *IOSYSCFG special authority.

Note: You can use the Change Controller Description (Local Work Station) (CHGCTLLWS) command to view or change values specified for the INZFILE and INZMBR parameters in the wireless controller description.

Top

Parameters

Keyword	Description	Choices	Notes
INZMBR	Initialization source member	*Name*	Required, Positional 1
INZFILE	Initialization source file	*Qualified object name*	Optional, Positional 2
	Qualifier 1: Initialization source file	*Name*, **QEWCSRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TXPADR	Destination ID	4001-4FFE, **4001**	Optional
TXPPORT	Transport port	0-15, **0**	Optional
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional
KBDMAPMBR	Keyboard remap source member	*Name*, ***NONE**	Optional
SCNFMTMBR	Screen reformat rules member	*Name*, ***NONE**	Optional
KBDMAPFILE	Keyboard remapping source file	*Qualified object name*	Optional
	Qualifier 1: Keyboard remapping source file	*Name*, **QEWCKBDMAP**	
	Qualifier 2: Library	*Name*, **QGPL**	
SCNFMTFILE	Screen reformatting rules file	*Qualified object name*	Optional
	Qualifier 1: Screen reformatting rules file	*Name*, **QEWCSCNFMT**	
	Qualifier 2: Library	*Name*, **QGPL**	

Top

Initialization source member (INZMBR)

Specifies the name of the source file member containing the extended wireless controller configuration data that is added to the source file.

Top

Initialization source file (INZFILE)

Specifies the name of a source physical file to contain extended configuration source file member. If the source physical file does not exist, this command will fail.

The name of the source file can be qualified by one of the following library values:

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name
Specify the name of the library to be searched.

The possible values are:

QEWCSRC
The source file name QEWCSRC is used.

source-file-name
Specify the name of an existing source physical file to which the member is added.

Top

Destination ID (TXPADR)

Specifies the local destination ID (transport address) that the PTC attempts to connect to at emulation startup time. This is a 4-byte hexadecimal number with valid values ranging from 4001 through 4FFE. The destination ID must match the destination ID used in the PTC(s).

The possible values are:

destination ID (transport-address)
Specify the local destination ID (transport address).

Top

Transport port (TXPPORT)

Specifies the local transport port connection number of the controller. The valid range is from 0 through 15. The value used by the PTC(s) is 0.

The possible values are:

0 The value 0 is used.

transport-port
Specify the local transport port connection number.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the program and its function. More information on this parameter is in the CL Reference book, Appendix A.

The possible values are:

***BLANK**
Text is not specified.

'description'
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Keyboard remap source member (KBDMAPMBR)

Specifies the source physical file member that contains the Keyboard Map information.

NOTE: The Keyboard Mapping function is not supported on the 2668 Feature Number.

The possible values are:

***NONE**
Keyboard Mapping file member is not used.

source-file-member
The name of the file member that specifies the Keyboard Map information.

Top

Screen reformat rules member (SCNFMTMBR)

Specifies the source physical file member that contains the Screen Reformatter information

NOTE: The Screen Reformatter function is not supported on the 2668 Feature Number.

The possible values are:

***NONE**
Screen Reformatter file member is not used.

source-file-member
The name of the file member that specifies the Screen Reformatter information

Top

Keyboard remapping source file (KBDMAPFILE)

Specifies the name of a source physical file to contain Keyboard Map source file member. If the source physical file does not exist, this command will fail.

The name of the source file can be qualified by one of the following library values:

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name
Specify the name of the library to be searched.

The possible values are:

QEWCKBDMAP
The source file name QEWCKBDMAP is used.

source-file-name
Specify the name of an existing source physical file to which the member is added.

Top

Screen reformatting rules file (SCNFMTFILE)

Specifies the name of a source physical file to contain Screen Reformatter source file member. If the source physical file does not exist, this command will fail.

The name of the source file can be qualified by one of the following library values:

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name
Specify the name of the library to be searched.

The possible values are:

QEWCSCNFMT
The source file name QEWCSCNFMT is used.

source-file-name
Specify the name of an existing source physical file to which the member is added.

Top

Examples

```
ADDEWCM   INZMBR(EWC01)  INZFILE(*LIBL/QEWCSRC)
```

This command adds a member named EWC01 in the default source physical file QEWCSRC in the library list with the default TXPADR of 4001 and default TXPPORT of 0.

Top

Error messages

None

Top

Add EWC PTC Entry (ADDEWCPTCE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Extended Wireless Controller PTC Entry (ADDEWCPTCE) command adds a set of Portable Transaction Computer (PTC) group parameters to an extended wireless controller source file member. The PTC group parameters are the configurable PTC 5250 emulation operating parameters. These parameters are sent to each configured PTC at emulation startup. The Add Extended Wireless Controller Member (ADDEWCM) command must be run before this command to create the source file member.

Restriction If the values specified for the INZFILE and INZMBR parameters of this command do not match the values specified for the corresponding parameters of the wireless controller description, extended wireless controller configuration data will not be downloaded to the wireless adapter.

Restriction: To execute this command, the user profile requires *IOSYSCFG special authority.

Note: You can use the Change Controller Description (Local Work Station) (CHGCTLLWS) command to view or change values specified for the INZFILE and INZMBR parameters in the wireless controller description.

Top

Parameters

Keyword	Description	Choices	Notes
PTCGRP	PTC group	*Name*	Required, Positional 1
INZMBR	Initialization source member	*Name*	Required, Positional 2
INZFILE	Initialization source file	*Qualified object name*	Optional, Positional 3
	Qualifier 1: Initialization source file	*Name*, **QEWCSRC**	
	Qualifier 2: Library	*Name*, ***LIBL**	
PTCRANGE	PTC ID range	*Element list*	Optional
	Element 1: Begin ID	0001-1022, **0001**	
	Element 2: End ID	0001-1022, **1022**	
INTENSITY	Intensity	***NORMAL**, *INVERSE	Optional
STSLINE	Status line	***YES**, *NO	Optional
CSRTYPE	Cursor type	***UNDERLINE**, *BLOCK	Optional
INACTTMR	Inactivity timer	0-9999, ***DEV**	Optional
BCKLTTMR	Backlight timer	0-9999, ***DEV**	Optional
BCKLTKEY	Backlight key	***ON**, *OFF	Optional
BYPASSEXIT	Bypass exit	***YES**, *NO	Optional
AUTORUN	Automatic run	***YES**, *NO	Optional
PRINTER	Printer	***SYSTEM**, *PTC	Optional
WANDTYPE	Wand type	***NONE**, *PENCIL, *LASER, *RS232	Optional
PECKRATE	Wand pecking rate	***DEV**, 2, 4, 8, 16, 32, 48	Optional

Keyword	Description	Choices	Notes
LASERTMR	Laser read timer	***DEV**, 1440, 2880, 4320, 5760	Optional
BCDFKEY	Barcode function keys	***OFF**, *ON	Optional
AUTOENTER	Auto Enter	***OFF**, *ON	Optional
CSRLOC	Cursor location	***HOLD**, *FIRST	Optional
SHORTSCAN	Short scan	***YES**, *NO	Optional
SCANEOF	Scan end of file	***YES**, *NO	Optional
POLL	Fast poll interval	0-9999, ***DEV**	Optional
POLLDLY	Fast poll delay	0-9999, ***DEV**	Optional
POLLDECAY	Fast poll decay	0-255, ***DEV**	Optional
SLOWPOLL	Slow poll interval	0-99999, ***DEV**	Optional
DESTHOP	Destination hop	Single values: ***NONE** Other values (up to 8 repetitions): *Element list*	Optional
	Element 1: Destination ID	4001-4FFE	
	Element 2: Frequency	**1**, 2, 3, 4, 5, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911	
	Element 3: Data rate	*NONE, 1M, **2M**	
	Element 4: Radio system identifier	000002-FFFFFE, **000002**	
BCDGRP	Barcode group	Single values: ***NONE** Other values (up to 6 repetitions): *Name*	Optional
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional
ENBKBDMAP	Enable keyboard remapping	***NO**, *YES	Optional
ENBSCNFMT	Enable screen reformatting	***NO**, *YES	Optional
DEVPFX	Device name prefix	*Simple name*, **QWLS**	Optional

Top

PTC group (PTCGRP)

Specifies the PTC group name to be added. This name is used to identify configuration data related to a group of PTCs bound by the PTCRANGE parameter. The PTC group name is a unique alphanumeric character string with a maximum of 16 characters in length.

Top

Initialization source member (INZMBR)

Specifies the extended wireless controller source file member in which the PTC entry is added. The PTC configuration data is added to this member.

Top

Initialization source file (INZFILE)

Specifies the name of the source physical file that contains the extended wireless controller source file member. If the source physical file does not exist, this command will fail.

The name of the source file can be qualified by one of the following library values:

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name
Specify the name of the library to be searched.

The possible values are:

QEWCSRC
The source file name QEWCSRC is used.

source-file-name
Specify the name of the source physical file.

Top

PTC ID range (PTCRANGE)

Specifies the beginning and ending 4-character decimal PTC ID to use for this PTC group. The defined configuration data is downloaded to any PTC detected within this range. A PTC group of a single PTC can be specified by setting the beginning and ending PTC ID to the same value.

The possible Beginning ID values are:

0001 The value 0001 is used as the beginning PTC ID of the PTC group.

begin-ID
Specify the beginning PTC ID of the PTC group. The valid values range from 0001 through 1022.

The possible Ending ID values are:

1022 The value 1022 is used as the ending PTC ID of the PTC group.

end-ID
Specify the ending PTC ID of the PTC group. The valid values range from 0001 through 1022.

Top

Intensity (INTENSITY)

Specifies how the emulation screen on the PTC handles a field with the intensity attribute set.

The possible values are:

***NORMAL**
The field is displayed as normal.

***INVERSE**
The field is displayed as reverse image.

Top

Status line (STSLINE)

Specifies whether the status line is displayed on the PTC.

The possible values are:

***YES** The status line is displayed.

***NO** The status line is not displayed.

Top

Cursor type (CSRTYPE)

Specifies the type of cursor for use on the PTC.

The possible values are:

***UNDERLINE**
The cursor is an underline cursor.

***BLOCK**
The cursor is a block cursor.

Top

Inactivity timer (INACTTMR)

Specifies, in seconds, the inactivity timer value for the PTC. The PTC will power down if no activity occurs on the PTC during this time period.

The possible values are:

***DEV** The default device inactivity timer value is used.

inactivity-timer
Specify the inactivity timer value in seconds. The valid range of values is from 0 through 9999.

Top

Backlight timer (BCKLTTMR)

Specifies, in seconds, the backlight inactivity timer value for the PTC. The PTC turns the backlight off if no activity occurs on the PTC during this time period.

The possible values are:

***DEV** The default device backlight inactivity timer value is used.

backlight-timer
Specify the backlight inactivity timer value in seconds. The valid range of values is from 0 through 9999.

Top

Backlight key (BCKLTKEY)

Specifies whether the backlight turns on when a key is pressed on the PTC.

The possible values are:

***ON** The backlight turns on when a key is pressed.

***OFF** The backlight does not turn on when a key is pressed

Top

Bypass exit (BYPASSEXIT)

Specifies whether to bypass exit processing when leaving emulation on the PTC.

The possible values are:

***NO** Exit processing runs.

***YES** Exit processing does not run.

Top

Automatic run (AUTORUN)

Specifies whether emulation software automatically runs on the PTC at system IPL.

The possible values are:

***NO** Emulation software does not automatically run at system IPL.

***YES** Emulation software automatically runs at system IPL.

Top

Printer (PRINTER)

Specifies whether the printer for the PTC is the system printer or a printer that is locally attached to the PTC.

The possible values are:

***SYSTEM**
The system printer is used.

***PTC** The printer that is locally attached to the PTC is used.

Top

Wand type (WANDTYPE)

Specifies the type of wand scanner being used.

The possible values are:

***NONE**
No wand scanner is used.

***PENCIL**
A pencil wand scanner is used.

***LASER**
A laser wand scanner is used.

***RS232**
The wand scanner is attached on the RS-232 connector on the PTC.

Top

Wand pecking rate (PECKRATE)

Specifies the wand pecking rate, in milliseconds. This value sets the time interval between the wand power on and power off states that is used to detect whether a label is present.

The possible values are:

***DEV** The default device wand pecking rate is used.

peck-rate
Specify the wand peck rate in milliseconds. Valid values are 2, 4, 8, 16, 32, and 48.

Top

Laser read timer (LASERTMR)

Specifies the laser read timer value, in milliseconds. If a good scan has not been performed before the given timer value, then the laser is turned off.

The possible values are:

***DEV** The default device laser read timer value is used.

laser-read-timer
Specify the laser read timer value in milliseconds. Valid values are 1440,2880,4320, and 5760.

Top

Barcode function keys (BCDFKEY)

Specifies whether function keys are entered by bar code labels.

The possible values are:

***NO** Function key entry by bar code is disabled.

***YES** Function key entry by bar code is enabled.

Top

Auto Enter (AUTOENTER)

Specifies whether the PTC Auto Enter function is on or off.

The possible values are:

***OFF** The Auto Enter function is disabled.

***ON** The Auto Enter function is enabled.

Top

Cursor location (CSRLOC)

Specifies when the cursor is moved from one window chunk to another window chunk, if the cursor defaults to the first input field in the chunk or hold its relative position within the window. A chunk is defined as being a portion of the 5250 emulation screen equal to the size of the PTC display.

The possible values are:

***HOLD**
Cursor holds its position when moving from one window chunk to another.

***FIRST**
Cursor moves to the first active field when moving from one window chunk to another.

Top

Short scan (SHORTSCAN)

Specifies whether a bar code label that does not completely fill an input field is processed as if it has filled that field.

The possible values are:

***YES** Bar code labels that do not fill an input field are processed as if they have filled the field.

***NO** Bar code labels must fill the input field before they are processed.

Top

Scan end of file (SCANEOF)

Specifies whether an erase end of field is done when a bar code label is shorter than the input field.

The possible values are:

***YES** An erase to end of field is done after a bar code scan.

***NO** An erase to end of field is not done after a bar code scan.

Top

Fast poll interval (POLL)

Specifies, in milliseconds, the fast poll interval for the radio module on the PTC.

The possible values are:

***DEV** The default device fast poll interval value is used.

poll-interval
Specify the fast poll interval in milliseconds. The valid range of values is from 0 through 9999.

Top

Fast poll delay (POLLDLY)

Specifies, in milliseconds, the fast poll delay parameter for the PTC radio module.

The possible values are:

***DEV** The default device fast poll delay value is used.

poll-delay
Specify the fast poll delay value in milliseconds. The valid range of values is from 0 through 9999.

Top

Fast poll decay (POLLDECAY)

Specifies the fast poll decay for the PTC radio module. The valid range of values is from 0 through 255.

The possible values are:

***DEV** The default device fast poll decay value is used.

poll-decay
Specify the fast poll decay value. The valid range of values is from 0 through 255.

Top

Slow poll interval (SLOWPOLL)

Specifies, in milliseconds, the slow poll interval for radio module on the PTC.

The possible values are:

***DEV** The default device slow poll interval value is used.

slow-poll
Specify the slow poll interval in milliseconds. The valid range of values is from 0 through 99999.

Top

Destination hop (DESTHOP)

Specifies the alternate destination ID (transport address) that the PTC attempts to connect to at emulation startup time. This is a four-element field and up to 8 destination hops can be specified.

The possible values are:

***NONE**
No destination hops are identified.

The possible Destination ID (Tranport Address) values are:

destination ID
Specify the 4-byte hexadecimal destination ID of a wireless controller. The valid range of values is from 4001 through 4FFE.

Specifies which center frequency to use on the radio based on the radio channel set. The radio channel set is determined by the radio country code.

Valid values are from 1 to 5 for PTC's operating in the 2.4 Gigahertz range and 900 to 911 for PTC's operating in the 900 Megahertz range. If a value in the 900 to 911 range is used, the data rate may not be specified (900 Megahertz data rates are fixed by the frequency channel number).

The possible Frequency values for the 2.4 Gigahertz range are:

For channel set 9, there are five center frequencies.

1 A center frequency of 2.412 gigahertz is used.

2 A center frequency of 2.427 gigahertz is used.

3 A center frequency of 2.442 gigahertz is used.

4 A center frequency of 2.457 gigahertz is used.

5 A center frequency of 2.465 gigahertz is used.

For channel set 10, there are five center frequencies. Channel set 10 is used in ETSI (European Telecommunication and Standards Institute) countries.

1 A center frequency of 2.412 gigahertz is used.

2 A center frequency of 2.427 gigahertz is used.

3 A center frequency of 2.442 gigahertz is used.

4 A center frequency of 2.457 gigahertz is used.

5 A center frequency of 2.472 gigahertz is used.

For channel set 11, there is one center frequency.

1 A center frequency of 2.484 gigahertz is used.

For channel set 12, there are three center frequencies.

1 A center frequency of 2.457 gigahertz is used.

2 A center frequency of 2.465 gigahertz is used.

3 A center frequency of 2.472 gigahertz is used.

For channel set 13, there are three center frequencies.

1 A center frequency of 2.411 gigahertz is used.

2 A center frequency of 2.425 gigahertz is used.

3 A center frequency of 2.439 gigahertz is used.

The possible Frequency values for the 900 Megahertz range are: 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, and 911.

The possible Datarate values are:

Specifies the wireless LAN data rate.

2M A wireless LAN data rate of 2 megabits per second is used.

1M A wireless LAN data rate of 1 megabit per second is used.

The possible Sys-ID values are:

Specifies the 6-character hexadecimal radio system identifier to be used.

000002 The value 000002 is used.

sys-ID Specify a system ID to be used. Valid values range from 000002 to FFFFFE in hexadecimal format and the last digit must be even (for example, 0, 2, 4, 6, 8, A, C, E).

Top

Barcode group (BCDGRP)

Specifies the bar code group names used to define the bar code scanning capability of the PTC group. Bar code group names are defined and modified by the ADDEWCBCDE and CHGEWCBCDE commands.

The possible values are:

***NONE**
No bar code group names are defined for this PTC group.

bar-code-group
Specify the bar code group name that corresponds to the bar code scanning capabilities required by the PTC. A maximum of 6 can be specified.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the PTC entry.

The possible values are:

***BLANK**
Text is not specified.

'description'
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Enable keyboard remapping (ENBKBDMAP)

Enables or disables Keyboard Mapping function.

NOTE: The Keyboard Mapping function is not supported on the 2668 Feature Number.

The possible values are:

***NO** The Keyboard Mapping function is disabled.

****YES*** The Keyboard Mapping function is enabled.

Top

Enable screen reformatting (ENBSCNFMT)

Enables or disables Screen Reformatter function.

NOTE: The Screen Reformatter function is not supported on the 2668 Feature Number.

The possible values are:

***NO** The Screen Reformatter function is disabled.

***YES** The Screen Reformatter function is enabled.

Top

Device name prefix (DEVPFX)

Specifies a PTC Device Name Prefix. The is a 6 character alpha numeric string that can be used to uniquely identify PTC displays. It will be used in the following format: PPPPPPXXXX. PPPPPP is the 6 character prefix and XXXX is the PTC ID Range.

Top

Examples

```
ADDEWCPTCE   PTCGRP(PTC01)  INZMBR(EWC01)  PTCRANGE(0001 0020)
```

This command adds the configuration parameters for the PTC for a PTC group named PTC01 for PTCs with addresses from 1 to 20 to the extended wireless controller source file member EWC01 in source physical file QEWCSRC in the library list.

Top

Error messages

None

Top

Add Wireless Line Member (ADDEWLM)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Extended Wireless Line Member (ADDEWLM) command adds a source file member that contains extended wireless line parameters to the specified source file. This data is downloaded to the wireless local area network (LAN) adapter when the line is varied on.

Restriction: If the values specified for the INZFILE and INZMBR parameters of this command do not match the values specified for the corresponding parameters of the wireless line description, extended wireless line configuration data will not be downloaded to the wireless adapter.

Restriction: To execute this command, the user profile requires *IOSYSCFG special authority.

Note: You can use the Change Line Description (Wireless) (CHGLINWLS) command to view or change values specified for the INZFILE and INZMBR parameters in the wireless line description.

Top

Parameters

Keyword	Description	Choices	Notes
INZMBR	Initialization source member	*Name*	Required, Positional 1
INZFILE	Initialization source file	*Qualified object name*	Optional, Positional 2
	Qualifier 1: Initialization source file	*Name*, **QEWLSRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
ADPTCFG	Adapter configuration	***ALL**, *RADIO, *WIRED	Optional
HOPID	Hop identifier	020000000000-FEFFFFFFFFFF, ***ADPT**	Optional
ROOT	Root cell	***YES**, *NO	Optional
FREQUENCY	Frequency	**1**, 2, 3, 4, 5	Optional
DATARATE	Data rate	**2M**, 1M	Optional
SYSID	Radio system identifier	000002-FFFFFE, **000002**	Optional
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Top

Initialization source member (INZMBR)

Specifies the name of the new source file member that contains the extended wireless controller configuration data.

Top

Initialization source file (INZFILE)

Specifies the name of the existing source physical file to contain extended configuration source file member. If the source physical file does not exist, this command will fail.

The name of the source file can be qualified by one of the following library values:

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name
Specify the name of the library to be searched.

The possible values are:

QEWLSRC
The source file name QEWLSRC is used.

source-file-name
Specify the name of an existing source physical file to add to the member.

Top

Adapter configuration (ADPTCFG)

Specifies the wireless LAN adapter configuration. The wireless LAN adapter has two communication methods: radio and wire backbone. The radio is a direct-sequence spread spectrum radio that can be used for wireless communications. The wire backbone is used to connect access points to a wireless LAN adapter. These access points also have a direct-sequence spread spectrum radio and are used to extend the radio coverage of the wireless network.

The possible values are:

***ALL** The wireless LAN adapter uses both radio and wire backbone communications.

***RADIO**
The wireless LAN adapter uses only radio communications.

***WIRED**
The wireless LAN adapter uses only wire backbone communications.

Top

Hop identifier (HOPID)

Specifies the 12-character hexadecimal radio identifier on the wireless LAN adapter. This is an internal identifier that is used to determine the destination of a data packet, during its next hop on the network.

Note: The value specified for the HOPID parameter is one of two different 12-character hexadecimal identifiers used by a wireless LAN adapter. The other is an endpoint identifier that is equivalent to the identifier used in Ethernet or token-ring networks.

The possible values are:

***ADPT**
The preset wireless input/output adapter (IOA) address is used.

hop-ID

Specify the wireless LAN adapter hop address that overrides the preset address. The hop address must be an individual address (it cannot be a group address). Valid values range from 020000000000 to FEFFFFFFFFFF in hexadecimal format. The second digit from the left of the address must be a 2, 6, A, or E.

Top

Root cell (ROOT)

Specifies whether the radio of the wireless LAN adapter is a root cell.

A wireless network consists of a group of wireless access points that are interconnected in the form of a logical spanning tree. One of these wireless access points must be designated as the root cell for the network.

The possible values are:

***YES** The radio on the wireless LAN adapter is a root cell.

***NO** The radio on the wireless LAN adapter is not a root cell.

Top

Frequency (FREQUENCY)

Specifies which center frequency to use on the radio based on the radio channel set. The radio channel set is determined by the radio country code.

The possible Frequency values are:

For channel set 9, there are five center frequencies.

1 A center frequency of 2.412 gigahertz is used.

2 A center frequency of 2.427 gigahertz is used.

3 A center frequency of 2.442 gigahertz is used.

4 A center frequency of 2.457 gigahertz is used.

5 A center frequency of 2.465 gigahertz is used.

For channel set 10, there are five center frequencies. Channel set 10 is used in ETSI (European Telecommunication and Standards Institute) countries.

1 A center frequency of 2.412 gigahertz is used.

2 A center frequency of 2.427 gigahertz is used.

3 A center frequency of 2.442 gigahertz is used.

4 A center frequency of 2.457 gigahertz is used.

5 A center frequency of 2.472 gigahertz is used.

For channel set 11, there is one center frequency.

1 A center frequency of 2.484 gigahertz is used.

For channel set 12, there are three center frequencies.

1 A center frequency of 2.457 gigahertz is used.

2 A center frequency of 2.465 gigahertz is used.

3 A center frequency of 2.472 gigahertz is used.

For channel set 13, there are three center frequencies.

1 A center frequency of 2.411 gigahertz is used.

2 A center frequency of 2.425 gigahertz is used.

3 A center frequency of 2.439 gigahertz is used.

Top

Data rate (DATARATE)

Specifies the wireless LAN data rate.

The possible values are:

2M A wireless LAN data rate of 2 megabits per second is used.

1M A wireless LAN data rate of 1 megabit per second is used.

Top

Radio system identifier (SYSID)

Specifies the 6-character hexadecimal radio system identifier to be used.

The possible values are:

000002 The value 000002 is used.

sys-ID Specify a radio system identifier. Valid values range from 000002 to FFFFFE in hexadecimal format, but the last digit must be even (i.e. 0,2,4,6,8,A,C,E).

Top

Text 'description' (TEXT)

Specifies text that briefly describes the program and its function. More information on this parameter is in the CL Reference book, Appendix A.

The possible values are:

***BLANK**
Text is not specified.

'description'
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Examples

```
ADDEWLM   INZMBR(EWL01)
```

This command adds a member named EWL01 in the default source physical file QEWLSRC in the library list.

Top

Error messages

None

Top

Add Exit Program (ADDEXITPGM)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The Add Exit Program (ADDEXITPGM) command adds an exit program entry for a specific exit point. Each exit point can have a single entry or multiple entries. The exit program number indicates the sequence in which the exit programs are run.

Top

Parameters

Keyword	Description	Choices	Notes
EXITPNT	Exit point	*Simple name*	Required, Positional 1
FORMAT	Exit point format	*Simple name*	Required, Positional 2
PGMNBR	Program number	1-2147483647, *LOW, *HIGH	Required, Positional 3
PGM	Program	*Qualified object name*	Required, Positional 4
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, *CURLIB	
THDSAFE	Threadsafe	***UNKNOWN**, *NO, *YES	Optional
MLTTHDACN	Multithreaded job action	***SYSVAL**, *RUN, *MSG, *NORUN	Optional
TEXT	Text 'description'	*Character value*, ***BLANK**, *MSGID	Optional
MSGID	Message identifier	*Simple name*	Optional
MSGF	Message file	*Qualified object name*	Optional
	Qualifier 1: Message file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**	
REPLACE	Replace existing entry	*YES, ***NO**	Optional
CRTEXITPNT	Create exit point	*YES, ***NO**	Optional
PGMDTA	Exit program data	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Coded character set ID	*Integer*, ***JOB**	
	Element 2: Length of data	0-2048, ***CALC**	
	Element 3: Program data	*Character value*, **X''**	

Top

Exit point (EXITPNT)

Specifies the exit point name to which the exit program is added. If no exit point by this name exists, and CRTEXITPNT(*YES) is specified, an exit point is created.

Top

Exit point format (FORMAT)

Specifies the exit point format name of the exit program that is added.

Top

Program number (PGMNBR)

Specifies the sequence in which the exit programs are run when multiple exit point programs for a specific exit point are defined.

The possible values are:

***LOW** The lowest available number for that specific exit point is assigned.

***HIGH**
The highest available number for that specific exit point is assigned.

program-number
Specify the exit program sequence number. Valid values range from 1 through 2,147,483,647. Processing sequence is from the lowest number to the highest number. Exit program numbers do not need to be consecutive.

Top

Program (PGM)

Specifies the name of exit program to be called. The program does not have to exist on the system when this command is run.

The name of the exit program can be qualified by one of the following library values:

***CURLIB**
The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is used.

library-name
Specify the name of the library where the exit program is located.

The possible value is:

program-name
Specify the name of the exit program.

Top

Threadsafe (THDSAFE)

Specify the threadsafe attribute for the exit program entry. If you do not know the threadsafety status of the exit program entry, use the default value of *UNKNOWN.

The possible values are:

***UNKNOWN**
The threadsafety status of this exit program entry is not known.

***NO** The exit program entry is not threadsafe.

***YES** The exit program entry is threadsafe.

Top

Multithreaded job action (MLTTHDACN)

Specify the multithreaded job action for this exit program entry. If you do not know the action to take in a multithreaded job, use the default value of *SYSVAL.

The possible values are:

***SYSVAL**
The multithreaded job action specified in the QMLTTHDACN system value is used.

***RUN** Run the exit program entry.

***MSG** Run the exit program entry and send an informational message.

***NORUN**
Do not run the exit program entry.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the exit program.

The possible values are:

***BLANK**
No text is specified.

***MSGID**
The description is taken from the message specified by the MSGID and MSGF parameters. The description is retrieved when exit program information is displayed using the WRKREGINF (Work with Registration Information) command or retrieved using the QusRetrieveExitInformation API.

'description'
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Message identifier (MSGID)

Specifies the message identifier that contains the text that describes the exit program. The message is retrieved from the message file specified by the MSGF parameter. This parameter can only be specified if TEXT(*MSGID) is specified.

The possible values are:

message-identifier
Specify the seven-character message identifier of the message that describes the exit program.

Top

Message file (MSGF)

Specifies the message file and library that contains the message specified by the MSGID parameter. This parameter can only be specified if TEXT(*MSGID) is specified. The name of the message file is qualified by one of the following library values:

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

library-name
Specify the name of the library where the message file is located.

The possible values are:

message-file
Specify the name of the message file.

Top

Replace existing entry (REPLACE)

Specify whether the attributes and data for an exit program entry are replaced. New values for the program data and the text or message identifier can be specified. The program name and library cannot be changed.

The possible values are:

***NO** Do not replace the attributes and data for an exit program.

***YES** Replace the attributes and data for and exit program entry.

Top

Create exit point (CRTEXITPNT)

Specify whether the exit point is automatically created if it does not already exist.

The possible values are:

***NO** Do not create the exit point. If the specified exit point does not exist the exit program is not added and an error message is returned.

***YES** Create the specified exit point.

Top

Exit program data (PGMDTA)

Specifies the data that is passed to the exit program. This data should correspond to the input data defined by the exit point provider.

The possible values are:

***NONE**
No data is passed to the exit program.

The possible CCSID for program data values are:

***JOB** The CCSID (coded character set identifier) of the current job is used.

CCSID-for-data
Specify the CCSID associated with the data passed to the exit program.

The possible Length of program data values are:

***CALC**
The length is determined by the number of bytes specified for the third element of this parameter.

length-of-data
Specify the number of bytes of data passed to the exit program.

The possible Program data value is:

program-data
Specify the character or hexadecimal program data to be passed to the exit program. If you specify more program data than the length specified, the program data passed to the exit program is truncated. If you specify less program data than the length specified, the program data passed to the exit program is padded on the right with blanks. You can specify up to 2048 bytes of program data.

Top

Examples

```
ADDEXITPGM   EXITPNT(USER_EXIT_ONE)  FORMAT(EXIT1)  PGMNBR(1)
             PGM(LIB2/MYPGM)  TEXT(*MSGID)  MSGID(TXT2345)
             MSGF(LIB1/MYMSGF)
```

This command adds exit program MYPGM in library LIB2 to exit point USER_EXIT_ONE. This is first exit program run for the exit point. The text description for the exit program is retrieved from message TXT2345 in message file MYMSGF in library LIB1. No program data is passed to the exit program.

Top

Error messages

***ESCAPE Messages**

CPF0001
Error found on &1 command.

Top

Add Font Table Entry (ADDFNTTBLE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Font Table Entry (ADDFNTTBLE) command adds an entry in the specified font mapping table. This command adds an entry in the user font mapping tables used by Print Services Facility (PSF) that controls:

- Host-resident to printer-resident font character set mapping
- Printer-resident to host-resident font character set mapping
- Host-resident to printer-resident code page mapping
- Printer-resident to host-resident code page mapping
- Printer-resident to printer-resident font substitution mapping

To override a mapping entry in the system font or code page mapping tables, add the new mapping entry to a user table. Mapping entries added to a user table override any corresponding entries in the system tables.

When performing the printer to host and host to printer font mapping (first four tables above), PSF first searches the user tables for a match. If no match is found in the user mapping tables, PSF searches the appropriate system tables.

PSF selects printer-resident fonts from the font mapping tables in the following order:

1. If the printer-resident font specified in the print job is supported by the printer, then it is used. The printer-resident to printer-resident font substitution table is not searched.
2. If the printer-resident font specified in the print job is not supported by the printer, then the printer-resident to printer-resident font substitution table is searched.
   a. If a matching entry is found in the printer-resident font substitution table and the entry is supported by the printer, then the specified substitute font in the printer-resident font substitution table is used.
   b. If a matching entry is not found in the printer-resident font substitution table or if the specified substitute font is not supported by the printer, then the system will use its internal font substitution tables to perform the font substitution.

Refer to Printer Device Programming, SC41-5713 for more information on font mapping tables.

Restrictions:

- The PSF feature is required to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
FNTTBL	Font table	Single values: *PHFCS, *HPFCS, *PHCP, *HPCP Other values: *Qualified object name*	Required, Positional 1
	Qualifier 1: Font table	*Name*	
	Qualifier 2: Library	*Name*, *CURLIB, ***LIBL**	

Keyword	Description	Choices	Notes
PHFCS	Printer to host font	*Element list*	Optional
	Element 1: Printer font	*Element list*	
	Element 1: Identifier	1-65535	
	Element 2: Width	1-32767, *NONE, *PTSIZE	
	Element 3: Attributes	***NONE**, *BOLD, *ITALIC, *BOLDITC, *DBLWIDE, *ITCDBLWIDE	
	Element 4: Graphic character set	*Integer*, ***SYSVAL**	
	Element 5: Point size	1.0-999.9, ***WIDTH**, *NONE	
	Element 2: Host font	*Element list*	
	Element 1: Font character set	*Name*	
	Element 2: Type	***RASTER**, *OUTLINE	
HPFCS	Host to printer font	*Element list*	Optional
	Element 1: Host font	*Element list*	
	Element 1: Font character set	*Name*	
	Element 2: Type	***RASTER**, *OUTLINE	
	Element 2: Printer font	*Element list*	
	Element 1: Identifier	1-65535, *NONE	
	Element 2: Width	1-32767, *NONE, *PTSIZE	
	Element 3: Attributes	***NONE**, *BOLD, *ITALIC, *BOLDITC, *DBLWIDE, *ITCDBLWIDE	
	Element 4: Graphic character set	*Integer*, ***SYSVAL**	
	Element 5: Point size	1.0-999.9, ***WIDTH**, *NONE	
PHCP	Printer to host code page	*Element list*	Optional
	Element 1: Printer code page	*Element list*	
	Element 1: Graphic character set	*Integer*, *SYSVAL	
	Element 2: Code page	*Integer*	
	Element 2: Host code page	*Element list*	
	Element 1: Code page	*Name*	
HPCP	Host to printer code page	*Element list*	Optional
	Element 1: Host code page	*Element list*	
	Element 1: Code page	*Name*	
	Element 2: Printer code page	*Element list*	
	Element 1: Graphic character set	*Integer*, *SYSVAL	
	Element 2: Code page	*Integer*	
PPFCS	Printer to printer font	*Element list*	Optional
	Element 1: From printer font	*Element list*	
	Element 1: Identifier	1-65535	
	Element 2: Point size	1.0-999.9, *ALL, ***NONE**	
	Element 2: To printer font	*Element list*	
	Element 1: Identifier	1-65535	
	Element 2: Point size	1.0-999.9, *ALL, ***NONE**	

Top

Font table (FNTTBL)

Specifies the font mapping table entry to be added.

Single values

***PHFCS**
Add an entry to the printer-resident to host-resident font character set mapping table.

This table would be used when your application, such as DDS, references printer-resident fonts and the printer does not support resident fonts, for example, an IBM 3827,3825, 3820, or 3900 Model 1. Print Services Facility (PSF) must map the references from printer-resident fonts to host-resident fonts and download them.

***PHCP**
Add an entry to the printer-resident to host-resident code page mapping table.

This table would be used when your application references printer-resident code pages and the printer being used does not support printer-resident code pages. The printer-resident code page must be mapped to a host-resident code page and downloaded to the printer by PSF.

***HPFCS**
Add an entry to the host-resident to printer-resident font character set mapping table.

This table would be used when your application references host-resident fonts (font character sets and code pages) and the printer, such as the 4224, 4234, 4230, and 64XX, does not support downloading of host-resident fonts. PSF must map the references from host-resident fonts to printer-resident fonts.

***HPCP**
Add an entry to the host-resident to printer-resident code page mapping table.

This table is similar to the QHPFCS table, in that it is used when the application references host-resident code pages and the printer being used does not support host-resident code pages. The host-resident code page must be mapped to a printer-resident code page and downloaded to the printer by PSF.

Qualifier 1: Font table

name Specify the name of the font table to be changed. You must specify a name in order to change a printer-resident to printer-resident font mapping table. You should use a printer-resident font mapping table when all three of the following conditions exist:

1. You are printing to a PSF attached printer
2. Your application specifies a printer-resident font which is not supported by the printer you are using.
3. You want to specify a different substitute printer-resident font than the one selected by the system.

To use a printer-resident to printer-resident font mapping table with a particular PSF printer, you need to specify the name of the font table on the FNTTBL parameter of the Create PSF Configuration (CRTPSFCFG) or Change PSF Configuration (CHGPSFCFG) command.

Qualifier 2: Library

***LIBL** Search all libraries in the job's library list until the first match is found.

***CURLIB**
Search the current library for the job. If no library is specified as the current library for the job, the QGPL library is used.

name Search the specified library.

Printer to host font (PHFCS)

Specifies the printer-resident to host-resident font character set mapping. The printer-resident font, along with its specified attributes will be mapped to a host-resident font character set.

Element 1: Printer font

Element 1: Identifier

1-65535
Specify the printer-resident font identifier to be mapped to a host-resident font.

Element 2: Width

***NONE**
No width is specified for this font identifier. *NONE should be specified when mapping to an outline font.

***PTSIZE**
The width for this font identifier will be calculated from the point size specified. When *PTSIZE is specified for width, the Point size parameter cannot be *NONE or *WIDTH. When mapping a typographic raster font (2304 - 3839, 4096 - 53247, 61440 - 65534), you should specify a value for the Point size parameter. The width value can be *PTSIZE or a value can be given.

1-32767
Specify a width for the font identifier. When mapping a fixed pitch raster font (1 - 750, 3840 - 4095), you should specify a numeric value for Width. Refer to Printer Device Programming,SC41-5713 for more information font mapping tables.

Element 3: Attributes

***NONE**
No special font attributes are specified on this font.

***BOLD**
The printer-resident font is a bold font.

***ITALIC**
The printer-resident font is an italic font.

***BOLDITC**
The printer-resident font is a bold italic font.

***DBLWIDE**
The printer-resident font is a double wide font.

***ITCDBLWIDE**
The printer-resident font is an italic double wide font.

Element 4: Graphic character set

***SYSVAL**
Use the graphic character set specified in the system value QCHRID.

graphic-character-identifier
Specify the graphic character set for the font. The graphic character set is the first part of the graphic character identifier which consists of the graphic character set and code page.

Element 5: Point size

***WIDTH**
The font point size is computed from the font width value specified. When mapping a fixed pitch raster font (1 - 750, 3840 - 4095), it is recommended that a width value should be specified and the point size value should be *WIDTH.

***NONE**
No point size is specified for this font identifier. *NONE should be specified when mapping to an outline font.

1.0-999.9
Specify a point size ranging from 1.0 through 999.9. When mapping a typographic raster font (2304 - 3839, 4096 - 53247, 61440 - 65534), a point size value should be specified.

Element 2: Host font

Element 1: Font character set

name Specify the font character set.

Element 2: Type

***RASTER**
The host-resident font is a raster font.

***OUTLINE**
The host-resident font is an outline font.

Top

Host to printer font (HPFCS)

Specifies the host-resident to printer-resident font character set mapping. The host-resident font, along with its specified attributes will be mapped to a printer-resident font.

Element 1: Host font

Element 1: Font character set

name Specify the font character set.

Element 2: Type

***RASTER**
The host-resident font is a raster font.

***OUTLINE**
The host-resident font is an outline font.

Element 2: Printer font

Element 1: Identifier

1-65535
Specify the printer-resident font identifier. This is the font to which the specified host-resident font will be mapped.

***NONE**
Specify *NONE for the font identifier in order to disable the mapping of a host-resident to a printer-resident font. Refer to Printer Device Programming , SC41-5713 for more information on disabling the mapping of host-resident to printer-resident fonts.

Element 2: Width

***NONE**
Specify *NONE when mapping to an outline font. Outline fonts do not require a width specification.

***PTSIZE**
The width for this font identifier will be calculated from the Point size parameter specified. When *PTSIZE is specified for width, the point size parameter cannot be *NONE or *WIDTH. When mapping a typographic raster font (2304 - 3839, 4096 - 53247, 61440 - 65534), a point size value should be specified. The width value can be *PTSIZE or a value can be given.

1-32767
Specify a width for the font identifier. When mapping a fixed pitch raster font (1 - 750, 3840 - 4095), a width should be specified. The point size value can be *WIDTH or a value can be given. Refer to Printer Device Programming, SC41-5713 for more information on font widths for printer-resident fonts.

Element 3: Attributes

***NONE**
No special font attributes are specified on this font.

***BOLD**
The printer-resident font is a bold font.

***ITALIC**
The printer-resident font is an italic font.

***BOLDITC**
The printer-resident font is a bold italic font.

***DBLWIDE**
The printer-resident font is a double wide font.

***ITCDBLWIDE**
The printer-resident font is an italic double wide font.

Element 4: Graphic character set

***SYSVAL**
The graphic character set specified in the system value QCHRID is used.

graphic-character-identifier
Specify the graphic character set for the font. The graphic character set is the first part of the graphic character identifier which consists of the graphic character set and code page.

Element 5: Point size

***WIDTH**
The font point size is computed from the font width value specified. When mapping a fixed pitch raster font (1 - 750, 3840 - 4095), it is recommended that a width value should be specified and the point size value should be *WIDTH.

***NONE**
No point size is specified for this font identifier. *NONE should be specified when mapping to an outline font.

1.0-999.9
Specify a point size ranging from 1.0 through 999.9. When mapping a typographic raster font (2304 - 3839, 4096 - 53247, 61440 - 65534), a point size value should be specified.

Top

Printer to host code page (PHCP)

Specifies the printer-resident to host-resident code page mapping. The printer-resident code page will be mapped to a host-resident code page.

Element 1: Printer code page

Element 1: Graphic character set

***SYSVAL**
The graphic character set specified in the system value QCHRID is used. A change to this system value will only take effect for the font mapping tables when the print writer is started. If QCHRID is changed and a printer is currently active, you must end the print writer and start it again.

integer-number
Specify the graphic character set for the printer-resident code page. The graphic character set is the first part of the graphic character identifier which consists of the graphic character set and code page. For example, for the graphic character identifier 697 500, 697 is the graphic character set and 500 is the code page. In this example, specify 697 for the graphic character set.

Element 2: Code page

integer-number
Specify the printer-resident code page value.

Element 2: Host code page

Element 1: Code page

name Specify the name of the host-resident code page.

Top

Host to printer code page (HPCP)

Specifies the mapping of a host-resident code page to a printer-resident code page.

Element 1: Host code page

Element 1: Code page

name Specify the name of the host-resident code page.

Element 2: Printer code page

Element 1: Graphic character set

***SYSVAL**
The graphic character set specified in the system value QCHRID is used. A change to this system value will only take effect for the font mapping tables when the print writer is started. If QCHRID is changed and a printer is currently active, you must end the print writer and start it again.

integer-number
Specify the graphic character set for the printer-resident code page. The graphic character set is the first part of the graphic character identifier which consists of the graphic character set and code page. For example, for the graphic character identifier 697 500, 697 is the graphic character set and 500 is the code page. In this example, specify 697 for the graphic character set.

Element 2: Code page

integer-number
Specify the printer-resident code page value.

Top

Printer to printer font (PPFCS)

Specifies the printer-resident font substitution mapping. When a printer-resident font is not supported by a printer, you can specify the substitute printer-resident font to be used instead of the substitute printer-resident font selected by the system. Caution should be used when doing the following types of mapping as undesirable results may occur.

- Mapping a monospaced font to a typographic font. Also for mapping a typographic font to a monospaced font.

- Mapping an outline font to a outline font with a different point size.

Element 1: From printer font

Element 1: Identifier

1-65535
Specify the printer-resident font identifier for which the substitution printer-resident font is to be added. Refer to Printer Device Programming, SC41-5713 for more information on printer-resident fonts that are supported, and which ones are scalable (require point size) and which ones are not scalable (specify point size *NONE).

Element 2: Point size

***NONE**
No font point size is specified. This should be specified for all non-scalable fonts.

***ALL** Specifies that all point sizes for an outline (scalable) font will be mapped. If the font is not scalable, then this will treated the same as *NONE.

1.0-999.9
Specify a point size ranging from 1.0 through 999.9. When mapping a typographic raster font (2304 - 3839, 4096 - 53247, 61440 - 65534), a point size value should be specified.

Element 2: To printer font

Element 1: Identifier

1-65535
Specify the substitute printer-resident font.

Element 2: Point size

***NONE**
No font point size is specified. This should be specified for all non-scalable fonts.

***ALL** Specifies that all point sizes for an outline (scalable) font will be mapped. If the font is not scalable, then this will treated the same as *NONE.

1.0-999.9
Specify a point size ranging from 1.0 through 999.9. When mapping a typographic raster font (2304 - 3839, 4096 - 53247, 61440 - 65534), a point size value should be specified.

Top

Examples

Example 1: Override Existing Font Entry in system table

```
ADDFNTTBLE   FNTTBL(*PHFCS)
             PHFCS((254 84 *NONE 2039 7.0) (C0D0GT18 *RASTER))
```

This command adds an entry to the QPHFCS table (printer-resident to host-resident font character set table). To override the mapping of an existing entry in the system printer-resident to host resident font character set table, you need to add a corresponding entry in the QPHFCS table. Following are the recommended steps to change the mapping of an entry in the system tables.

- Use the DSPFNTTBL command (DSPFNTTBL FNTTBL(*SYSPHFCS) OUTPUT(*PRINT)) to print the entries in the system font mapping table.
- Find the entry you want to change, and add an entry to the corresponding user font mapping table. In the above example, font identifier 254, width of 84, and point size 7.0 is to be added to the user font table (QPHFCS). The width of 84 and point size of 7.0 is were chosen based on an entry in the system table. The entry has no special attributes (*NONE) and graphic character set identifier 2039 will be used.

The attributes of the resident font specified in the print application are compared to those in the font table QPHFCS. If a match is found, then the specified host resident font (C0D0GT18) is downloaded to the printer. If no match is found, then the system printer-resident to host-resident font character set table is searched.

Note that the print application may specify the normal graphic character set (for example, 697 in 697 500 specified in QCHRID system value). The 697 is mapped to 2039 and will result in a match for this entry.

Example 2: Override Existing Font Symbol Entry in system table

```
ADDFNTTBLE   FNTTBL(*PHFCS)
             PHFCS((254 84 *NONE 1275 7.0) (C0SYMBOL *RASTER))
```

This command adds an entry to the QPHFCS table (printer-resident to host-resident font character set table) for use when using the special symbols code page (code page 259). As specified in Example 1, to override an existing entry in the system printer-resident to host-resident font character set table, you need to add a corresponding entry in the QPHFCS table.

Use the DSPFNTTBL command to display the system font mapping table and find the entry you want to change. In this example, you want to add an entry that maps a printer-resident to host resident font character set for the special symbol code page (259). As in the previous example, font identifier 254, width of 84, and point size 7.0 is to be added to the user font table (QPHFCS). The width of 84 and point size of 7.0 is gotten from the system table. The entry has no special attributes (*NONE) and graphic character set 1275 is used.

We now have two entries in the printer-resident to host-resident font character set table. Both entries have the same font identifier, width, and point size. The first entry will be used when the standard code page (500) and graphic set (697) is used by the application. The second entry will be used when a print application specifies special symbols (340 259).

Example 3: Add Font Entry that does not exist in system table

```
ADDFNTTBLE   FNTTBL(*PHFCS)
             PHFCS((65500 *PTSIZE *NONE *SYSVAL 7.0)
                   (C0NEWFNT *RASTER))
```

This command adds an entry to the QPHFCS table (printer-resident to host-resident font character set table) that does not exist in the system printer-resident to host-resident font character set table.

When adding entries that do not exist in the system printer resident to host-resident font character set table, it is recommended that you specify a specific value for font width or point size, but not both. For fixed pitch fonts, you should specify a font width and *WIDTH for point size. For typographic fonts, you

should specify a point size and *PTSIZE for font width. In this example, a typographic font of 65500 with point size 7.0 is added to the printer-resident to host-resident font character set table (QPHFCS).

Example 4: Override Existing Code Page Entry in system table

```
ADDFNTTBLE   FNTTBL(*PHCP)  PHCP((*SYSVAL 38) (T1V00038))
```

This command adds an entry to the QPHCP table (printer-resident to host-resident code page table). To override an existing entry in the system printer-resident to host-resident code page table, you need to add a corresponding entry in the QPHCP table. Following are the recommended steps to change the mapping of an entry in the system tables.

- Use the DSPFNTTBL command (For example, DSPFNTTBL FNTTBL(*SYSPHCP) OUTPUT(*PRINT)) to print the entries in the system code page table.
- Find the entry you want to change, and add an entry into the corresponding user code page table. In the above example, code page 38 is to be added to the user code page table (QPHCP).

In performing the font mapping, the attributes of the resident code page specified in the print application are compared to those in the code page table (QPHCP). If a match is found, then the specified host-resident code page (T1V00038) is downloaded to the printer. If no match is found, then the system printer-resident to host-resident code page table is searched.

Top

Error messages

***ESCAPE Messages**

CPF2182
Not authorized to library &1.

CPF2283
Authorization list &1 does not exist.

CPF88D2
Font table &1 in library &2 not changed.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9822
Not authorized to file &1 in library &2.

Top

Add Host Database to DLFM (ADDHDBDLFM)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Host Database to DLFM (ADDHDBDLFM) command registers a host database with the DataLink File Manager (DLFM). This host database can then be used to call the DLFM to link files.

Restrictions:

- To use this command, you must have input/output system configuration (*IOSYSCFG) special authority.

Top

Parameters

Keyword	Description	Choices	Notes
HOSTDBLIB	Host database library	Values (up to 300 repetitions): *Element list*	Optional, Positional 1
	Element 1: Name	*Character value*	
HOSTDBINST	Host database instance	*Character value*, **QSYS**	Optional, Positional 2
HOSTDB	Host database	*Character value*	Optional
SRCFILE	Source file	*Qualified object name*	Optional
	Qualifier 1: Source file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Character value*	Optional

Top

Host database library (HOSTDBLIB)

Specifies one or more libraries on the host database that may contain database files with DataLinks. Up to 300 names can be specified.

character-value
Specify the library name.

Top

Host database instance (HOSTDBINST)

Specifies the database instance to be registered with the DataLink File Manager. This is used to specify the installation path of the DB2 product on the host system. For iSeries host systems, QSYS should be specified for this parameter.

QSYS The database instance is named QSYS.

character-value
Specify the name of the database instance.

Top

Host database (HOSTDB)

Specifies the host database name to be registered. This is the remote relational database (or server) that will be connecting to the local DataLink File Manager (DLFM) for link and unlink requests of DataLinks.

character-value
Specify the name of the host database.

Top

Source file (SRCFILE)

Specifies the source file that will be used to provide host database information to be registered.

Qualifier 1: Source file

name Specify the name of the source file.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the file. If no library is specified as the current library, QGPL is used.

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the source member that contains the host database information to be registered. Each row of the source member must contain a host database library, a host database instance, and a host database server name, with each value being separated by a space.

character-value
Specify the name of the source file member.

Top

Examples

Registering a Datalink File Manager Host Database

```
ADDHDBDLFM   HOSTDBLIB(MYLIB TESTLIB PERSONNEL)
             HOSTDBINST(QSYS)  HOSTDB(RCHASXYZ)
```

This command registers RCHASXYZ as a valid system for calling the DataLink File Manager with a link request. Libraries MYLIB, TESTLIB, and PERSONNEL are libraries (or collections) on the host database system that can have database files containing DataLinks. QSYS is used as the host database instance, since the system is an iSeries 400 server.

Top

Error messages

***ESCAPE Messages**

CPF3168
DataLink File Manager (DLFM) command failed.

Top

Add ICF Device Entry (ADDICFDEVE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Intersystem Communications Function Device Entry (ADDICFDEVE) command adds a program device entry to the specified intersystem communications function (ICF) file.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File	*Qualified object name*	Required, Positional 1
	Qualifier 1: File	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
PGMDEV	Program device	*Character value*	Required, Positional 2
RMTLOCNAME	Remote location	*Communications name*, *REQUESTER	Required, Positional 3
CMNTYPE	Communication type	***ALL**, *APPC, *ASYNC, *BSCEL, *FINANCE, *INTRA, *RETAIL, *SNUF	Optional, Positional 4
DEV	Device	*Name*, ***LOC**	Optional
LCLLOCNAME	Local location	*Communications name*, ***LOC**, *NETATR	Optional
MODE	Mode	*Communications name*, ***NETATR**	Optional
RMTNETID	Remote network identifier	*Communications name*, ***LOC**, *NETATR, *NONE	Optional
FMTSLT	Format select	***PGM**, *RECID, *RMTFMT	Optional
APPID	Application identifier	*Name*, ***DEVD**, *USER	Optional
BATCH	Batch activity	*YES, ***NO**	Optional
HOST	Host type	***DEVD**, *CICS, *IMS, *IMSRTR	Optional
ENDSSNHOST	End session with host	***RSHUTD**, *TERMSELF	Optional
SPCHOSTAPP	Special host application	***DEVD**, *NONE, *FLASH	Optional
INZSELF	Initialize self	***NO**, *YES	Optional
HDRPROC	Header processing	***SYS**, *USER	Optional
MSGPTC	Message protection	***YES**, *NO	Optional
EMLDEV	Emulation device	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Device type	3278, 3284, 3286, 3287, 3288, 3289	
	Element 2: Data format	***UNFORMAT**, *FIELD, *EXTFIELD, *NOFIELD	
CNVTYPE	Conversation type	***SYS**, *USER, *SRCPGM	Optional
BLOCK	Blocking type	*Element list*	Optional
	Element 1: Blocking type	***DEVD**, *NONE, *ITB, *IRS, *NOSEP, *USER, *SEP	
	Element 2: Record separator, if *SEP	*Hexadecimal value*, **X'1E'**	
RCDLEN	Record length	1-32767, ***DEVD**	Optional
BLKLEN	Block length	1-32767, ***DEVD**	Optional

Keyword	Description	Choices	Notes
TRNSPY	Transmit in transparent mode	***DEVD**, *NO, *YES	Optional
DTACPR	Compress and decompress data	***DEVD**, *NO, *YES	Optional
TRUNC	Truncate trailing blanks	***DEVD**, *NO, *YES	Optional
OVRFLWDTA	Overflow data	***DISCARD**, *RETAIN	Optional
GRPSEP	Group separator type	***DEVD**, *EOT, *DEV3740, *OFCSYS	Optional
RMTBSCEL	Remote BSCEL	***DEVD**, *NO, *YES	Optional
INLCNN	Initial connection	***CTLD**, *DIAL, *ANS	Optional

Top

File (FILE)

Specifies the intersystem communications function (ICF) file to which the ICF program device entry is to be added.

This is a required parameter.

Qualifier 1: File

name Specify the name of the ICF file.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library is used to locate the ICF file. If no library is specified as the current library for the job, QGPL is used.

name Specify the library where the ICF file is located.

Top

Program device (PGMDEV)

Specifies the name by which the ICF program device entry being added is known. The total number of program device entries that may be added to an ICF file (as specified on the ADDICFDEVE or OVRICFDEVE command) is determined by the **Maximum program devices (MAXPGMDEV)** parameter on the Create Intersystem Communications Function File (CRTICFF) or Change Intersystem Communications Function File (CHGICFF) command.

The name specified for this parameter is the ICF program device entry with which the user's program communicates. This name is used on device-specific input/output operations to identify the program device and its attributes. Although the user may specify the same remote location name on more than one device entry, each program device name must be unique among the entries for the ICF file. This allows the user to have more than one session to the same remote location, or to have different attribute values for each session to the same remote location.

Note: Refer to the APPC Programming book, SC41-5443 for information on how the system uses the RMTLOCNAME, DEV, LCLLOCNAME, and RMTNETID parameters to select an APPC device description.

This is a required parameter.

character-value
Specify the name of the ICF program device entry being added. This name is used on device-specific input/output operations to identify the program device entry and the attributes. This program device entry name must be unique throughout the program device entries for the ICF file.

Top

Remote location (RMTLOCNAME)

Specifies the name of the remote location with which your program communicates.

This is a required parameter.

***REQUESTER**
The name used to refer to the communications device through which the program is started is used. The session that is assigned when the program device is acquired is the same session in which the program start request is received. If the program is not started as a result of a program start request, the acquire operation of the program device fails. The target program uses *REQUESTER as the remote location name in the intersystem communications function (ICF) file to connect to the session that the source program used to send the program start request.

The *REQUESTER value can be specified on only one program device entry and is valid only for a target communication job. If *REQUESTER is specified in any other type of job, a message is sent.

communications-name
Specify the full name of a remote location. The remote location does not need to exist at the time this command is run, but it must exist (be configured on the system as a device description or in the advanced peer-to-peer networking (APPN) function) for this remote location at the time the program acquires the program device. The same remote location name may be specified for many different program device entries. However, only one program device name associated with each asynchronous (ASYNC), SNA upline facility (SNUF), or binary synchronous communication equivalence link (BSCEL) remote location may be added to the file at any one time. This value cannot be specified with CNVTYPE(*SRCPGM).

Top

Communication type (CMNTYPE)

Specifies the communications types that may appear in the prompt display. This parameter is used only for the purpose of prompting. The value specified for this parameter determines the subset of other parameters that are shown (prompted) for the user.

***ALL** All parameters appear in the prompt.

***APPC**
Only the advanced program-to-program communications (APPC) parameters appear in the prompt.

***ASYNC**
Only the asynchronous communications (ASYNC) parameters appear in the prompt.

***BSCEL**
Only the binary synchronous communications equivalence link (BSCEL) parameters appear in the prompt.

***FINANCE**
Only the FINANCE parameters appear in the prompt.

***INTRA**
Only the INTRA parameters appear in the prompt.

***RETAIL**
Only the RETAIL parameters appear in the prompt.

***SNUF**
Only the Systems Network Architecture Upline Facility (SNUF) parameters appear in the prompt.

Top

Device (DEV)

Specifies the communications device used in the remote location. This parameter is specified only for APPC, Finance, Retail, SNUF, and INTRA communications.

***LOC** The device associated with the remote location is used. If several devices are associated with the remote location, the system determines which device is used.

name Specify the name of a communications device associated with the remote location. If the device name is not valid for the remote location, an escape message is sent when the program device is acquired. More information on device names is in the APPC Programming book, SC41-5443.

Top

Local location (LCLLOCNAME)

Specifies the local location name.

***LOC** The local location name associated with the remote location is used.

***NETATR**
The LCLLOCNAME value specified in the system network attributes is used.

communications-name
Specify the local location name associated with the program device. The local location name is specified only if you want to indicate a different specific local location name for the remote location. If the local location name is not valid, an escape message is sent when the program device is acquired.

Top

Mode (MODE)

Specifies the mode name being used. This parameter applies only to APPC communications.

***NETATR**
The mode in the network attributes is used.

BLANK
A mode name consisting of 8 blank characters is used.

communications-name
Specify a mode name for the APPC communication device. If the mode is not valid for any combination of remote location device, local location, and remote network ID, an escape message is sent when the program device is acquired.

Top

Remote network identifier (RMTNETID)

Specifies the remote network identifier used with the remote location. This parameter applies only to the APPC communications.

***LOC** Any remote network identifier for the remote location may be used. If several remote network IDs are associated with the remote location, the system determines which remote network ID is used.

***NETATR**
The remote network identifier specified in the network attributes is used.

***NONE**
No remote network identifier is used.

communications-name
Specify a remote network identifier to be associated with the program device entry.

Top

Format select (FMTSLT)

Specifies the type of record format selection used for input operations.

***PGM** The program determines record format selections. If an input (read) operation with a record format name is specified, that format is always selected. If a record format is not specified for the input operation, the default format (the first record format in the file) is always selected.

***RECID**
The RECID keywords specified in the data description specifications (DDS) for the file are used to do a record selection. If there are no RECID keywords in the DDS, an error message is returned, and the program device is not acquired.

***RMTFMT**
The remote format names received from the sending system are used to do a record selection. If the device is not an APPC device and *RMTFMT is specified, a run time error message is sent at the time the program device is acquired.

Top

Application identifier (APPID)

Specifies (in characters) the Virtual Telecommunications Access Method (VTAM) identifier of the Customer Information Control System for Virtual Storage (CICS/VS) or the Information Management System for Virtual Storage (IMS/VS) host subsystem sent with the sign-on message. This parameter applies only to SNUF communications.

***DEVD**
The application identifier specified in the device description is used.

***USER**
The application program can send messages or a logon to the host. This is valid only when using the 3270 program interface.

name Specify an application identifier for the VTAM of the CICS/VS or IMS/VS.

Top

Batch activity (BATCH)

Specifies, for both the Customer Information Control System for Virtual Storage(CICS/VS) and the Information Management System for Virtual Storage (IMS/VS), whether this session is used for batch jobs. This parameter applies to SNUF, INTRA, and RETAIL communications.

***NO** Batch jobs do not occur in this session.

***YES** Batch jobs occur and SNUF does not assemble physical records into logical records. If *YES is specified, *NO must be specified for the **Message protection (MSGPTC)** parameter.

Top

Host type (HOST)

Specifies the host system or remote subsystem with which the local system or subsystem is communicating during this session. This parameter applies only to SNUF communications.

***DEVD**
The host system specified in the device description is used.

***CICS** The Customer Information Control System for Virtual Storage is used during this communication session.

***IMS** The Information Management System for Virtual Storage is used during this communication session.

***IMSRTR**
The Information Management System for Virtual Storage uses the ready-to-receive option during this communication session.

Top

End session with host (ENDSSNHOST)

Specifies how the Systems Network Architecture upline facility (SNUF) ends the session with the host.

***RSHUTD**
SNUF sends a request to turn off command to the host.

***TERMSELF**
SNUF sends an end-session command to the host. This value may have to be used if the value *RSHUTD fails to end a session with a non-IBM host.

Top

Special host application (SPCHOSTAPP)

Specifies whether SNUF customizes support for special host applications outside the CICS or IMS application layer.

***DEVD**
The special host application specified in the device description is used.

***NONE**
SNUF does not customize support for special host applications.

***FLASH**
SNUF customizes support for the Federal Reserve Flash application.

Top

Initialize self (INZSELF)

Specifies whether a formatted INIT-SELF is built in place of the unformatted sign-on normally sent by SNUF to the host.

***NO** The unformatted default sign-on provided by SNUF is used.

***YES** The formatted INIT-SELF provided by SNUF is used.

Top

Header processing (HDRPROC)

Specifies, for both the Customer Information Control System for Virtual Storage (CICS/VS) and the Information Management System for Virtual Storage (IMS/VS) whether the received function management headers are passed to the application program. This parameter applies only to SNUF communications.

***SYS** SNUF removes function management headers before passing data to the program.

***USER**
SNUF does not remove function management headers before passing data to the program.

Top

Message protection (MSGPTC)

Specifies, for both the Customer Information Control System for Virtual Storage (CICS/VS) and Information Management System for Virtual Storage (IMS/VS), whether message protection is being used for this session. This parameter applies only to SNUF communications.

***YES** Message protection is used. SNUF saves messages until you respond to them, and tries synchronization again if additional errors occur. *YES is valid only when *NO is specified for the **Batch activity (BATCH)** parameter.

***NO** Message protection is not used.

Top

Emulation device (EMLDEV)

Specifies that this program device entry is used to send and receive 3270 data streams. The emulation device parameter consists of an emulation device type and an emulation device data format. The emulation device data format specifies the format of the type 3270 data stream being sent or received. A 20- or 32-byte common header that contains type 3270 command and data flow information is located at the start of the I/O buffer that is sending or receiving the type 3270 data stream. This parameter applies only to SNUF communications. This parameter can be specified as a list of two values (elements) or as a single value (*NONE).

The emulation device data format specifies the format of the type 3270 data stream being sent or received. A 20- or 32-byte common header that contains type 3270 command and data flow information is located at the start of the I/O buffer that is sending or receiving the type 3270 data stream.

Single values

***NONE**
This program device entry is not used to send and receive 3270 data streams.

Element 1: Device type

3278 The data stream is for a 3278, 3277, or 3279 display device.

3284 The data stream is for a 3284 printer device.

3286 The data stream is for a 3286 printer device.

3287 The data stream is for a 3287 printer device.

3288 The data stream is for a 3288 printer device.

3289 The data stream is for a 3289 printer device.

Element 2: Data format

***UNFORMAT**
An unformatted 3270 data stream is sent or received. The user application program must translate the data stream into a display or printer image.

***FIELD**
A formatted 3270 data stream is sent or received. The formatted 3270 data stream contains a display or printer image that contains field definitions. The field definitions indicate the location and characteristics of the fields. *FIELD is valid only if *NO is specified for the **Batch activity (BATCH)** parameter.

***NOFIELD**
A formatted 3270 data stream is sent or received. The formatted 3270 data stream contains a display or printer image without field definitions. *NOFIELD is valid only if *NO is specified for the BATCH parameter.

***EXTFIELD**
A formatted 3270 data stream is sent or received. The formatted 3270 data stream contains a display or printer image followed by field definitions. The field definitions indicate the location and characteristics of fields. *EXTFIELD is valid only if *NO is specified for the BATCH parameter and 3278 is specified as the emulation device type.

Top

Conversation type (CNVTYPE)

Specifies the conversation type for which the application program is designed. This parameter is valid only for (APPC) communications. More information on the APPC communications type can be found in the APPC Programming book, SC41-5443.

***SYS** The system gives the length and general data stream identifier values that precede each section of user data in the APPC device protocol. The application gives the data portion of the general data stream on output operations, and receives only the data portion of the general data stream on input operations. This is the mapped conversation support for LU 6.2 architecture.

***USER**
The application program gives the length and general data stream identifier values that precede each section of user data in the APPC device protocol. This is the basic conversation support for LU 6.2 architecture.

***SRCPGM**
The target program accepts the conversation type specified by the source program. If this value is specified, RMTLOCNAME(*REQUESTER) must also be specified.

Top

Blocking type (BLOCK)

Specifies whether the system or the user controls how records are combined into blocks when they are sent. This parameter is only for BSCEL communications. With this parameter, you may specify one of the following conditions of record formatting:

- No blocking or deblocking: The record format described in the DDS is the format for both the record and the block.
- User blocking or deblocking: Gives the BSC controls needed to describe the record format of the system.
- System blocking with record separator characters: Specify the record separator character used by the system to determine record boundaries within the block.
- System blocking of fixed-length records: The system uses fixed-length records, and blocks or deblocks accordingly.

If you specify a value other than *NONE or *USER, records are blocked as required by the system for output, and are deblocked on input.

Element 1: Blocking type

***DEVD**
The block option specified in the device description is used.

***NONE**
Blocking or deblocking is not done by the system.

***ITB** The records are blocked or deblocked based on the location of an intermediate text block (ITB) control character. For input files, a record is delimited by locating the next intermediate text block character. An end-of-text or end-of-transmission block character is used as an intermediate text block character to delimit a block. For output files, an ITB character is added after the record. If it is the last character of the block, the ITB is replaced by an end-of-text or end-of-transmission block character.

***IRS** The records are blocked or deblocked based on the location of an interrecord separator (IRS) character. For input files, a record is delimited by locating the next IRS character. For output files, an IRS character is added following the record.

***NOSEP**
No record separator character is contained in the block that is either sent to or received from the device. The system blocks and deblocks the records by using a fixed-length record, as specified in the data description specifications (DDS) format specifications.

***USER**
The program supplies the control characters, including record separator characters, BSCEL framing characters, transparency characters, and so forth, that are necessary to send records. More information about the device and binary synchronous communications equivalence link (BSCEL) support characteristics is in the BSC Equivalence Link Programming book, SC41-5445.

***SEP** The records are blocked or deblocked based on the location of a record separator character specified by the user. For input files, a record is delimited by locating the next record separator character. For output files, a record separator character is added after the record.

Element 2: Record separator, if *SEP

X'1E' Record separator character X'1E' is used.

hexadecimal-value
Specify a unique, 1-byte record separator character. The record separator character may be specified as 2 hexadecimal characters, as in BLOCK(*SEP FD) where X'FD' is the separator character.

The following characters are not not valid as separator characters because these characters are used for BSC control:

Table 1. **Figure: Characters Unavailable for Record Separators**

EBCDIC	ASCII	BSC Control
X'01'	X'01'	SOH (start-of-header)
X'02'	X'02'	STX (start-of-text)
X'03'	X'03'	ETX (end-of-text)
X'10'	X'10'	DLE (data-link escape)
X'1D'	X'1D'	IGS (interchange group separator)
X'1F'	X'1F'	ITB (intermediate text block)
X'26'	X'17'	ETB (end-of-transmission block)
X'2D'	X'05'	ENQ (enquiry)
X'32'	X'16'	SYN (synchronization)
X'37'	X'04'	EOT (end-of-transmission)
X'3D'	X'15'	NAK (negative acknowledgment)

Top

Record length (RCDLEN)

Specifies the maximum record length (in bytes) for data being sent and received. This parameter applies only to SNUF and BSCEL communications.

***DEVD**
The record length specified in the device description is used. If a record is longer than the specified record length, a run time error occurs at the time the record is sent or received.

1-32767
Specify the maximum allowed record length when this device file is used. If a record is longer than the specified record length, a run time error occurs at the time the record is sent or received. Valid values range from 1 through 32767 bytes for SNUF communications. For BSCEL communications, the maximum record length is 8192 bytes.

Top

Block length (BLKLEN)

Specifies the maximum block length (in bytes) for data being sent. This parameter applies only to BSCEL and SNUF communications.

***DEVD**
The block length specified in the device description is used.

1-32767
Specify the maximum block length (in bytes) of records sent when using this device file. The value must be at least the size of the largest record sent. Valid values range from 1 through 32767 for SNUF communications. For BSCEL communications, the maximum block length is 8192 bytes.

Top

Transmit in transparent mode (TRNSPY)

Specifies whether data is sent in transparent text mode. Text transparency allows all 256 EBCDIC character codes to be sent. Use this function when sending packed or binary data fields. This parameter applies only to BSCEL communications.

***DEVD**
The text transparency option specified in the device description is used.

***NO** Text transparency is not used.

***YES** Text transparency is used, which allows all 256 EBCDIC character codes to be sent. *YES is valid only when *NONE, *NOSEP, or *USER is specified for the **Blocking type (BLOCK)** parameter.

Note: Transparency of received data is determined by the data stream; therefore, this parameter is not relevant for received data. If TRNSPY(*YES) is specified with BLOCK(*USER), BSCEL ignores the transparency indicator during write operations. Correct controls must be given with the data to get transparent sending of data. For example, the data-link escape (DLE) and start-of-text (STX) control characters must first be specified; the system provides the remaining control characters required for transparent sending of data.

Top

Compress and decompress data (DTACPR)

Specifies whether blanks in BSCEL data are compressed for output and decompressed for input. This parameter applies only to BSCEL communications.

***DEVD**
The data compression option specified in the device description is used.

***NO** No data compression or decompression is used.

***YES** Data is compressed for output and decompressed for input. *YES cannot be specified here if *YES is specified for the **Transmit in transparent mode (TRNSPY)** parameter.

Top

Truncate trailing blanks (TRUNC)

Specifies whether trailing blanks are removed from output records. This parameter applies only to BSCEL communications.

***DEVD**
The trailing blanks option specified in the device description is used.

***NO** Trailing blanks are not removed from output records.

***YES** Trailing blanks are removed from output records. *YES cannot be specified if BLOCK(*NOSEP) is specified for the **Blocking type (BLOCK)** parameter. If *YES is specified here and *YES is also specified for the **Compress and decompress data (DTACPR)** parameter, trailing blanks are not removed from output records.

Top

Overflow data (OVRFLWDTA)

Specifies whether overflow data is discarded or retained.

***DISCARD**
Overflow data is not kept.

***RETAIN**
Overflow data is kept.

Top

Group separator type (GRPSEP)

Specifies a separator for groups of data (for example, data sets and documents). This parameter applies only to BSCEL communications.

***DEVD**
The group separator option specified in the device description is used.

***DEV3740**
A null record (STXETX) is used as a data group separator.

***EOT** A block is sent that ends with the BSCEL end-of-transmission (EOT) control character used as a data group separator.

***OFCSYS**
A block is sent that ends with the BSCEL end-of-information (ETX) control character is used as a data group separator.

Top

Remote BSCEL (RMTBSCEL)

Specifies the type of BSCEL session established with the remote system. This parameter applies only to BSCEL communications.

***DEVD**
The RMTBSCEL option specified in the device description is used.

***NO** The remote system cannot recognize BSCEL commands or messages. In most cases, *NO is used when communicating with remote systems such as a 3741 Data Entry Station, an Office System 6, a 5230 Data Collection System, or a System/38.

***YES** The remote system recognizes the BSCEL transaction starting commands, transaction ending commands, and online messages. In most cases, *YES indicates that the remote system is another System i5, or a System/38, System/36, or System/34 with BSCEL support.

Top

Initial connection (INLCNN)

Specifies the method of making a connection on the line for the session being accessed. This parameter applies only to binary synchronous communications equivalence link (BSCEL) communications.

***CTLD**
The initial connection option specified in the controller description is used.

***ANS** The remote system starts the call, and the local system answers the call.

***DIAL**
The local system starts the call, and the remote system answers the call.

Top

Examples

Example 1: Using RECID Keywords for Record Selection

```
ADDICFDEVE   FILE(ICFFILE1)  PGMDEV(BSCEL2)
             RMTLOCNAME(BSCNYC)  FMTSLT(*RECID)
```

This command adds the program device entry named BSCEL2 with a corresponding remote location named BSCNYC for the ICF file ICFFILE1. The program device is added with the attributes of FMTSLT(*RECID).

Example 2: Using Remote Format Names for Record Selection

```
ADDICFDEVE   FILE(QGPL/ICFTEST)  PGMDEV(APPC1)
             RMTLOCNAME(*REQUESTER)
             FMTSLT(*RMTFMT)  CNVTYPE(*SYS)
```

This command adds the program device entry named APPC1 with a remote location name of *REQUESTER for the ICF file ICFTEST in the QGPL library. This program device entry has the FMTSLT(*RMTFMT) and CNVTYPE(*SYS) attributes.

Example 3: Adding a Program Device Entry

```
ADDICFDEVE   FILE(ICFLIB/TESTFILE)  PGMDEV(JOE)
             RMTLOCNAME(LU0MPLS)
```

This command adds the program device entry named JOE with remote location named LU0MPLS for the ICF file TESTFILE in library ICFLIB.

Example 4: Adding a Program Device Entry

```
ADDICFDEVE   FILE(TESTFILE)  PGMDEV(APPC)
             RMTLOCNAME(APPCMPLS)  DEV(MPLSLINE2)
```

This command adds the program device entry named APPC with a remote location name of APPCMPLS using device MPLSLINE2 to the ICF file TESTFILE.

Top

Error messages

***ESCAPE Messages**

CPF7365

Device not added to file &1 in &2.

Top

Add Image Catalog Entry (ADDIMGCLGE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Image Catalog Entry (ADDIMGCLGE) command is used to create a virtual image in the image catalog directory (as specified by the **Directory (DIR)** parameter on the CRTIMGCLG command). If the image is added successfully, the image will be loaded and the image catalog (*IMGCLG) in library QUSRSYS will be updated. Images can be added from the following sources:

1. CD or DVD media (by specifying an optical device containing the CD or DVD image to be added). This source is only allowed for optical image catalogs.
2. An image file located in a directory. The image file can exist in any directory that is accessible through the Copy Object (CPY) command.
3. A new image file by specifying the FROMFILE(*NEW) parameter

If an image catalog entry exists with the same index as specified for the **Image catalog index (IMGCLGIDX)** parameter, one of the following will occur:

1. If REPLACE(*NO) is specified, an error message will be issued and the catalog entry in the image catalog will not be replaced.
2. If REPLACE(*YES) is specified, the catalog entry in the image catalog will be replaced.
3. If REPLACE(*INSERT) is specified, the catalog entry in the image catalog will be inserted. If a catalog entry already exists at the index specified, the remaining entries will be incremented by one up to the next available index.

If the image does not exist in the target directory, the image is copied and the image catalog is updated with information about this image.

Restrictions:

- The following authorities are required to add an image catalog entry:
  1. Execute (*EXECUTE) authority to library QUSRSYS.
  2. Change (*CHANGE) authority to the image catalog.
  3. Execute (*X) authority to each directory in the image catalog path name.
  4. If adding an image file from physical media, use (*USE) authority to the physical device is required.
  5. If adding an image file from an existing image located in a different directory, the authorities required are the same as those required by the Copy Object (CPY) command.
  6. If adding an image file from an existing image located in the image catalog directory, read (*R) and object management (*OBJMGT) authorities are required.
- Adding from physical CD or DVD media is only supported for optical image catalogs.
- This command does not support dependent image catalogs.
- The following parameters are only valid for tape image catalogs:
  1. ALCSTG
  2. VOLNAM
  3. VOLTYP
  4. DENSITY
  5. NEWOWNID
  6. CODE

If any of the above parameters are specified for an optical image catalog, they will be ignored.

Top

Parameters

Keyword	Description	Choices	Notes
IMGCLG	Image catalog	*Name*	Required, Positional 1
FROMDEV	From optical device, or	*Name*	Optional
FROMFILE	From image file	*Path name*, *NEW	Optional
TOFILE	To image file	*Character value*, ***GEN**, *FROMFILE	Optional
IMGCLGIDX	Image catalog index	1-256, ***AVAIL**	Optional
REPLACE	Replace catalog entry	***NO**, *YES, *INSERT	Optional
IMGSIZ	Image size	48-1000000, ***IMGCLGTYPE**, *CD650, *DVD2600, *DVD4700	Optional
TEXT	Text 'description'	*Character value*, ***GEN**, *BLANK	Optional
ALCSTG	Allocate storage size	***MIN**, *IMGSIZ	Optional
VOLNAM	Tape volume name	*Character value*, ***GEN**	Optional
VOLTYP	Volume type	***SL**, *NL	Optional
DENSITY	Tape density	***VRT256K**, *VRT240K, *VRT64K, *VRT32K	Optional
NEWOWNID	New owner identifier	*Character value*, ***BLANK**	Optional
CODE	Code	***EBCDIC**, *ASCII	Optional

Top

Image catalog (IMGCLG)

Specifies the image catalog that the new entry is to be added to.

This is a required parameter.

name Specify the name of the image catalog.

Top

From optical device, or (FROMDEV)

Specifies the CD or DVD device from which the optical image is to be copied. For optical image catalogs, a value must be specified for either the FROMDEV or FROMFILE parameter to identify the optical image file to be copied to the image catalog directory. For tape image catalogs, the FROMDEV parameter is not allowed.

name Specify the name of the optical device from which the optical image file is to be copied.

Top

From image file (FROMFILE)

Specifies the image file to be copied. For optical image catalogs, a value must be specified for either the FROMDEV or FROMFILE parameter to identify the optical image file to be copied to the image catalog directory. For tape image catalogs, a value must be specified on the FROMFILE parameter to identify the tape image file to be copied to the image catalog directory.

***NEW** A new catalog entry is to be added and an empty image file created.

name Specify the name of the optical or tape image file to be added to the image catalog.

Top

To image file (TOFILE)

Specifies the name given to the file that will be copied to the target directory.

***GEN** The file name will be generated from the source image.

***FROMFILE**
The file name will be the same name as source image. If the source image is from optical media, the name given to the file will be the volume ID. If the source image is from another directory, the file will be given the same name as that in the source directory.

name Specify up to 255 characters for the name of the optical or tape image file to be added to the image catalog.

Top

Image catalog index (IMGCLGIDX)

Specifies the image catalog index to be assigned to the image being added.

***AVAIL**
The image catalog index number assigned to the image will be the first index number available.

1-256 Specify the image catalog index number to be used.

Top

Replace catalog entry (REPLACE)

Specifies the action to take if a catalog entry with the same index number as specified on the IMGCLGIDX parameter already exists in the image catalog.

***NO** The existing catalog entry will not be replaced and an error message will be issued.

***YES** The existing image catalog entry will be replaced.

***INSERT**
The catalog entry specified will be added (or inserted if an existing index number already exists). If the image catalog entry is inserted, the remaining catalog entries will be incremented by one up to the next available index number.

Top

Image size (IMGSIZ)

Specifies the size of the new image file that is to be created.

For optical image catalogs, this parameter indicates the size of the optical image file that is to be created and the amount of system storage to be allocated.

For tape image catalogs, this parameter indicates the maximum size allowed for this tape image file. If ALCSTG(*MIN) is specified, only the amount of storage required to initialize the image file will be allocated. If ALCSTG(*IMGSIZ) is specified, the amount of storage allocated will be the value specified on the IMGSIZ parameter.

***IMGCLGTYPE**
The size of the image file to be created will be determined by the type of image catalog.

For optical image catalogs, the size of the image file created will be 650 MB. Optical images created with this size can be written to any standard 650 MB media.

For tape image catalogs, the maximum size of the image file created will be set to 1 gigabyte.

***CD650**
The size of the optical image file will be 650 megabytes. Optical images created with this size can be written to any standard 650 MB optical media.

***DVD2600**
The size of the optical image file will be 2.6 gigabytes. Optical images created with this size can be written to any 2.6 GB media.

***DVD4700**
The size of the optical image file will be 4.7 gigabytes. Optical images created with this size can be written to any 4.7 GB media.

number
Specify the number of megabytes for the new image file. For optical image files, the valid range is 48-16000 megabytes. For tape image files, the valid range is 48-1000000 megabytes.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the image being loaded.

***GEN** For optical image catalogs, the text field will be set to the description on the media if it exists, otherwise it will be set to the volume ID of the media. For tape image catalogs, the text will be generated by the system using the current date and time.

***BLANK**
The text description will be blank.

character-value
Specify up to 50 characters of text for this image file.

Top

Allocate storage size (ALCSTG)

Specifies for tape image files, whether to allocate the entire amount of storage specified on the **Image size (IMGSIZ)** parameter.

***MIN** Only allocate storage required to initialize the tape image file at the time of the create.

***IMGSIZ**
Allocate the entire amount of storage specified on the **Image size (IMGSIZ)** parameter.

Top

Tape volume name (VOLNAM)

Specifies the volume name for this virtual tape volume.

***GEN** The volume name will be generated by the system.

character-value
Specify up to a 6 character volume name used to identify this virtual tape volume. The volume name must contain only alphanumeric characters (A through Z, $, #, @, and 0 through 9), and cannot have a prefix or contain blanks. If the virtual tape volume is being created as a standard labeled tape, the volume name will also be used as the volume identifier in the tape volume label. Each volume name in the catalog must be unique.

Top

Volume type (VOLTYP)

Specifies the type of volume for this virtual tape volume.

***SL** The type of volume is a standard labeled tape volume.

***NL** The type of volume is a non-labeled tape volume.

Top

Tape density (DENSITY)

Specifies the density for this virtual tape volume.

***VRT256K**
The format of the volume is *VRT256K. It is used to write data to a virtual volume using a maximum data block size of 256KB. Volumes written using this format can only be duplicated to tape devices that support a maximum block size of 256KB or greater.

***VRT240K**
The format of the volume is *VRT240K. It is used to write data to a virtual volume using a maximum data block size of 240KB. Volumes written using this format can only be duplicated to tape devices that support a maximum block size of 240KB or greater.

***VRT64K**
The format of the volume is *VRT64K. It is used to write data to a virtual volume using a maximum data block size of 64KB. Volumes written using this format can only be duplicated to tape devices that support a maximum block size of 64KB or greater.

***VRT32K**
The format of the volume is *VRT32K. It is used to write data to a virtual volume using a maximum data block size of 32KB. Volumes written using this format can be duplicated to all supported tape devices.

Top

New owner identifier (NEWOWNID)

Specifies the identifier of the tape owner for this virtual tape volume. This parameter is only used for a volume type of *SL.

***BLANK**
No identifier will be specified.

name Specify no more than 14 characters that identify the owner of the virtual tape volume. If fewer than 14 characters are specified, the field is left-justified and padded on the right with blanks.

Top

Code (CODE)

Specifies the character code in which the volume label is written. All data that is not save data written after the label must be in the same code. Codes cannot be intermixed on a tape that is not a save tape. This parameter is only used for a volume type of *SL.

***EBCDIC**
The volume label is written in EBCDIC and is an IBM standard label; all additional data must also be written in EBCDIC.

***ASCII**
The volume label is written in ASCII and is an ANSI standard label; all additional data must also be written in ASCII.

Top

Examples

Example 1: Adding an Image Catalog Entry from CD/DVD Media

```
ADDIMGCLGE   IMGCLG(MYCLG)  FROMDEV(OPT01)
```

This command adds the optical image in device **OPT01** to the image catalog directory and updates catalog **MYCLG**. The image catalog directory is the directory specified for the **Directory (DIR)** parameter of the CRTIMGCLG command.

Example 2: Adding an Image Catalog Entry from a Directory

```
ADDIMGCLGE   IMGCLG(MYCLG)  FROMFILE('/MyDir/MyFile.img')
```

This command adds the image file **/MyDir/Myfile.img** to the image catalog directory /MyNewDir and updates catalog MYCLG. The image catalog directory is the directory specified for the **Directory (DIR)** parameter of the CRTIMGCLG command.

Example 3: Inserting an Image Catalog Entry

```
ADDIMGCLGE   IMGCLG(MYCLG)  FROMDEV(OPT01)  REPLACE(*INSERT)
```

This command adds the optical image in device **OPT01** to the image catalog directory and, if a catalog entry exists at the specified index, will insert the entry in the catalog and increment the remaining entries up to the next available index. The image catalog directory is the directory specified for the **Directory (DIR)** parameter of the CRTIMGCLG command.

Example 4: Adding a New Optical Image Catalog Entry

```
ADDIMGCLGE   IMGCLG(MYCLG)  FROMFILE(*NEW)  TOFILE(MYFILE)
             IMGSIZ(*CD650)
```

This command adds new optical image file **MYFILE** to the image catalog directory with a size of 650 megabytes. The image catalog directory is the directory specified for the **Directory (DIR)** parameter of the CRTIMGCLG command.

Example 5: Adding a New Tape Image Catalog Entry

```
ADDIMGCLGE   IMGCLG(TAPECLG)  FROMFILE(*NEW)  TOFILE(TAP001)
             IMGSIZ(*IMGCLGTYPE) ALCSTG(*MIN) VOLNAM(MYVOL)
             VOLTYP(*SL) DENSITY(*VRT256K) NEWOWNID(MYNAME)
             CODE(*EBCDIC)
```

This command adds new image file **TAP001** with a volume name of **MYVOL** to the target directory with the default maximum allowed size of 1000 megabytes. Only storage required to create and initialize the virtual volume will be allocated at the time of the create. The target directory is the directory specified for the **Directory (DIR)** parameter of the CRTIMGCLG command.

Example 6: Adding a New Tape Image Catalog Entry with Allocation.

```
ADDIMGCLGE   IMGCLG(TAPECLG)  FROMFILE(*NEW)  TOFILE(TAP002)
             IMGSIZ(5000) ALCSTG(*IMGSIZ)
```

This command adds new image file **TAP002** to the target directory with a maximum size of 5000 megabytes. All 5000 megabytes of storage will be allocated for this image file when it is created. The target directory is the directory specified for the **Directory (DIR)** parameter of the CRTIMGCLG command.

Example 7: Adding an Image Catalog Entry from the Image Catalog Directory

```
ADDIMGCLGE   IMGCLG(MYCLG)  FROMFILE('/MyClgDir/MyFile.img')
             TOFILE(*FROMFILE)
```

This command updates the image catalog MYCLG with information about image file **MyFile.img**.

Top

Error messages

***ESCAPE Messages**

CPFBC28
Image catalog entry not added to image catalog &1.

CPFBC45
Image catalog &1 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9820
Not authorized to use library &1.

Top

Add IP over SNA Interface (ADDIPSIFC)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add IP over SNA Interface (ADDIPSIFC) command is used to define AF_INET Sockets over SNA interfaces. An interface is an IP address by which this local host is known on the SNA transport. The interfaces defined by the ADDIPSIFC command are logical interfaces. They are not physical interfaces and they are not associated with any line description or network interface. There may be multiple AF_INET Sockets over SNA logical interfaces defined on a host.

Note: When an interface is added it is activated by default if AF_INET Sockets over SNA communications is active and the user issuing the ADDIPSIFC command is authorized to start AF_INET Sockets over SNA interfaces. A user must have authority to the Start IP over SNA Interface (STRIPSIFC) CL command to have the authority to start AF_INET Sockets over SNA interfaces. If the interface should not be active, use the End IP over SNA Interface (ENDIPSIFC) CL command to deactivate it.

Only eight (8) AF_INET sockets over SNA interfaces can be active on a single host. If the maximum number of interfaces are already active, the interface being added will not be started. If you want the interface started, you must first end one or more active interfaces using the End IP over SNA interfaces (ENDIPSIFC) CL command and then use the Start IP over SNA interfaces (STRIPSIFC) CL command to start the interface.

Restriction: The user must have *IOSYSCFG authority to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
INTNETADR	Internet address	*Character value*	Required, Positional 1
SUBNETMASK	Subnet mask	*Character value*	Required, Positional 2

Top

Internet address (INTNETADR)

Specifies an internet address that the local system responds to on this interface. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

This is a required parameter.

Restrictions:

1. The internet address cannot begin with 0 (for example, 0.nnn.nnn.nnn).

2. The internet address cannot begin with 127 (for example, 127.nnn.nnn.nnn). This address range is reserved for TCP/IP loopback addresses.
3. The internet address cannot be a class D or class E address. Class D addresses range from 224.nnn.nnn.nnn to 239.nnn.nnn.nnn. Class E addresses range from 240.nnn.nnn.nnn to 255.nnn.nnn.nnn.
4. Each interface must have a unique internet address and cannot be the same as any defined TCP/IP interface internet address.

Top

Subnet mask (SUBNETMASK)

Specifies the subnet mask, which is a bit mask that defines the part of the network where this interface attaches. The mask is a 32-bit combination that is logically ANDed with the internet address to determine a particular subnetwork. The bits of the mask set to the value one (1) determine the network and subnetwork portions of the address. The bits set to the value zero (0) determine the host portion of the address.

This is a required parameter.

Note: The network portion must be equal to one bits in the subnetmask. The host portion and the subnetmask portion of an address must both be at least two bits wide.

The possible values are:

subnet-mask
Specify the mask for the network subnet field and host address field of the internet address that defines a subnetwork. The subnetwork mask is in the form, *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. The subnet mask must mask off all bits of the network class's network ID portion of the internet address. This means the subnet mask for a class A address must be 255.nnn.nnn.nnn, the subnet mask for a class B address must be 255.255.nnn.nnn, and the subnet mask for a class C address must be 255.255.255.nnn. For example, 255.255.255.0 could define a subnet mask for an interface with a class B internet address. In this example, the first two octets must be 1 bits because these octets define the network ID portion of the class B internet address. The third octet of this subnet mask defines the actual subnet mask ID portion of the interface's internet address. It is also all 1 bits. This leaves the fourth octet to define the host ID portion of the interface's internet address.

Note: The bits that identify the subnetwork are not required to be adjacent in the address. However, it is strongly recommended that the subnet bits be contiguous and located in the most significant bits of the host address.

Note: If the subnet mask is entered from a command line, the address must be enclosed in apostrophes.

Restriction: The subnet mask cannot be 255.255.255.255 for a class A, class B, or class C interface internet address.

Top

Examples

```
ADDIPSIFC   INTNETADR('9.5.1.248')  SUBNETMASK('255.255.255.0')
```

This command designates an IP address for this local host on the SNA transport which is 9.5.1.248. The SUBNETMASK indicates that network 9 is subnetted using bytes 2 and 3 of the internet address as the subnetwork.

Top

Error messages

***ESCAPE Messages**

CPFA108
IP over SNA interface added but not started.

TCP8050
*IOSYSCFG authority required to use &1.

TCP9999
Internal system error in program &1.

Top

Add IP over SNA Location (ADDIPSLOC)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add IP over SNA Location Entry (ADDIPSLOC) command is used to define AF_INET sockets over SNA location mapping entries. AF_INET sockets over SNA requires that an SNA location (network identifier/location name) be defined for each IP address that can be reached on an SNA transport. The location mapping entries define the SNA location for each IP address.

The SNA locations can be identified in one of two ways:

- A single host (or IP address) is specified with a single SNA location (network identifier/location name).
- A group of hosts designated by the network ID portion of the IP address is specified with a given SNA network identifier and a location name template.

Restriction: The user must have *IOSYSCFG authority to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
RMTDEST	Remote destination	*Character value*	Required, Positional 1
SUBNETMASK	Subnet mask	*Character value*, *HOST	Required, Positional 2
RMTNETID	Remote network identifier	*Communications name*, ***NETATR**	Optional
LOCTPL	Location template	*Character value*	Optional

Top

Remote destination (RMTDEST)

Specifies the remote network, subnetwork or host associated with this location entry. You must specify all four bytes that make up an internet address though some of the bytes may be equal to 0. For example, a remote route destination to all the hosts on the 9.5.11 subnetwork is identified by entering 9.5.11.0 for the remote route destination. Used in combination with a subnet mask value, the remote route destination will identify a remote network or system.

The remote route destination can be specified in the form nnn.0.0.0 for class A, nnn.nnn.0.0 for class B, and nnn.nnn.nnn.0 for class C, or nnn.nnn.nnn.nnn for any combination thereof, where nnn is a decimal number ranging from 0 through 255. Any combination thereof means that you may specify remote route destination such as 9.5.0.0 to the hosts on the 9.5 subnet, even though all 9.5.x.x addresses are class A network addresses.

This is a required parameter.

Restrictions:

1. The remote route destination cannot start with a zero (0); for example, 0.nnn.nnn.nnn.

2. The remote route destination cannot start with 127; for example, 127.nnn.nnn.nnn. This address range is reserved for TCP/IP loopback addresses.
3. The remote route destination cannot be a class D or class E address. Class D addresses range from 224.nnn.nnn.nnn to 239.nnn.nnn.nnn. Class E addresses range from 240.nnn.nnn.nnn to 255.nnn.nnn.nnn.
4. You cannot specify a remote route destination of 255.255.255.255. This is the limited broadcast address.
5. You cannot specify a directed broadcast address for the remote route destination; for example, nnn.255.255.255 for class A, nnn.nnn.255.255 for class B, and nnn.nnn.nnn.255 for class C.
6. For a single host remote route destination, all bits in the host portion of the IP address cannot be zero (0).

Top

Subnet mask (SUBNETMASK)

Specifies a bit mask that identifies to AF_INET sockets over SNA which bits of the value specified for the remote route destination (RMTDEST) compose the network and subnet portions of the internet address. By defining the network portion and subnetwork portion of the RMTDEST address, the subnet mask also defines which bits of the RMTDEST address make up the host portion. The mask is a 32-bit combination that is logically ANDed with the internet address to determine a particular subnetwork. The bits of the mask set to the value one (1) determine the network and subnetwork portions of the address. The bits set to the value zero (0) determine the host portion of the address.

This is a required parameter.

Note: The network portion must be equal to one bits in the subnetmask. The host portion and the subnetmask portion of an address must both be at least two bits wide.

The possible values are:

***HOST**
The internet address value specified in the remote route destination field is a host address. The subnet mask value is calculated to be 255.255.255.255.

subnet-mask
Specify the mask for the network subnet field and host address field of the internet address that defines a subnetwork. The subnetwork mask is in the form, *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. The subnet mask must mask off all bits of the network class's network ID portion of the internet address. This means the subnet mask for a class A address must be 255.nnn.nnn.nnn, the subnet mask for a class B address must be 255.255.nnn.nnn, and the subnet mask for a class C address must be 255.255.255.nnn.

For example, 255.255.255.0 could define a subnet mask for an interface with a class B internet address. In this example, the first two octets must be 1 bits because these octets define the network ID portion of the class B internet address. The third octet of this subnet mask defines the actual subnet mask ID portion of the interface's internet address. It is also all 1 bits. This leaves the fourth octet to define the host ID portion of the interface's internet address.

For example, a remote route destination's internet address value of 129.35.192.0 identifies a class B subnetwork. The network ID part of its address is 129.35. The portion of the subnetmask that is associated with the network portion of a particular class of address must equal 255. Therefore, the upper 2 bytes must be equal to 255.255 in the subnetmask. The subnetmask in this example may be 255.255.192.0 if the third octet is used as the subnetwork ID portion of the internet address.

Top

Remote network identifier (RMTNETID)

Specifies the name of the remote SNA network associated with this IP network or IP address.

The possible values are:

***NETATR**
The remote network identifier specified in the network attributes is used.

remote-sna-network-identifier
Specify the remote network identifier. This identifier can be one to eight characters in length. The first character must be A (or a) through Z (or z), or special characters $, #, or @ followed by 0 through 9, A (or a) through Z (or z), $, #, or @.

Top

Location template (LOCTPL)

Specifies the SNA location names associated with the IP network or subnetwork specified by the remote route destination or a single location name if the remote route destination address is for a single host.

This is a required parameter.

The possible values are:

location-name-template
Specify an 8 character template to be used by the system for generating remote location names based on the remote IP address specified on socket system calls. The first character must be A (or a) through Z (or z), or special characters $, #, or @ followed by 0 through 9, A (or a) through Z (or z), $, #, @, or ?. The template must specify some of the characters for the location name. The system generates the remaining characters based on the class of the IP address.

System-generated location name characters are identified by a question mark (?) character. Each question mark represents a single character that is generated by the system. A question mark may be used anywhere within the location name template except in the first character position. For example, the following location name templates are valid:

- ABCD????
- AB??CD??
- A?B?C?D?

Notes:

1. The number of question mark (?) characters in the template is dependent on how the remote destination (RMTDEST) parameter is used:
   - If the internet address specified for the RMTDEST is for a single host, the SUBNETMASK value must be *HOST or 255.255.255.255 and the LOCTPL value must not contain any question mark (?) characters.
   - If the internet address specified for the RMTDEST is for a network or subnetwork, the LOCTPL value must be an 8 character template containing a minimum number of question mark (?) characters based on the number of host mask bits contained in the SUBNETMASK parameter value. The minimum number of question mark characters is determined by dividing the number of host mask bits in the SUBNETMASK value by 5 and rounding up to the next whole number. For example: If the SUBNETMASK is 255.255.255.128, there are 7 host mask bits. In this case there must be 2 question mark characters in the template.
2. If the RMTDEST is for a group of hosts, a location name template must be specified. A single location name will not work.

location-name

Specify the remote location name. This name can be one to eight characters in length. The first character must be A (or a) through Z (or z), or special characters $, #, or @ followed by 0 through 9, A (or a) through Z (or z), $, #, or @.

If the RMTDEST is for a single host, a single location name must be specified. A location template will not work.

Top

Examples

Example 1: Adding an AF_INET Sockets over SNA Location Entry

```
ADDIPSLOC   RMTDEST('128.2.0.0')  SUBNETMASK('255.255.255.128')
            LOCTPL('ABCD????')
```

This command adds an AF_INET sockets over SNA location entry for a subnetwork with network 128.2 and subnet mask of 255.255.255.128. Remote IP addresses for subnetwork 128.2 specified on socket system calls are algorithmically mapped into SNA names that use the SNA network identifier specified in the network attributes and location names which start with ABCD. The system creates the remaining four characters of the location name based on the IP address. See the Convert IP over SNA Interface (CVTIPSIFC) command for more information.

Example 2: Adding an AF_INET Sockets over SNA Location Entry for a Host

```
ADDIPSLOC   RMTDEST('128.2.3.4')  SUBNETMASK(*HOST)  LOCTPL(XYZ00001)
```

This command adds an AF_INET sockets over SNA location entry for a host at IP address 128.2.3.4. The address 128.2.3.4 is mapped to the SNA location name of XYZ00001 and uses the default SNA network identifier specified in the network attributes.

Top

Error messages

None

Top

Add IP over SNA Route (ADDIPSRTE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add IP over SNA Route (ADDIPSRTE) command is used to identify a route to a remote network or a route to a remote destination system in the AF_INET sockets over SNA configuration.

Restrictions:

1. The user must have *IOSYSCFG authority to use this command.
2. A route cannot be added unless the internet address of the gateway system specified by the NEXTHOP parameter can be reached directly through a network associated with a previously defined AF_INET sockets over SNA interface. An interface can be added using the Add IP over SNA Interface (ADDIPSIFC) CL command.

Top

Parameters

Keyword	Description	Choices	Notes
RTEDEST	Route destination	*Character value*	Required, Positional 1
SUBNETMASK	Subnet mask	*Character value*, *HOST	Required, Positional 2
NEXTHOP	Next hop	*Character value*	Required, Positional 3

Top

Route destination (RTEDEST)

Specifies the route destination being added. You must specify all 4 bytes that make up an internet address though some of the bytes may be equal to 0. For example, a route to all the hosts on the 9.5.11 subnetwork is identified by entering 9.5.11.0 for the route destination. Used in combination with a subnet mask and next hop, the route destination uniquely identifies a route to a network or system.

route-destination: Specify the route destination being added. The route destination can be specified in the form *nnn.0.0.0*, for Class A, *nnn.nnn.0.0* for Class B, and *nnn.nnn.nnn.0* for Class C, or *nnn.nnn.nnn.nnn* for any combination thereof, where *nnn* is a decimal number ranging from 0 through 255.

Any combination thereof means that you may specify a route, such as 9.5.0.0 to the hosts on the 9.5 subnet, even though all 9.5.x.x addresses are class A network addresses.

Restrictions:

1. The route destination cannot start with a zero (0); for example, 0.nnn.nnn.nnn.
2. The route destination cannot start with 127; for example, 127.nnn.nnn.nnn. This address range is reserved for TCP/IP loopback addresses.

3. The route destination cannot be a class D or class E address. Class D addresses range from 224.nnn.nnn.nnn to 239.nnn.nnn.nnn. Class E addresses range from 240.nnn.nnn.nnn to 255.nnn.nnn.nnn.
4. You cannot specify a route destination of 255.255.255.255.
5. You cannot specify a directed broadcast address for the route destination; for example, nnn.255.255.255 for class A, nnn.nnn.255.255 for class B, and nnn.nnn.nnn.255 for class C.
6. For a single host route destination, all bits in the host portion of the IP address cannot be zero (0).
7. For a single host route destination, an interface cannot exist with the same internet address as the RTEDEST internet address.

This is a required parameter.

Top

Subnet mask (SUBNETMASK)

Specifies a bit mask that identifies to AF_INET sockets over SNA which bits of the value specified for the route destination (RTEDEST) compose the network and subnet portions of the internet address. By defining the network portion and subnetwork portion of the RTEDEST address, the subnet mask also defines which bits of the RTEDEST address make up the host portion. The mask is a 32-bit combination that is logically ANDed with the internet address to determine a particular subnetwork. The bits of the mask set to the value one (1) determine the network and subnetwork portions of the address. The bits set to the value zero (0) determine the host portion of the address.

This is a required parameter.

The possible values are:

***HOST**
Specify this value when the internet address value specified in the route destination field is a host address. The subnet mask value is calculated to be 255.255.255.255.

subnet-mask
Specify the mask for the network subnet field and host address field of the internet address that defines a subnetwork. The subnetwork mask is in the form, *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. The subnet mask must mask off all bits of the network class's network ID portion of the internet address. This means the subnet mask for a class A address must be 255.nnn.nnn.nnn, the subnet mask for a class B address must be 255.255.nnn.nnn, and the subnet mask for a class C address must be 255.255.255.nnn.

For example, a destination route's internet address value of 129.35.192.0 identifies a class B subnetwork. The network ID part of its address is 129.35. The portion of the subnetmask that is associated with the network portion of a particular class of address must equal 255. Therefore, the upper 2 bytes must be equal to 255.255 in the subnetmask. The subnetmask in this example may be 255.255.192.0 if the third octet is used as the subnetwork ID portion of the internet address.

Top

Next hop (NEXTHOP)

Specifies the internet address of the next system (gateway) on the route. A route cannot be added unless the internet address specified by the NEXTHOP parameter can be reached directly through a network associated with a previously defined AF_INET sockets over SNA interface. An interface can be added by using the Add IP over SNA Interface (ADDIPSIFC) CL command.

The possible values are:

internet-address
Specify the internet address. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Restrictions:

1. The next hop internet address cannot begin with 0 or 127 (for example, 0.nnn.nnn.nnn).
2. The next hop internet address cannot be a class D or class E address. Class D addresses range from 224.nnn.nnn.nnn to 239.nnn.nnn.nnn. Class E addresses range from 240.nnn.nnn.nnn to 255.nnn.nnn.nnn.

Top

Examples

Example 1: Adding a Network Route on the Network

```
ADDIPSRTE   RTEDEST('128.2.0.0')  SUBNETMASK('255.255.0.0')
            NEXTHOP('9.2.3.4')
```

This command defines a network route for all remote hosts on the network 128.2. Network 128.2 is not subnetted since the first two octets of a class B internet address are the network ID portion of the address and the subnet mask is only masking off the first two octets. The gateway specified by NEXTHOP must be in the same network or subnetwork as one of the AF_INET sockets of SNA interfaces that has already been defined on the local host.

Example 2: Adding a Network Route on the Subnetwork

```
ADDIPSRTE   RTEDEST('129.1.1.0')  SUBNETMASK'(255.255.255.0')
            NEXTHOP('128.3.4.5')
```

This command defines a network route for all remote hosts on the subnetwork 129.1.1. Network 129.1 is subnetted, with the subnet portion of the IP address contained in byte 3. For this example assume that subnet mask 129.1.1 is directly accessible only through the AF_INET Sockets over SNA interface 128.3.4.5 that has already been defined on the local host. Since the network id portion of interface 128.3.4.5 is not the same as the route destination's network id, we need to specify that the NEXTHOP is the local interface 128.3.4.5. This tells AF_INET Sockets over SNA to use local interface 128.3.4.5 to get to subnetwork 129.1.1.

Top

Error messages

***ESCAPE Messages**

TCP2665
&2 &1 not added successfully.

TCP2666
&2 &1 not added.

TCP8050
*IOSYSCFG authority required to use &1.

TCP9999
Internal system error in program &1.

Top

Add Job Queue Entry (ADDJOBQE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Job Queue Entry (ADDJOBQE) command adds a job queue entry to the specified subsystem description. A job queue entry identifies a job queue from which jobs are selected for running in the subsystem. Jobs can be placed on a job queue by spooling readers or by using the following commands:

- Submit Job (SBMJOB)
- Submit Data Base Jobs (SBMDBJOB)
- Submit Diskette Jobs (SBMDKTJOB)
- Transfer Job (TFRJOB)
- Transfer Batch Job (TFRBCHJOB)

Jobs started from a job queue are batch jobs (except for interactive jobs that use the TFRJOB command). In a subsystem, job queues with lower sequence numbers are processed first. For more information, refer to the **Sequence number (SEQNBR)** parameter.

Restrictions:

1. To use this command, you must have object operational (*OBJOPR), object management (*OBJMGT), and read (*READ) authority to the specified subsystem description and execute (*EXECUTE) authority to the library containing that subsystem description.

Top

Parameters

Keyword	Description	Choices	Notes
SBSD	Subsystem description	*Qualified object name*	Required, Positional 1
	Qualifier 1: Subsystem description	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
JOBQ	Job queue	*Qualified object name*	Required, Positional 2
	Qualifier 1: Job queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MAXACT	Maximum active jobs	0-1000, **1**, *NOMAX	Optional
SEQNBR	Sequence number	1-9999, **10**	Optional
MAXPTY1	Max active priority 1	0-99, ***NOMAX**	Optional
MAXPTY2	Max active priority 2	0-99, ***NOMAX**	Optional
MAXPTY3	Max active priority 3	0-99, ***NOMAX**	Optional
MAXPTY4	Max active priority 4	0-99, ***NOMAX**	Optional
MAXPTY5	Max active priority 5	0-99, ***NOMAX**	Optional
MAXPTY6	Max active priority 6	0-99, ***NOMAX**	Optional
MAXPTY7	Max active priority 7	0-99, ***NOMAX**	Optional
MAXPTY8	Max active priority 8	0-99, ***NOMAX**	Optional
MAXPTY9	Max active priority 9	0-99, ***NOMAX**	Optional

Top

Subsystem description (SBSD)

Specifies the name and library of the subsystem description to which the job queue entry is added.

This is a required parameter.

Qualifier 1: Subsystem description

name Specify the name of the subsystem description where the job queue entry is being added.

Note: The IBM-supplied object QSYSSBSD is not valid on this parameter.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the library where the subsystem description is located.

Top

Job queue (JOBQ)

Specifies the name and library of the job queue that is a source of batch jobs that are started by the subsystem. If the job queue does not exist when the entry is added or changed, a library qualifier other than *LIBL must be specified because the job queue name is kept in the subsystem description.

This is a required parameter.

Qualifier 1: Job queue

name Specifies the name of the job queue that is a source of batch jobs that are started by the subsystem.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the library where the job queue is located.

Top

Maximum active jobs (MAXACT)

Specifies the maximum number of jobs that can be active at the same time from this job queue.

1 Only one job from the job queue can be active at any time.

***NOMAX**
There is no maximum for the number of jobs that can be started at the same time through this

job queue entry. However, the maximum activity level of the routing entries, the subsystem, or the specific job priority level might prevent jobs from being started. If *NOMAX is specified, all the jobs on the job queue are started, even though the activity level of the storage pool being used might prohibit them from running at the same time.

0-1000 Specify the maximum number of jobs that can be active at the same time from this job queue.

Top

Sequence number (SEQNBR)

Specifies a sequence number for this job queue, which is used by the subsystem to determine the order in which the job queues are processed.

The subsystem first selects jobs from the job queue with the lowest sequence number. When all jobs on that queue have been processed or the number of jobs specified on the **Maximum active jobs (MAXACT)** parameter has been reached, the subsystem processes jobs on the queue with the next higher sequence number. This sequence continues until all job queue entries have been processed, or until the subsystem has reached its limit for overall maximum jobs. In some cases, the sequence is interrupted and the subsystem processes a job queue with a lower sequence number. This occurs for this subsystem when one of the following conditions occurs:

- A held job or job queue is released.
- A job is placed on or transferred to a job queue.
- A new job queue is allocated.
- A job ends.

10 A sequence number of 10 is assigned to this job queue.

1-9999 Specify the sequence number assigned to this job queue. The sequence number must be unique in the subsystem description.

Top

Max active priority 1 (MAXPTY1)

Specifies the number of jobs that can be started for a specified job priority level.

***NOMAX**
The number of jobs started in a specific priority level is not limited.

0-99 Specify the number of jobs started in a specific priority level. A value of 0 indicates that no jobs are started from a specific priority level.

Top

Max active priority 2 (MAXPTY2)

Specifies the number of jobs that can be started for a specified job priority level.

***NOMAX**
The number of jobs started in a specific priority level is not limited.

0-99 Specify the number of jobs started in a specific priority level. A value of 0 indicates that no jobs are started from a specific priority level.

Top

Max active priority 3 (MAXPTY3)

Specifies the number of jobs that can be started for a specified job priority level.

***NOMAX**
The number of jobs started in a specific priority level is not limited.

0-99 Specify the number of jobs started in a specific priority level. A value of 0 indicates that no jobs are started from a specific priority level.

Top

Max active priority 4 (MAXPTY4)

Specifies the number of jobs that can be started for a specified job priority level.

***NOMAX**
The number of jobs started in a specific priority level is not limited.

0-99 Specify the number of jobs started in a specific priority level. A value of 0 indicates that no jobs are started from a specific priority level.

Top

Max active priority 5 (MAXPTY5)

Specifies the number of jobs that can be started for a specified job priority level.

***NOMAX**
The number of jobs started in a specific priority level is not limited.

0-99 Specify the number of jobs started in a specific priority level. A value of 0 indicates that no jobs are started from a specific priority level.

Top

Max active priority 6 (MAXPTY6)

Specifies the number of jobs that can be started for a specified job priority level.

***NOMAX**
The number of jobs started in a specific priority level is not limited.

0-99 Specify the number of jobs started in a specific priority level. A value of 0 indicates that no jobs are started from a specific priority level.

Top

Max active priority 7 (MAXPTY7)

Specifies the number of jobs that can be started for a specified job priority level.

***NOMAX**
The number of jobs started in a specific priority level is not limited.

0-99 Specify the number of jobs started in a specific priority level. A value of 0 indicates that no jobs are started from a specific priority level.

Top

Max active priority 8 (MAXPTY8)

Specifies the number of jobs that can be started for a specified job priority level.

***NOMAX**
The number of jobs started in a specific priority level is not limited.

0-99 Specify the number of jobs started in a specific priority level. A value of 0 indicates that no jobs are started from a specific priority level.

Top

Max active priority 9 (MAXPTY9)

Specifies the number of jobs that can be started for a specified job priority level.

***NOMAX**
The number of jobs started in a specific priority level is not limited.

0-99 Specify the number of jobs started in a specific priority level. A value of 0 indicates that no jobs are started from a specific priority level.

Top

Examples

Example 1: Adding a Job Queue

```
ADDJOBQE   SBSD(QGPL/NIGHTSBS)  JOBQ(QGPL/NIGHT)  MAXACT(3)
```

This command adds a job queue entry for the NIGHT job queue (in the QGPL library) to the NIGHTSBS subsystem description contained in the QGPL library. The entry specifies that up to three batch jobs from the NIGHT job queue can be active at the same time in the subsystem. The default sequence number of 10 is assumed.

Example 2: Running Jobs in Specific Priority Levels

```
ADDJOBQE   SBSD(QBASE)  JOBQ(JOBQ1)  MAXPTY1(2)  MAXPTY7(0)
           MAXPTY8(0)  MAXPTY9(0) SEQNBR(99)
```

This command adds a job queue entry for the JOBQ1 job queue to the QBASE subsystem description. The sequence number for this job queue is set to 99. The priority levels specified prevent any jobs with priority levels 7 through 9 from running.

Top

Error messages

***ESCAPE Messages**

CPF1619
Subsystem description &1 in library &2 damaged.

CPF1691

Active subsystem description may or may not have changed.

CPF1697

Subsystem description &1 not changed.

Top

Add Job Schedule Entry (ADDJOBSCDE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Job Schedule Entry (ADDJOBSCDE) command allows you to schedule batch jobs by adding an entry to the job schedule. You can use this command to schedule a batch job to be submitted once, or to schedule a batch job to be submitted at regular intervals.

The job schedule entry contains all of the information needed to submit the job, including the command the job runs, the job description and user profile under which the job is run, the job queue to which the job is submitted, and the message queue to which messages are sent.

At the date and time you specify on this command, the job is submitted to the specified job queue. This command does not guarantee that the job will begin running at the scheduled time, however. The job will not begin running if the job queue is held or not attached to an active subsystem, or if the maximum number of active jobs allowed to run in the subsystem or on the system at one time has been reached.

Each job schedule entry is identified by the job name specified for the JOB parameter of this command and an entry number assigned by the system when the entry is added. The message replacement text sent when an entry is added contains the entry number. If there is more than one entry with the same job name, you may need to specify the number when changing the entry using the Change Job Schedule Entry (CHGJOBSCDE) command, removing the entry using the Remove Job Schedule Entry (RMVJOBSCDE) command, or when holding or releasing the entry using the Hold Job Schedule Entry (HLDJOBSCDE) or Release Job Schedule Entry (RLSJOBSCDE) command. You can use the Work with Job Schedule Entries (WRKJOBSCDE) command to show or print entries.

More information is in the Work management topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Restrictions:

1. To use this command, you must have:
   - use (*USE) authority to the command specified by the **Command to run (CMD)** parameter and execute (*EXECUTE) authority to the library containing that command.
   - use (*USE) authority to the job description (JOBD) and execute (*EXECUTE) authority to the library containing that job description.
   - use (*USE) authority to the job queue (JOBQ) and execute (*EXECUTE) authority to the library containing that job queue. Authority to the job queue cannot come from adopted authority.
   - use (*USE) authority to the user profile (USER) that the job is to run under.
   - use (*USE) and add (*ADD) authority to the message queue (MSGQ), and execute (*EXECUTE) authority to the library containing that message queue. Authority to the message queue cannot come from adopted authority.
   - change (*CHANGE) authority to object QDFTJOBSCD, type *JOBSCD, in library QUSRSYS and execute (*EXECUTE) authority to library QUSRSYS.

Top

Parameters

Keyword	Description	Choices	Notes
JOB	Job name	*Name*, *JOBD	Required, Positional 1
CMD	Command to run	*Command string*	Required, Positional 2
FRQ	Frequency	*ONCE, *WEEKLY, *MONTHLY	Required, Positional 3
SCDDATE	Schedule date	*Date*, ***CURRENT**, *MONTHSTR, *MONTHEND, *NONE	Optional
SCDDAY	Schedule day	Single values: ***NONE**, *ALL Other values (up to 7 repetitions): *MON, *TUE, *WED, *THU, *FRI, *SAT, *SUN	Optional
SCDTIME	Schedule time	*Time*, ***CURRENT**	Optional
RELDAYMON	Relative day of month	Values (up to 5 repetitions): *LAST, **1**, 2, 3, 4, 5	Optional
SAVE	Save	***NO**, *YES	Optional
OMITDATE	Omit date	Values (up to 20 repetitions): *Date*, ***NONE**	Optional
RCYACN	Recovery action	***SBMRLS**, *SBMHLD, *NOSBM	Optional
JOBD	Job description	Single values: ***USRPRF** Other values: *Qualified object name*	Optional
	Qualifier 1: Job description	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
JOBQ	Job queue	Single values: ***JOBD** Other values: *Qualified object name*	Optional
	Qualifier 1: Job queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
USER	User	*Name*, ***CURRENT**, *JOBD	Optional
MSGQ	Message queue	Single values: ***USRPRF**, *NONE Other values: *Qualified object name*	Optional
	Qualifier 1: Message queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Top

Job name (JOB)

Specifies the name of the job schedule entry.

This is a required parameter.

***JOBD**
The job description specified for the JOBD parameter is used for the name of the job schedule entry.

name Specify the name of the job schedule entry.

Note: To avoid deleting, holding, or releasing entries created by IBM products when you are using generic names to delete, hold, or release your entries, do not add entries with job names beginning with the letter Q.

Top

Command to run (CMD)

Specifies the command that runs in the submitted job. Because the command you specify is used for the request data, the value specified for the RQSDTA parameter in the job description is ignored. The command you specify is syntax-checked when the entry is added.

This is a required parameter.

command-string
Specify a maximum of 512 characters.

Top

Frequency (FRQ)

Specifies how often the job is submitted.

This is a required parameter.

***ONCE**
The job is submitted once.

***WEEKLY**
The job is submitted on the same day or days of each week at the scheduled time.

***MONTHLY**
The job is submitted on the same day or days of each month at the scheduled time.

If you specify *MONTHLY and a month does not contain the day specified for the SCDDATE parameter, the job is not run that month. For example, if SCDDATE(01/31/2004) and FRQ(*MONTHLY) are specified, the job is submitted on 01/31, 03/31, 5/31, 7/31, 8/31, 10/31 and 12/31, but will not run in February, April, June, September, or November. To submit a job on the last day of every month, specify SCDDATE(*MONTHEND).

If you specify *MONTHLY and your system or your job is configured to use Julian date format, the job is submitted to run on the day of the month that it would run if the system or job did not use Julian date format.

Top

Schedule date (SCDDATE)

Specifies the date on which the job is submitted.

If your system or your job is configured to use the Julian date format, *MONTHSTR and *MONTHEND are calculated as if the system or job did not use the Julian date format.

The SCDDATE and SCDDAY parameters are mutually exclusive.

***CURRENT**
The job is submitted on the current date.

***MONTHSTR**
The job is submitted on the first day of the month. If you specify *MONTHSTR, and if today is the first day of the month and the time you specify on the SCDTIME parameter has not passed, the job is submitted today. Otherwise, the job is submitted on the first day of the next month.

***MONTHEND**
The job is submitted on the last day of the month. If you specify *MONTHEND, and if today is

the last day of the month and the time you specify on the SCDTIME has not passed, the job is submitted today. Otherwise it is submitted on the last day of the next month.

NONE
No start date is specified. When SCDDATE(*NONE) is specified, a value other than *NONE must be specified for SCDDAY.

date Specify the date in the job date format.

Top

Schedule day (SCDDAY)

Specifies the day of the week on which the job is submitted.

The SCDDATE and SCDDAY parameters are mutually exclusive.

If today is the day of the week specified for this parameter and the time specified for the SCDTIME parameter has not passed, the job is submitted today. Otherwise, the job is submitted on the next occurrence of the specified day. For example, if SCDDAY(*FRI) and SCDTIME(12:00:00) are specified, and you are adding this job schedule entry at 11:00 a.m. on a Friday, the job is submitted today. If you are adding the entry at 4:00 p.m. on a Friday, or at 11:00 a.m. on a Monday, the job is submitted the following Friday.

Single values

***NONE**
No start day is specified. SCDDAY(*NONE) is not valid when SCDDATE(*NONE) is specified.

***ALL** The job is submitted every day.

Other values (up to 7 repetitions)

***MON**
The job is submitted on Monday.

***TUE** The job is submitted on Tuesday.

***WED** The job is submitted on Wednesday.

***THU** The job is submitted on Thursday.

***FRI** The job is submitted on Friday.

***SAT** The job is submitted on Saturday.

***SUN** The job is submitted on Sunday.

Top

Schedule time (SCDTIME)

Specifies the time on the scheduled date at which the job is submitted.

Note: Although the time can be specified to the second, the activity involved in submitting a job and the load on the system may affect the exact time at which the job is submitted.

***CURRENT**
The job is submitted at the current time. If you specify SCDTIME(*CURRENT) and SCDDATE(*CURRENT), the job is immediately submitted to the specified job queue.

time Specify the time you want the job to be submitted. The time is specified in 24-hour format and can be specified with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 to 23. Valid values for **mm** and **ss** range from 00 to 59.
- With a time separator, specify a string of 5 or 8 characters where the time separator specified for your job is used to separate the hours, minutes, and seconds. If this command is entered from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Top

Relative day of month (RELDAYMON)

Specifies the relative day of the month on which the job is submitted to run.

This parameter is valid only if a value is specified for the SCDDAY parameter and FRQ(*MONTHLY) is specified.

You can specify 5 values for this parameter.

1 The job is submitted on the specified day of the week the first time it occurs in the month. For example, if you specify SCDDAY(*TUE), FRQ(*MONTHLY) and RELDAYMON(1), the job is submitted on the first Tuesday of every month.

2 The job is submitted on the specified day of the week the second time it occurs in the month.

3 The job is submitted on the specified day of the week the third time it occurs in the month.

4 The job is submitted on the specified day of the week the fourth time it occurs in the month.

5 The job is submitted on the specified day of the week the fifth time it occurs in the month.

***LAST**
The job is submitted on the specified day of the week the last time it occurs in the month.

Top

Save (SAVE)

Specifies whether the entry for a job that is submitted only once is kept after the job is submitted. This parameter is valid only if FRQ(*ONCE) is specified.

***NO** The entry is not kept after the job is submitted.

***YES** The entry is kept after the job is submitted. If you specify *YES, the job is submitted once. The job is not submitted again until the Change Job Schedule Entry (CHGJOBSCDE) command is used to specify a new date and time.

Top

Omit date (OMITDATE)

Specifies a maximum of 20 dates on which the job is not submitted. You can, for example, use this parameter to prevent recurring jobs from running on holidays. The date must be specified in the job date format.

***NONE**
There are no dates when a job is not submitted.

date Specify a date when a job is not submitted.

Top

Recovery action (RCYACN)

Specifies the recovery action to be taken if the job cannot be submitted at the designated time because the system is powered down or in restricted state. The action specified for this parameter occurs at the next initial program load (IPL) or when the system comes out of restricted state.

Jobs submitted during IPL or when the system comes out of restricted state are submitted in the same order that they would have been had the jobs been submitted at the times specified in the job schedule entries. If multiple occurrences of a recurring job are missed, the job is submitted only once. The first missed occurrence of a recurring job is used to order the jobs. The next occurrence of the job is calculated from the current date.

Since the scheduler portion of IPL need not be complete for the IPL of the system to be complete, other jobs may start on the system before all of the jobs have been submitted.

This parameter does not apply:

- When a job is released after being held at the date and time it was to be submitted
- When the date and time at which a job is to be submitted passes because of changes to date and time system values

***SBMRLS**
The job is submitted in the released (RLS) state.

***SBMHLD**
The job is submitted in the held (HLD) state.

***NOSBM**
The job is not submitted.

Specifying *NOSBM affects only missed occurrences of the job. If the job schedule entry is a recurring job, future occurrences are not affected.

Top

Job description (JOBD)

Specifies the qualified name of the job description used when submitting the job.

Single values

***USRPRF**
The job description specified in the user profile under which the submitted job runs is used. The USER parameter specifies the user profile.

Qualifier 1: Job description

name Specify the name of the job description.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the name of the library where the job description is located.

Top

Job queue (JOBQ)

Specifies the qualified name of the job queue on which this job is placed.

You must have authority to the queue to specify a name on this parameter. Authority to the queue cannot come from adopted authority.

Single values

***JOBD**
The submitted job is placed on the job queue specified in the job description. The JOBD parameter specifies the job description.

Qualifier 1: Job queue

name Specify the name of the job queue.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the name of the library where the job queue is located.

Top

User (USER)

Specifies the name of the user profile under which the job is submitted.

***CURRENT**
The job is submitted under the user profile of the user that is adding the entry.

***JOBD**
The user profile specified for the USER parameter of the job description is used. The JOBD parameter of this command specifies the job description. This value is not valid if USER(*RQD) is specified in the job description.

name Specify the name of the user profile that is used. You must be authorized to the user profile. The user profile must be authorized to the job description and message queue specified for this command.

Top

Message queue (MSGQ)

Specifies the qualified name of the message queue to which messages are sent.

Messages are sent when the job is submitted and when a submitted job has completed running. Messages indicating a serious error are sent to the QSYSOPR message queue regardless of the value specified for this parameter when:

- The message queue specified for this parameter is damaged.
- MSGQ(*NONE) is specified.
- MSGQ(*USRPRF) and USER(*JOBD) are specified, and the job description specified for the JOBD parameter is changed to USER(*RQD) after the entry is added.

Single values

***USRPRF**
The message queue specified in the user profile under which the submitted job runs is used. The USER parameter specifies the user profile.

***NONE**
Completion messages are not sent. Error messages are sent to the QSYSOPR message queue.

Qualifier 1: Message queue

name Specify the name of the message queue.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the name of the library where the message queue is located.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the job schedule entry.

***BLANK**
No text is specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Examples

Example 1: Scheduling a Weekly Job

```
ADDJOBSCDE   JOB(CLEANUP)  SCDDATE(*NONE)
             CMD(CALL PGM(CLNUPLIB/CLNUPPGM))
             SCDDAY(*FRI)  SCDTIME('23:00:00')
             FRQ(*WEEKLY)  RCYACN(*NOSBM)
             JOBD(CLNUPLIB/CLNUPJOBD)
```

This command submits a job named CLEANUP every Friday at 11 p.m. The job uses job description CLNUPJOBD in library CLNUPLIB. If the system is powered down or is in the restricted state at 11 p.m. on Friday, the job is not submitted at IPL or when the system comes out of restricted state.

Example 2: Scheduling a Monthly Job

```
ADDJOBSCDE   JOB(PAYROLLJOB)  CMD(CALL PAYROLL)
             SCDDATE(*NONE)  SCDDAY(*MON)
             SCDTIME('09:00:00')  FRQ(*MONTHLY)
             RELDAYMON(1)
```

This command submits a job to run program PAYROLL at 9 a.m. on the first Monday of every month.

Example 3: Omitting Dates

```
ADDJOBSCDE   JOB(MONTHEND)  CMD(CALL INVENTORY)
             SCDDATE(*MONTHEND)  SCDTIME('23:30:00')
             FRQ(*MONTHLY)  OMITDATE('12/31/2003')
```

This command submits a job to run program INVENTORY at 11:30 p.m. on the last day of every month except December 31, 2003.

Example 4: Scheduling a Daily Job

```
ADDJOBSCDE   JOB(*JOBD)  CMD(CALL DAILYCLEAN)  SCDDATE(*NONE)
             SCDDAY(*ALL)  SCDTIME('18:00:00')
             FRQ(*WEEKLY)  RCYACN(*NOSBM)  USER(SOMEPGMER)
```

This command submits a job to run program DAILYCLEAN every day at 6 p.m. The job runs under user profile SOMEPGMER. If the system is powered down or is in the restricted state at 6 p.m., the job is not submitted at IPL or when the system comes out of restricted state.

Example 5: Scheduling a Weekly Job

```
ADDJOBSCDE   JOB(*JOBD)  CMD(CALL PGM1)  SCDDATE('06/01/2003')
             FRQ(*WEEKLY)  USER(PGMR1)
```

This command submits a job to run program PGM1 every week starting on June 1, 2003 at the current time. Because June 1 is a Saturday, the job is submitted every Saturday.

Example 6: Scheduling a Job to Run Twice a Month

```
ADDJOBSCDE   JOB(*JOBD)  CMD(CALL PGM2)
             SCDDATE(*NONE)  SCDDAY(*MON *WED)
             FRQ(*MONTHLY)  RELDAYMON(3)
             SCDTIME('23:30:00')
```

This command submits a job to run program PGM2 every third Monday and every third Wednesday at 11:30 p.m. The job is submitted this month if the third Monday and Wednesday have not passed when this entry is added. If, for example, yesterday was the third Monday, today is the third Tuesday, and tomorrow is the third Wednesday, the job is submitted tomorrow, and then not again until next month.

Example 7: Scheduling a Job to Run Twice a Month

```
ADDJOBSCDE  JOB(*JOBD)  CMD(CALL PGM3)
            SCDDATE(*NONE)  SCDDAY(*MON)
            FRQ(*MONTHLY)  RELDAYMON(1 3)
            SCDTIME('09:00:00')
            USER(PGMR3)
```

This command submits a job to run program PGM3 on the 1st and 3rd Monday of every month at 9:00 a.m. The job runs under user profile PGMR3.

Example 8: Scheduling a Job to Run Every Weekday

```
ADDJOBSCDE   JOB(*JOBD)  CMD(CALL PGM4)
             SCDDATE(*NONE) SCDDAY(*MON *TUE *WED *THU *FRI)
             SCDTIME('19:00:00') FRQ(*WEEKLY)
```

This command submits a job to run program PGM4 every weekday at 7 p.m.

Top

Error messages

***ESCAPE Messages**

CPF1633
Job schedule entry &3 number &4 not added.

CPF9872
Program or service program &1 in library &2 ended. Reason code &3.

Top

Add Job Watcher Definition (ADDJWDFN)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Job Watcher Definition (ADDJWDFN) command adds a new Job Watcher definition to the system. A Job Watcher definition identifies the performance data that is to be collected during a Job Watcher collection. A session can be started using the Start Job Watcher (STRJW) command. A definition is required when starting a new Job Watcher collection.

Restrictions:

- To use this command, you must have service (*SERVICE) special authority, or be authorized to the Job Watcher function of the Operating System through System i5 Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_JOB_WATCHER, can also be used to change the list of users that are allowed to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
DFN	Definition	*Name*	Required, Positional 1
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional
COLITV	Collection interval	0.1-3600.0, **10**, *NODELAY	Optional
ADDDTACGY	Additional data categories	Single values: ***NONE**, *ALL Other values (up to 8 repetitions): *Element list*	Optional
	Element 1: Category	*ACTGRPDTL, *ACTGRPSUM, *SQLSTMT, *SQLCURSTMT, *SQLDETAIL, *CALLSTACK, *SOCKETTCP, *SOCKETJOBS, *JAVA, *JAVASTACK	
	Element 2: Interval frequency	*Integer*, ***ALWAYS**	
WAITSTK	Wait-based call stack data	Single values: ***NONE** Other values (up to 2 repetitions): *Element list*	Optional
	Element 1: Category	*CONFLICT, *ABNWAIT	
	Element 2: Minimum duration (microsecs)	*Integer*, **10000**	
JOB	Job name	Single values: ***ALL**, *NONE Other values (up to 20 repetitions): *Qualified job name*	Optional
	Qualifier 1: Job name	*Generic name, name*	
	Qualifier 2: User	*Generic name, name*, *ALL	
	Qualifier 3: Number	000000-999999, *ALL	
TASKNAME	Task name	Single values: ***ALL**, *NONE Other values (up to 20 repetitions): *Character value*	Optional
TDENBR	TDE number	Values (up to 20 repetitions): *Hexadecimal value*	Optional
CURUSRPRF	Current user profile	Values (up to 20 repetitions): *Name*	Optional
SBS	Subsystem	Values (up to 20 repetitions): *Name*	Optional
CURPOOL	Current storage pool	Values (up to 20 repetitions): 1-64	Optional

Keyword	Description	Choices	Notes
FRCRCD	Force record write	***ITVEND**, *CALC	Optional
INCALLFST	Include inactive jobs/tasks	***NO**, *YES	Optional
TOASPTHLD	To file ASP threshold	1-99, ***SYSTEM**	Optional
SYSASPTHLD	System ASP threshold	1-99, ***SYSTEM**	Optional
CONDCTLF	Condition control file	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Condition control file	*Name*	
	Qualifier 2: Library	*Name*	
CONDCTLMBR	Condition control member	*Name*, ***FIRST**	Optional
CONDTYPE	Condition type	***PERITV**, *TRIGGER, *UNTILMET	Optional
TIMEOUT	Timeout option	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Option	*NBRSEC, *NBRITV	
	Element 2: Value	*Integer*	
OCCURS	Consecutive occurrence count	*Integer*, **1**	Optional
HSTSIZE	History size	*Integer*, **0**	Optional
EXITPGM	User exit program	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: User exit program	*Name*	
	Qualifier 2: Library	*Name*	
EXITPGMDTA	Exit program data	*Character value*, ***NONE**	Optional

Top

Definition (DFN)

Specifies the name of the Job Watcher definition being added. If the specified definition already exists an error condition will occur. This is a required parameter.

name Specify the name of the new Job Watcher definition.

Top

Text 'description' (TEXT)

The text description of the Job Watcher definition to be added.

***BLANK**
The Job Watcher definition will have no text description.

character-value
Specify a text description for the Job Watcher definition. The description should be no more that 50 characters of text, enclosed in apostrophes.

Top

Collection interval (COLITV)

Specifies the interval between retrieval of job/task data. Job/task data is collected from the system on a sampling basis. This value specifies the amount of time that will elapse between the collection of each sample.

10 The delay between the collection of interval data will be 10 seconds.

***NODELAY**
Data will be collected as fast as possible, with no delay between the collection of interval data.

0.1-3600.0
Specify the number of seconds to delay between the collection of interval data. If a value is specified on this parameter it will override the value in the Job Watcher definition.

Top

Additional data categories (ADDDTACGY)

Specifies additional types of data to include in the collection. Data types selected on this parameter will be collected in addition to the standard data collected by the Job Watcher function.

Single values

***NONE**
Only standard data will be included in this collection. Standard data collected includes: run information (QAPYJWRUNI), interval information (QAPYJWINTI), TDE and process data (QAPYJWSTS, QAPYJWTDE, and QAPYJWPRC), wait bucket mapping (QAPYJWBKT), Classic Java virtual machine data (QAPYJWJVM), and Classic Java thread data (QAPYJWJVTH).

***ALL** This collection will include standard data and all optional data categories.

Other values (up to 10 repetitions)

Element 1: Category

Specifies the additional types of data to collect.

***ACTGRPDTL**
This data type will collect detailed activation group data. This data type will write activation group data to the database files QAPYJWAIGP, QAPYJWAIHP, and QAPYJWAIPA.

***ACTGRPSUM**
This data type will collect summary data for activation groups. This summary data will be written to the file QAPYJWPRC in the fields CURNUMACTG and CURNUMACT.

***SQLCURSTMT**
This data type will collect ″in progress″ SQL statement and host variable data. SQL current statement data will be written to the files QAPYJWSQL and QAPYJWSQLH.

***SQLSTMT**
This data type will collect ″in progress″ or ″last completed″ SQL statement and host variable data. SQL statement data will be written to the files QAPYJWSQL and QAPYJWSQLH.

***SQLDTL**
This data type will collect detailed SQL data which includes ″in progress″ or ″last completed″ SQL statement, host variable, open cursor list, and prepared statement area data. Detailed SQL data will be written to the files QAPYJWSQL, QAPYJWSQLH, QAPYJWSQLO, and QAPYJWSQLP.

***CALLSTK**
This data type will collect call stacks for each TDE included in the collection. Call stack data will be written to the files QAPYJWSTK and QAPYJWPROC.

***SCKTCP**
This data type will collect socket and TCP data. Socket and TCP data will be written to the file QAPYJWSKTC.

***SCKJOB**
This data type will collect the same data collected by the *SCKTCP data type and will additionally collect information regarding the jobs using each socket. Data collected for this data type will be written to the files QAPYJWSKTC and QAPYJWSKJB.

***JAVA** This data type will collect IBM Technology for Java VM and thread data. Data collected for this data type will be written to the files QAPYJWIJVM and QAPYJWIJVT.

***JAVASTACK**
This data type will collect call stacks for IBM Technology for Java threads included in the collection. Data collected for this data type will be written to the file QAPYJWIJVS.

Element 2: Interval frequency

Specifies how frequently (in intervals) each type of data should be collected.

***ALWAYS**
Data for the corresponding category will be collected in every interval.

integer
Specify the interval frequency in intervals. For example, if 5 is entered in this parameter, data for the corresponding category will be collected in intervals 5, 10, 15, 20, etc.

Top

Wait-based call stack data (WAITSTK)

Specifies a special case where call stacks can be collected for certain jobs and/or tasks which have been in an abnormal type of wait state on the system.

Single values

***NONE**
No wait-based call stacks will be included in the collection.

Other values (up to 2 repetitions)

Element 1: Category

Specifies the type of wait for which call stacks will be collected.

***ABNWAIT**
This category will collect call stacks which are in an abnormal wait. Abnormal waits are those that only occur in problem situations. These are either waits that rarely occur or waits that are taking a longer amount of time than normal to complete.

***CONFLICT**
This category will collect call stacks which are in a conflict wait. Conflict waits occur when a job or task is either in a seize condition or is waiting for a lock, mutex, or gate that is being held by another job or task.

Element 2: Minimum duration (microsecs)

Specifies the minimum duration which the job or task must be in the abnormal or conflict wait state before call stacks will be collected.

10000 A minimum duration value of 10,000 microseconds is used.

integer
Specify the number of microseconds that the job or task must be in the wait state.

Top

Job name (JOB)

Specifies the jobs that will be included in the Job Watcher collection. Job(s) selected on this parameter will be collected in addition to any jobs or tasks specified on the **Task name (TASKNAME)**, **TDE number (TDENBR)**, **Current user profile (CURUSRPRF)**, **Subsystem (SBS)**, or **Current storage pool (CURPOOL)** parameters.

Single values

***ALL** All jobs on the system are included.

***NONE**
None of the jobs on the system are included.

Other values (up to 20 repetitions)

Qualifier 1: Job name

name Specify the name of the job to include in the Job Watcher collection.

generic-name
Specify the generic name of the job to be included. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. For more information about generic object names, see the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/

Qualifier 2: User

***ALL** All jobs that match the specified job name are included.

name Specify the name of the user of the job to be included.

generic-name
Specify the generic user name of the jobs to be included.

Qualifier 3: Number

***ALL** All jobs that match the specified job name and user name are included.

number
Specify the job number to further qualify the job name and user name.

Top

Task name (TASKNAME)

Specifies the name of the task(s) which will be included in the Job Watcher collection. Task(s) selected on this parameter will be collected in addition to any jobs or tasks specified on the **Job name (JOB)**, **TDE number (TDENBR)**, **Current user profile (CURUSRPRF)**, **Subsystem (SBS)**, or **Current storage pool (CURPOOL)** parameters.

Single values

***ALL** All tasks on the system will be included in the collection.

***NONE**
None of the tasks on the system will be included in the collection.

Other values (up to 20 repetitions)

name Specify the name of the tasks(s) which will be included in the collection.

Top

TDE number (TDENBR)

Specifies the Task Dispatching Element (TDE) number of the TDE(s) which will be included in the Job Watcher collection. The TDE number is a unique identifier assigned to each job, thread, and task running in the system. The TDE number may be found by using the Display list of tasks option in the Display/Alter/Dump function of Start System Service Tools (STRSST). TDE(s) selected on this parameter will be collected in addition to any jobs or tasks specified on the **Job name (JOB)**, **Task name (TASKNAME)**, **Current user profile (CURUSRPRF)**, **Subsystem (SBS)**, or **Current storage pool (CURPOOL)** parameters.

You can specify twenty values for this parameter.

number
The TDE number of the TDE(s) which will be included in the collection.

Top

Current user profile (CURUSRPRF)

Specifies the current user profile of the jobs which will be included in the collection. Jobs selected on this parameter will be collected in addition to any jobs or tasks specified on the **Job name (JOB)**, **Task name (TASKNAME)**, **TDE number (TDENBR)**, **Subsystem (SBS)**, or **Current storage pool (CURPOOL)** parameters.

You can specify twenty values for this parameter.

name The current user profile name of the jobs which will be included in the collection. Only jobs which are running under the specified current user profile during the interval will be included in the collection.

Top

Subsystem (SBS)

Specifies the subsystem of the jobs which will be included in the Job Watcher collection. Jobs selected on this parameter will be collected in addition to any jobs or tasks specified on the **Job name (JOB)**, **Task name (TASKNAME)**, **TDE number (TDENBR)**, **Current user profile (CURUSRPRF)**, or **Current storage pool (CURPOOL)** parameters.

You can specify twenty values for this parameter.

name The name of the subsystem of the jobs which will be included in the collection. Only jobs which are running in the specified subsystem at the time data is gathered will be included in the collection.

Top

Current storage pool (CURPOOL)

Specifies the current system pool of the jobs and/or tasks which will be included in the Job Watcher collection. Jobs and/or tasks selected on this parameter will be collected in addition to any jobs or tasks specified on the **Job name (JOB)**, **Task name (TASKNAME)**, **TDE number (TDENBR)**, **Current user profile (CURUSRPRF)**, or **Subsystem (SBS)** parameters.

You can specify twenty values for this parameter.

1-64 The current pool number of the jobs and/or tasks which will be included in the collection. Only jobs and/or tasks which are running in the specified pool at the time data is gathered will be included in the collection.

Top

Force record write (FRCRCD)

Specifies when data records will be written to the Job Watcher database files.

***ITVEND**
Collected data will be written to the database files at the end of each interval.

***CALC**
Collected data will be written to the files at a time determined by the system. Data will always be available in the files after data collection is ended.

Top

Include inactive jobs/tasks (INCALLFST)

Specifies whether detailed data for all jobs and tasks, including inactive jobs and tasks, will be collected for the first interval.

***NO** Inactive jobs and tasks will not be collected on the first interval.

***YES** Inactive jobs and tasks will be collected on the first interval. This option will provide detailed information about the jobs and tasks that are waiting at the time the collection started. This information would otherwise not be available until an interval in which the job or task becomes active. Collecting this additional information may cause the duration of the first interval to be significantly larger than the rest of the collection.

Top

To file ASP threshold (TOASPTHLD)

Specifies the percentage of the auxiliary storage pool (ASP) that contains the Job Watcher database files which can be used before the collection is forced to end. Because the amount of data collected can be very large, this parameter allows you to limit how much of the ASP is consumed. If the database files exist in the system ASP and values are specified in both this parameter and the **System ASP threshold (SYSASPTHLD)**, the SYSASPTHLD parameter will override the value specified here.

***SYSTEM**
The threshold which is configured on the system for this ASP. This value is a percentage which is configured using the Change Storage Threshold function of the Start System Service Tools (STRSST) command. Data collection will be forced to end if this percentage of the ASP storage is consumed.

1-99 The percentage of the database files' ASP which may be used before data collection will be forced to end. For example, if you specify 80, data collection will be forced to end if more than 80% of the ASP storage is consumed. A value specified on this parameter will override the threshold which is configured on the system.

Top

System ASP threshold (SYSASPTHLD)

Specifies the percentage of the system auxiliary storage pool (ASP) which can be used before the collection is forced to end. Because Job Watcher allocates temporary storage and the amount of data collected can be very large, this parameter allows you to limit how much of the system ASP is consumed. If the database files exist in the system ASP and values are specified in both this parameter and the **To file ASP threshold (TOASPTHLD)** parameter, the value specified here will take precedence.

***SYSTEM**
The threshold which is configured on the system for the system ASP. This value is a percentage which is configured using the Change Storage Threshold function of the Start System Service Tools (STRSST) command. Data collection will be forced to end if this percentage of the system ASP storage is consumed.

1-99 The percentage of the system ASP which may be used before data collection will be forced to end. For example, if you specify 80, data collection will be forced to end if more than 80% of the system ASP storage is consumed. A value specified on this parameter will override the threshold which is configured on the system.

Top

Condition control file (CONDCTLF)

Specifies the name of the condition control file. This file must be a source physical file and may be used to specify conditions that Job Watcher will use to limit data collection. If a file is specified on this parameter, the data collected will be compared against the conditions defined in the file.

Conditions must be specified in a specific format. A description of valid conditions and formats follows:

Direct field comparison

Syntax: FIELDNAME.COMPARAND.VALUE

Example 1: The condition will be met when more than 75 synchronous database writes occur in the interval

SYNDBWRT.GT.75

Example 2: The condition will be met when the wait time in bucket 6 is between 30 and 80 microseconds.

QTIME06.GE.30.AND.QTIME06.LE.80

Rate condition

Syntax: RATE(FIELDNAME).COMPARAND.VALUE

Example: The condition will be met when the rate of synchronous database writes is greater than 10 per second

RATE(SYNDBWRT).GT.10

Percent condition (applies to time spent in a particular wait bucket)

Syntax: PERCENT(FIELDNAME).COMPARAND.VALUE

Example: The condition will be met when more than 10 percent of time spent waiting was counted in bucket 9

PERCENT(QTIME09).GT.10

Average condition

Syntax: AVERAGE(FIELDNAME1,FIELDNAME2).COMPARAND.VALUE where FIELDNAME1 is a time and FIELDNAME2 is a corresponding count

Note: At this time the only time/count combinations reported in Job Watcher are the wait bucket times and counts reported in the QAPYJWTDE file.

Example: The condition will be met when the average wait time for a wait counted in bucket 5 is greater than 50 microseconds

AVERAGE(QTIME05,QCOUNT05).GT.50

Single values

***NONE**
No condition will be specified for the Job Watcher collection.

Qualifier 1: File

name Specify the name of the file which contains the condition information.

Qualifier 2: Library

name Specify the name of the library which contains the condition control file.

Top

Condition control member (CONDCTLMBR)

Specifies the file member which contains the condition information.

***FIRST**
The first member in the file will be used.

name Specify the name of the member which contains the condition information.

Top

Condition type (CONDTYPE)

Specifies the type of conditional collection.

***PERITV**
The specified condition will be checked in every interval. In this type of collection, data will only be written to the database files for intervals in which the condition was satisfied. If an exit program is specified on the on the **User exit program (EXITPGM)** parameter it will be called in each interval where the condition was satisfied.

***TRIGGER**
The specified condition will be checked in each interval until the condition is satisfied. Once the condition has been met, data will be unconditionally written to the database files for the remainder of the collection. If an exit program is specified on the on the **User exit program (EXITPGM)** parameter it will be called one time in the interval where the condition was satisfied.

***UNTILMET**
Data will be unconditionally written to the database files until the condition is satisfied. Once the condition has been met, the collection will end. If an exit program is specified on the on the **User exit program (EXITPGM)** parameter it will be called one time before the collection ends.

Top

Timeout option (TIMEOUT)

Specifies how long the collection should run without writing any data to the database files.

Single values

***NONE**
The collection will not time out.

Element 1: Option

***NBRSEC**
The collection will time out after the number of seconds specified in element 2 of this parameter has elapsed without the condition being met.

***NBRITV**
The collection will time out after the number of intervals specified in element 2 of this parameter have been collected without the condition being met.

Element 2: Value

integer
Specify the number of seconds (for *NBRSEC) or the number of intervals (for *NBRITV) to use as the timeout criteria for the conditional collection.

Top

Consecutive occurrence count (OCCURS)

Specifies number of consecutive intervals in which the specified condition must occur before it is considered satisfied.

1 The condition must occur in one interval to be considered satisfied.

integer
Specify the number of consecutive intervals in which the condition must occur.

Top

History size (HSTSIZE)

Specifies the amount of data (in intervals) that should be buffered as history during the conditional collection. The specified amount of data will be maintained until the condition has been satisfied, at which time all buffered data will be written to the database files along with the data from the current interval.

0 No history data will be buffered.

integer
Specify the number of intervals which should be buffered as history.

Top

User exit program (EXITPGM)

Specifies the user exit program which will be called at the time the condition is satisfied.

Single values.

***NONE**
No exit program will be called.

Qualifier 1: User exit program

name Specify the name of the user exit program.

Qualifier 2: Library

name Specify the name of the library where the user exit program is located.

Top

Exit program data (EXITPGMDTA)

Specifies any data which should be passed to the user exit program specified on the **User exit program (EXITPGM)** parameter.

***NONE**
No data will be passed to the user exit program.

character-value
Specify any data which should be passed to the user exit program.

Top

Examples

Example 1: Add a Job Watcher Definition which will Collect Data for a Specific Job with No Delay Between Samples

```
ADDJWDFN   DFN(MYJOB) COLITV(*NODELAY)
           JOB(123456/MYUSER/MYJOB)  TASKNAME(*NONE)
```

This command will add a Job Watcher definition which will collect data for job 123456/MYUSER/MYJOB. Data will be collected as fast as possible, with no delay between intervals. To collect Job Watcher data using this definition use the **Start Job Watcher (STRJW)** command with MYJOB specified for the **Definition (DFN)** parameter.

Example 2: Add a Job Watcher Definition which will Collect Data for a Generic Job, Collecting SQL Data

```
ADDJWDFN   DFN(GENJOB)  COLITV(5)  ADDDTACGY((*SQLDETAIL))
           JOB(*ALL/*ALL/QPADEV*)  TASKNAME(*NONE)
```

This command will add a Job Watcher definition which will collect data for all jobs with names which begin with 'QPADEV'. Data will be collected with a delay of 5 seconds between intervals. Standard data and SQL data will be collected. To collect Job Watcher data using this definition use the **Start Job Watcher (STRJW)** command with GENJOB specified for the **Definition (DFN)** parameter.

Example 3: Add a Job Watcher Definition which will Collect Data for All Jobs and Tasks, Collecting Call Stacks for those in Conflict Waits

```
ADDJWDFN   DFN(CONFLICT)  WAITSTK((*CONFLICT 100))
```

This command will add a Job Watcher definition which will collect data for all jobs and tasks on the system. Data will be collected at the default interval of 10 seconds. Data collected with this definition will include standard data, as well as call stacks for any job or task which was in a conflict wait for 100 microseconds or longer. To collect Job Watcher data using this definition use the **Start Job Watcher (STRJW)** command with CONFLICT specified for the **Definition (DFN)** parameter.

Example 4: Add a Job Watcher Definition which will Conditionally Collect Data for a Generic Job Name

```
ADDJWDFN   DFN(COND)  JOB(*ALL/*ALL/QSQ*)
           TASKNAME(*NONE)  CONDCTLF(MYLIB/MYCTLFILE)
           CONDCTLMBR(MYCTLMBR)  CONDTYPE(*TRIGGER)
           TIMEOUT(1000/*NBRITV)  HSTSIZE(5)
```

This command will add a Job Watcher definition which will conditionally collect data for all jobs with names which begin with 'QSQ'. The file/member which specifies the condition to evaluate is MYCTLFILE/MYCTLMBR and this file exists in library MYLIB. No data will be written to the Job Watcher database files until the specified condition has been satisfied. At the time the condition is met, the 5 previous intervals of history data will be written to the database files. If data collection continues for 1000 intervals and the condition is not satisfied in any interval, the collection will end without writing any data to the database files. To collect Job Watcher data using this definition use the **Start Job Watcher (STRJW)** command with COND specified for the **Definition (DFN)** parameter.

Top

Error messages

***ESCAPE Messages**

CPFAF10
Definition or filter already exists.

CPFB518
The user does not have the required authority.

Top

Add Kerberos Keytab Entry (ADDKRBKTE)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The Add Kerberos Keytab Entry (ADDKRBKTE) command is used to add an entry to the Kerberos keytab file for a specified principal name. A principal name consists of the user name or service name and the name of the realm in which that user or service belongs. If keytab entries exist for the specified principal name, the default is to add one to the largest version number of the existing entries.

Restrictions:

The Network Authentication Service Commands and APIs support job environments for most EBCDIC CCSIDs. CCSID 290 and 5026 are not supported because of the variance of lower-case letters a to z.

Top

Parameters

Keyword	Description	Choices	Notes
PRINCIPAL	Principal	*Element list*	Required, Positional 1
	Element 1: Name	*Character value*	
	Element 2: Realm	*Character value*, ***DFT**	
PASSWORD	Password	*Character value*	Required, Positional 2
KEYTABFILE	Keytab file	*Path name*, ***DFT**	Optional
VERSION	Version	1-255, ***GEN**	Optional

Top

Principal (PRINCIPAL)

Specifies the principal name of a user or service principal on a host name in a Kerberos network. The principal and key pairs in the keytab file allow services running on the host to be authenticated by a Key Distribution Center (KDC). All the principals are added to the Kerberos server which maintains a database of all users and services within a Kerberos realm.

This is a required parameter.

Element 1: Name

Specifies the principal name or service principal on a specified host name.

character-value

Specify the user name of the Kerberos principal.

The Kerberos principal has a minimum length of 1 character and a maximum length of 256 characters. Valid characters are case sensitive and include all alpha-numeric characters (a-z, A-Z, 0-9) and any printable ASCII character. The principal name format is taken from the Kerberos 5 GSS-API mechanism (RFC 1964).

Special characters allowed:

`/ - delimit name components.`

Element 2: Realm

Specifies the realm in which the Kerberos user is registered and in which initial authentication took place.

***DFT** The default realm for the local system will be used. Typically, the default realm and the KDC for that realm are indicated in the Kerberos krb5.conf configuration file. If the default realm has not been set, it is obtained from the default_realm entry in the [libdefaults] section of the Kerberos configuration file.

character-value
Specify the name of the Kerberos realm where the user specified for the first element of this parameter is registered.

The name has a minimum length of 1 character and a maximum length of 256 characters. Valid characters are case sensitive and include all alpha-numeric characters (a-z, A-Z, 0-9) and any printable ASCII character. The principal name format is taken from the Kerberos 5 GSS-API mechanism (RFC 1964).

Special characters allowed:

`@ - start realm.`

Top

Password (PASSWORD)

Specifies the password that allows the principal to authenticate in the Key Distribution Center (KDC).

This is a required parameter.

character-value
Specify the password value. The password can be up to 255 characters long.

Top

Keytab file (KEYTABFILE)

Specifies the Kerberos keytab file where the group of principals and its keys are stored.

***DFT** The default keytab file for the current user will be used. If the KRB5_KTNAME environment variable is set, this is the name of the default keytab file. Otherwise, the keytab file name is obtained from the default_keytab_name entry in the [libdefaults] section of the Kerberos configuration file. If this entry is not defined, the default keytab file name is /QIBM/UserData/OS400/NetworkAuthentication/keytab/krb5.keytab.

path-name
Specify the path name of the stream file which contains the Kerberos keytab file to use.

Top

Version (VERSION)

Specifies the key version number of the keytab entry.

***GEN** Generate the version number based on existing keytab entries. The first time a keytab entry is created for the specified principal, the default version number will be 1. If keytab entries exists for the specified principal, the default version number will be one greater than the largest version number of the existing entries.

1-255 Specify the version number for the keytab entry for the specified principal.

Top

Examples

Example 1: Adding a Service Principal Keytab Entry

```
ADDKRBKTE   PRINCIPAL('krbsvr400/camolts.myco.com'  MYCO.COM)
            PASSWORD(uneed2chg)  VERSION(*GEN)  KEYTABFILE(*DFT)
```

This command adds a service principal entry into the default Key Table file.

Example 2: Adding a Principal Name Keytab Entry

```
ADDKRBKTE   PRINCIPAL('julius'  GUADA.LAJARA.COM)
            PASSWORD(uneed2chg)  VERSION(4)  KEYTABFILE(*DFT)
```

This command adds a principal name entry into the default Key Table file.

Top

Error messages

***ESCAPE Messages**

CPFC601
No default keytab file found.

CPFC602
Keytab file &3 not found.

CPFC603
Keytab entry &2 not found.

CPFC605
Entry &1 could not be added to keytab file &2.

CPFC607
Key version &1 not found for &2.

CPFC61B
The principal name &3 can not be parsed.

Top

Add Kerberos Ticket (ADDKRBTKT)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The Add Kerberos Ticket (ADDKRBTKT) command is used to obtain and cache Kerberos ticket-granting tickets. This command is similar to the **kinit** tool that is commonly found in other implementations of the Kerberos protocol, such as the SEAM and MIT Reference implementations.

Restrictions:

- The user must be registered as a principal with the Key Distribution Center (KDC) prior to running this command.

  The Network Authentication Service Commands and APIs support job environments for most EBCDIC CCSIDs. CCSID 290 and 5026 are not supported because of the variance of lower-case letters a to z.

Top

Parameters

Keyword	Description	Choices	Notes
PRINCIPAL	Principal	*Element list*	Required, Positional 1
	Element 1: Name	*Character value*	
	Element 2: Realm	*Character value*, ***DFT**	
PASSWORD	Password	*Character value*	Optional, Positional 2
KEYTABFILE	Keytab file	*Path name*, *DFT	Optional
CCF	Credentials cache file	*Path name*, ***DFT**	Optional
ALWFWD	Allow forwarding	***NO**, *YES	Optional
ALWPRX	Allow proxy	***NO**, *YES	Optional

Top

Principal (PRINCIPAL)

Specifies the principal name of a user or service principal on a host name in a Kerberos network. The principal and key pairs in the keytab file allow services running on the host to be authenticated by a Key Distribution Center (KDC). All the principals are added to the Kerberos server which maintains a database of all users and services within a Kerberos realm.

This is a required parameter.

Element 1: Name

Specifies the principal name or service principal on a specified host name.

character-value
Specify the user name of the Kerberos principal.

The Kerberos principal has a minimum length of 1 character and a maximum length of 256 characters. Valid characters are case sensitive and include all alpha-numeric characters (a-z, A-Z, 0-9) and any printable ASCII character. The principal name format is taken from the Kerberos 5 GSS-API mechanism (RFC 1964).

Special characters allowed:

```
/ - delimit name components.
```

Element 2: Realm

Specifies the realm in which the Kerberos user is registered and in which initial authentication took place.

***DFT** The default realm for the local system will be used. Typically, the default realm and the KDC for that realm are indicated in the Kerberos krb5.conf configuration file. If the default realm has not been set, it is obtained from the default_realm entry in the [libdefaults] section of the Kerberos configuration file.

character-value
Specify the name of the Kerberos realm where the user specified for the first element of this parameter is registered.

The name has a minimum length of 1 character and a maximum length of 256 characters. Valid characters are case sensitive and include all alpha-numeric characters (a-z, A-Z, 0-9) and any printable ASCII character. The principal name format is taken from the Kerberos 5 GSS-API mechanism (RFC 1964).

Special characters allowed:

```
@ - start realm.
```

Top

Password (PASSWORD)

Specifies the password that allows the principal to authenticate in the Key Distribution Center (KDC).

This is a required parameter if the keytab file name is not defined.

character-value
Specify the password value. The password can be up to 255 characters long.

Top

Keytab file (KEYTABFILE)

Specifies the Kerberos keytab file where the group of principals and its keys are stored.

***DFT** The default keytab file for the current user will be used. If the KRB5_KTNAME environment variable is set, this is the name of the default keytab file. Otherwise, the keytab file name is obtained from the default_keytab_name entry in the [libdefaults] section of the Kerberos configuration file. If this entry is not defined, the default keytab file name is /QIBM/UserData/OS400/NetworkAuthentication/keytab/krb5.keytab.

path-name
Specify the path name of the stream file which contains the Kerberos keytab file to use.

Top

Credentials cache file (CCF)

Specifies the credentials cache file that this command will use. This cache is used to store each of the new tickets, the session key and other information in the corresponding KrbCredInfo sequence from the encrypted part of the KRB_CRED message.

***DFT** The default credentials cache file for the current user is used. If the KRB5CCNAME environment variable is set, this is the name of the default cache. Otherwise, the name is obtained from the file specified by the _EUV_SEC_KRB5CCNAME_FILE environment variable. If this environment variable is not set, the name is obtained from the krb5ccname in the HOME directory. If this file does not exist or if there is no default credentials cache name set in the file, a new credentials cache file is created.

path-name
Specify the path name of the credentials cache file to use.

Top

Allow forwarding (ALWFWD)

Specifies whether the Kerberos ticket will be forwardable.

The FORWARDABLE flag in a ticket is normally only interpreted by the ticket-granting service. It can be ignored by the application server.

The FORWARDABLE flag has an interpretation similar to that of the PROXIABLE flag, except ticket-granting tickets may also be issued with different network addresses.

This flag allows for authentication forwarding without requiring the user to enter a password again. If the flag is not set, then authentication forwarding is not permitted, but the same end result can still be achieved if the user engages in the authentication server exchange with the requested network addresses and supplies a password.

***NO** The ticket will not be forwardable.

***YES** The ticket will be forwardable.

Top

Allow proxy (ALWPRX)

Specifies whether the Kerberos ticket will be a proxiable ticket.

The PROXIABLE flag in a ticket is normally only interpreted by the service-granting service. It can be ignored by application servers. When this parameter is set to *YES, the ticket-granting server can issue a new ticket (but not a ticket-granting ticket) with a different network address based on this ticket.

***NO** The ticket is not proxiable.

***YES** The ticket is proxiable.

Top

Examples

Example 1: Adding a Forwardable Ticket

```
ADDKRBTKT   PRINCIPAL('krbsrv400/guada.lajara.com')
            PASSWORD('my1pwd')  ALWFWD(*YES)
```

This command adds a forwardable ticket using the 'krbsrv400/guada.lajara.com' principal and the default realm.

Example 2: Adding a Proxiable Ticket

```
ADDKRBTKT   PRINCIPAL('krbsrv400/guada.lajara.com')
            PASSWORD('my1pwd')  ALWPRX(*YES)
```

This command adds a proxiable ticket using the :'krbsrv400/guada.lajara.com' principal and the default realm.

Example 3: Adding a Ticket for Non-default Realm

```
ADDKRBTKT   PRINCIPAL('krbsrv400/guada.lajara.com'
                      'MEX.ICO.COM')
            PASSWORD('my1pwd')  ALWFWD(*YES)
```

This command adds a forwardable ticket using the principal with user name 'krbsrv400/ guada.lajara.com' and realm 'MEX.ICO.COM'.

Top

Error messages

***ESCAPE Messages**

CPFC602
Keytab file &3 not found.

CPFC608
The default credential cache file name can not be obtained.

CPFC609
The principal from credential cache file &1 can not be retrieved.

CPFC60A
No initial ticket granting ticket (TGT) available.

CPFC60B
The initial credentials can not be obtained.

CPFC60C
The ticket from credentials cache &1 can not be retrieved.

CPFC60E
Password is not correct for principal.

CPFC60F
Initial credential can not be stored in credentials cache &1.

CPFC610
No default credentials cache found.

CPFC611
Credentials cache file operation fails.

CPFC613
The credentials cache directory can not be read.

CPFC615
The password can not be read.

CPFC61B
The principal name &3 can not be parsed.

Top

Add LAN Adapter Information (ADDLANADPI)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Local Area Network Adapter Information (ADDLANADPI) command adds an adapter name entry to the adapter file.

Top

Parameters

Keyword	Description	Choices	Notes
ADPTNAME	Adapter	*Name*	Required, Positional 1
ADPTADR	Adapter address	000000000001-7FFFFFFFFFFF	Required, Positional 2
LINETYPE	Line type	*DDI, *TRN	Required, Positional 3
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Top

Adapter (ADPTNAME)

Specifies the name of the adapter being added to the adapter file. The name can be a maximum of 10 characters in length.

This is a required parameter.

Top

Adapter address (ADPTADR)

Specifies the 12-character hexadecimal adapter address.

This is a required parameter.

Top

Line description (LINETYPE)

Specifies the line type of the entry.

***DDI** A distributed data interface (DDI) line type is used.

***TRN** A token-ring network (TRN) line type is used.

This is a required parameter.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the object.

The possible values are:

***BLANK**
Text is not specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Examples

```
ADDLANADPI   ADPTNAME(PAYROLL)  ADPTADR(00000000012B)
             LINETYPE(*TRN)
```

This command adds the adapter PAYROLL, with the address 00000000012B, to the network adapter file. The line is a token-ring line.

Top

Error messages

***ESCAPE Messages**

CPF8B48
Adapter name - &29 or address - &30 already in the network adapter file

CPF8B68
Line description &23 not found.

CPF8B69
Line description &23 not valid for requested action.

CPF8B74
Request to display active adapters failed.

CPF8B75
No adapter entries in network adapter file.

CPF8B76
No functional addresses for adapter.

Top

Add Logical File Member (ADDLFM)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The Add Logical File Member (ADDLFM) command adds a named file member to a logical file which must already exist on the system. The maximum number of members that can be added to the file is specified for the **Maximum members (MAXMBRS)** parameter on the Create Logical File (CRTLF) command or the Change Logical File (CHGLF) command. To add other members to the file, use the ADDLFM command to specify each one.

A logical file member can use the data from all, or a subset of, the physical files referenced by the logical file. Each member has its own set of data and can have its own access path that provides an organization to that data. The system attempts to implicitly share an access path already on the system.

Each added member has the same attributes as those defined in the logical file.

Restrictions:

- To add a member to a keyed logical file, you must have object operational (*OBJOPR) authority and either object management (*OBJMGT) or object alter (*OBJALTER) authority for each of the physical files on which the logical file member is based (specified explicitly by the DTAMBRS parameter or implicitly by the PFILE or JFILE keyword specified in DDS).
- For a member added to a non-keyed logical file, *OBJOPR authority for each of the physical files is required.
- This command is conditionally threadsafe. In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA. This command is also not threadsafe and fails for Distributed Data Management (DDM) files of type *SNA, when SYSTEM(*RMT) or SYSTEM(*FILETYPE) is specified.

Note: An *EXCLRD lock is required on the file to add a member. Because this command adds a member to a file in a library, the library must not be locked (*SHRNUP or *EXCLRD with the Allocate Object (ALCOBJ) command) in another job.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	Logical file	*Qualified object name*	Required, Positional 1
	Qualifier 1: Logical file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MBR	Logical file member	*Name*	Required, Positional 2

Keyword	Description	Choices	Notes
DTAMBRS	Physical file data members	Single values: ***ALL** Other values (up to 32 repetitions): *Element list*	Optional
	Element 1: Physical file	*Qualified object name*	
	Qualifier 1: Physical file	*Name*	
	Qualifier 2: Library	*Name*, ***CURRENT**	
	Element 2: Member	Single values: ***NONE** Other values (up to 32 repetitions): *Name*	
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional
SHARE	Share open data path	***NO**, *YES	Optional

Top

Logical file (FILE)

Specifies the logical file to which the member is to be added.

This is a required parameter.

Qualifier 1: Logical file

name Specify the name of the logical file.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library is used to locate the logical file. If no library is specified as the current library for the job, QGPL is used.

name Specify the library where the logical file is located.

Top

Logical file member (MBR)

Specifies the logical file member to be added. The member name must be unique in the file.

If the FILE parameter specifies a DDM file and a member name is specified as part of the remote file name in the DDM file, the member names must be the same.

This is a required parameter.

name Specify the name of the logical file member to be added.

Top

Physical file data members (DTAMBRS)

Specifies the physical files and members that contain the data associated with the logical file member being added by this command. A logical file member can be based on all (*ALL) of the physical files and members on which the logical file itself is based, or the member can be based on a subset of the total files and members.

Note: When adding a member to a logical file that is a DDM file, the physical file, if specified, must also be a DDM file with its library and member(s) specified explicitly. *CURRENT is not supported when the logical file is a DDM file.

When a logical file is created, the physical files specified for the PFILE or JFILE DDS keyword are used to create the logical file. If no library name is specified for the physical files on the PFILE or JFILE keyword, the library list (*LIBL) at file creation time is used to find the physical files; the physical files from the library list are used to create the logical file. The qualified physical files from the PFILE or JFILE keyword (regardless of whether a library name was specified or if the library list was used to find the files) are the physical files associated with the logical file. The names of the physical files associated with the logical file are saved in the description of the logical file. When a member is added to the logical file, the DTAMBRS parameter is used to specify the physical file members associated with the logical file member. Each physical file name specified on the DTAMBRS parameter must be the name of a physical file that is associated with the logical file (saved in the description of the logical file).

Single values

***ALL** The logical file member being added is based on all the physical files and members (that exist at the time this CRTLF command is entered) used by the logical file. At least one member must exist in at least one of the physical files. The physical file names are specified for the PFILE or JFILE parameter in the DDS.

Element 1: Physical file

Qualifier 1: Physical file

name Specify the names of the physical files that contain the data being accessed by the logical file member being added.

The physical file names must match a name on the PFILE or JFILE keywords in the DDS and cannot be specified more often on the DTAMBRS parameter than on the PFILE or JFILE keywords in the DDS. For join logical files, all physical files specified for the JFILE keyword must be specified for the DTAMBRS parameter and each physical file must contain only one member. If a physical file name is not specified for a physical file that is on a PFILE or JFILE keyword in the DDS, the logical file member is not based on any member of that physical file.

Qualifier 2: Library

***CURRENT**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

If a library name is not specified, the current library name (*CURRENT) from the logical file description is used. If the library name is specified, the physical file must be a physical file associated with the logical file. If the logical file is associated with more than one physical file of the same name, the library name must be specified.

name Specify the name of the library to be searched.

Element 2: Member

Single values

***NONE**

A member name is not specified.

Other values (up to 32 repetitions)

name Specify the names of the physical file members that contain the data being accessed by the logical file member being added.

When the FILE parameter specifies a join logical file or an arrival sequence logical file, only one data member must be specified for the DTAMBRS parameter for each physical file that was specified for the PFILE or JFILE keyword in the DDS. *ALL is valid only if each based-on physical file has only one member. If any of the physical files has more than one member, the specific physical file member must be specified for the DTAMBRS parameter.

The same physical file name can be specified more than once on the JFILE keyword. In this case, each occurrence of the file name is treated as a different based-on physical file, and must be specified for the DTAMBRS parameter.

Up to 32 qualified physical file names and physical file member names can be specified. Also, the total number of member names cannot exceed 32. For example, one file can specify 32 members, two files can each have 16 members, or 32 files can each have one member specified.

For DDM file:

- The file names specified in the DTAMBRS parameter must be the names of the DDM files that represent the remote based-on physical files. If a member name was specified as part of the remote file name in the DDM file, only that member name can be specified for the DTAMBRS parameter. The member names must be the actual remote file member names.
- The based-on physical files must be at the same system location as the logical file to which the member is being added.
- When no member name is specified for the remote file name in the DDM file, all members are accessible. When only one member name is specified, only that member is accessible through that DDM file.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the object.

***BLANK**
No text is specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Share open data path (SHARE)

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

***NO** The ODP is not shared with other programs in the routing step. A new ODP for the file is created and used every time a program opens the file.

***YES** The same ODP is shared with each program in the job that also specifies *YES when it opens the file.

Top

Examples

```
ADDLFM   FILE(INVENLIB/STOCKTXS)  MBR(JANUARY)
         DTAMBRS((INVENTXS (JANUARY)))
         TEXT('JANUARY STOCK ACTIVITY BY LOCATION')
```

This command adds a member named JANUARY to the logical file named STOCKTXS in the INVENLIB library. The logical file has access to the data stored in the JANUARY member of the INVENTXS physical file.

Top

Error messages

***ESCAPE Messages**

CPF3204
Cannot find object needed for file &1 in &2.

CPF7306
Member &1 not added to file &2 in &3.

Top

Add Library List Entry (ADDLIBLE)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The Add Library List Entry (ADDLIBLE) command adds a library name to the user portion of the library list for the current thread. The user portion is the last portion of the library list. It follows the system portion and, if they exist, any product libraries and the current library entry. You can specify where the library should be placed in the user portion of the library list.

Top

Parameters

Keyword	Description	Choices	Notes
LIB	Library	*Name*	Required, Positional 1
POSITION	Library list position	Single values: ***FIRST**, *LAST Other values: *Element list*	Optional, Positional 2
	Element 1: List position	*AFTER, *BEFORE, *REPLACE	
	Element 2: Reference library	*Name*	

Top

Library (LIB)

Specifies the library to be added to the user portion of the library list for the current thread. A maximum of 250 libraries may exist in the user portion of the library list. Only one library name is added at a time with this command.

This is a required parameter.

name Specify the name of the library to be added to the user portion of the library list for the current thread.

Top

Library list position (POSITION)

Specifies the position in the user portion of the library list for the current thread where the library is inserted. This parameter can be specified as a single value or as a list of two elements.

Single values

***FIRST**
The library is inserted in front of the libraries existing in the user portion of the library list for the current thread.

***LAST**
The library is added to the end of the user portion of the library list for the current thread.

Element 1: List position

***AFTER**
The library specified for the **Library (LIB)** parameter is added to the user portion of the library list for the current thread after the reference library (element 2).

***BEFORE**
The library specified for the LIB parameter is added to the user portion of the library list for the current thread before the reference library (element 2).

***REPLACE**
The library specified for the LIB parameter is inserted into the library list for the current thread in the position currently held by the reference library (element 2) and the reference library is then removed from the list.

Element 2: Reference library

name Specify the name of the library to be the reference library when *AFTER, *BEFORE, or *REPLACE is specified for the list position (element 1). This library must exist in the user portion of the library list for the current thread.

Top

Examples

```
ADDLIBLE  LIB(TESTLIB)  POSITION(*LAST)
```

This command adds the library TESTLIB to the end of the user portion of the library list.

Top

Error messages

***ESCAPE Messages**

CPF2103
Library &1 already exists in library list.

CPF2106
Library list not available.

CPF2110
Library &1 not found.

CPF2113
Cannot allocate library &1.

CPF2118
Library &1 not added.

CPF2149
Library &1 was not found in the user library list.

CPF2176
Library &1 damaged.

CPF2182
Not authorized to library &1.

Add License Key Information (ADDLICKEY)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add License Key Information (ADDLICKEY) command can be used to add the software license key information to the license repository for products with keyed compliance. Products with ″keyed compliance″ require that you have a software license key from the software provider in order to change the usage limit or the expiration date of the license information.

The license repository stores product license information for each unique product, license term, feature, and system. The repository can contain licenses for any system, and the product need not be installed.

If the product is installed on the system and the license is for this system, this command installs the license, which changes the usage limit from the product's default usage limit to the licensed usage limit. The expiration date is also set.

If a software license key already exists in the repository for the unique product, license term, feature, and system, the software license key information is replaced.

If a license does not exist on the system, the added software license key uses the default values for its threshold (90 percent of the usage limit), message queue (*OPSYS), and log (usage limit violations are not logged) attributes. If a license exists, the values on these attributes do not change. To change any of these values, you can use the Change License Information (CHGLICINF) command.

Restrictions: This command is shipped with public *EXCLUDE authority.

Top

Parameters

Keyword	Description	Choices	Notes
LICKEYINP	License key input	***PROMPT**, *LICKEYFILE, *TAPE	Optional, Positional 1
PRDID	Product identifier	*Character value*	Optional, Positional 2
LICTRM	License term	*Character value*	Optional, Positional 3
FEATURE	Feature	*Character value*	Optional, Positional 4
SERIAL	System serial number	*Character value*, ***LOCAL**, *REMOTE, *ALL	Optional
PRCGRP	Processor group	*Character value*, ***ANY**	Optional
LICKEY	License key	*Element list*	Optional
	Element 1: Characters 1 - 6	*Character value*	
	Element 2: Characters 7 - 12	*Character value*	
	Element 3: Characters 13 - 18	*Character value*	
USGLMT	Usage limit	0-999999, **1**, *NOMAX	Optional
EXPDATE	Expiration date	*Date*, ***NONE**	Optional
VNDDTA	Vendor data	*Character value*, ***NONE**	Optional

Keyword	Description	Choices	Notes
LICKEYFILE	License key file	*Qualified object name*	Optional
	Qualifier 1: License key file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LICKEYMBR	License key member	*Name*, ***FIRST**, *LAST	Optional
DEV	Device	*Name*	Optional
VOL	Volume identifier	*Character value*, ***MOUNTED**	Optional
SEQNBR	Sequence number	1-9999, ***SEARCH**, *NEXT	Optional
ENDOPT	End of tape option	***REWIND**, *LEAVE, *UNLOAD	Optional

Top

License key input (LICKEYINP)

Specifies how the software license key information to be added is supplied.

***PROMPT**
The software license key information is supplied through prompting.

***LICKEYFILE**
The software license key information is taken from the file specified on the LICKEYFILE parameter.

***TAPE** The software license key information is taken from a data file with the label QFILEPGMKEY on the tape device specified on the DEV parameter. The QFILEPGMKEY data file must be created on the tape using the Copy To Tape (CPYTOTAP) command, specifying a file in the format of QSYS/QALZAKEY on the FROMFILE parameter.

Top

Product identifier (PRDID)

Specifies the seven-character identifier of the product for which software license key information is added.

Top

License term (LICTRM)

Specifies the license term for which software license key information is added. This information is supplied by the software provider. Specify the license term in Vx, VxRy, or VxRyMz format, where x and y can be a number from 0 through 9, and z can be a number 0 through 9 or a letter A through Z.

Top

Feature (FEATURE)

Specifies the feature of the product specified on the PRDID parameter for which the software license key information is added.

5001 The software license key information for feature 5001 is added.

feature
Specify the number of the feature for which software license key information is added. Valid values range from 5001 through 9999.

Top

System serial number (SERIAL)

Specifies the serial number of the system for which software license key information is added.

Note: The *REMOTE and *ALL values are valid only when *LICKEYFILE or *TAPE is specified for the **License key input (LICKEYINP)** parameter.

***LOCAL**
The software license key information for the local system is added.

***REMOTE**
The software license key information for all remote systems named in the file specified on the LICKEYFILE parameter or in the tape file with the label QFILEPGMKEY is added.

***ALL** The software license key information for all systems named in the file specified on the LICKEYFILE parameter or in the tape file with the label QFILEPGMKEY is added.

system-serial-number
Specify the serial number of the system for which software license key information is added.

Top

Processor group (PRCGRP)

Specifies the system processor group for which software license key information is added. The software provider supplies this information with the software license key.

***ANY** The software license key is for any processor group.

processor-group
Specify the processor group of the system for which software license key information is added.

Top

License key (LICKEY)

Specifies the software license key that is supplied by the software provider. The license key has three elements, each of which are six characters long. Valid values for the characters are A-F and 0-9, and can be specified in the format: cccccc cccccc cccccc, where c is a valid character.

Top

Usage limit (USGLMT)

Specifies the usage limit for this product or feature. The software provider authorizes the value of the usage limit. For a concurrent usage limit, this is the maximum number of jobs allowed to access the product or feature at any given time. For a registered usage limit, this is the maximum number of license users that can be registered to use this product or feature.

1 The number of users is limited to one.

***NOMAX**
The number of users is not limited.

usage-limit
Specify the maximum number of users for this product or feature. Valid values range from 0 through 999999.

Top

Expiration date (EXPDATE)

Specifies the expiration date of the product license. After this date, no users over the default usage limit can use the product or feature in compliance with this software license key.

The software provider supplies the expiration date with the software license key. To use the product after the expiration date, you must obtain a new software license key from the software provider.

***NONE**
The license has no expiration date.

expiration-date
Specify the expiration date of the product license.

Top

Vendor data (VNDDTA)

Specifies the vendor data. The software provider supplies this information with the software license key.

***NONE**
No vendor data is specified.

vendor-data
Specify a maximum of eight characters of vendor data.

Top

License key file (LICKEYFILE)

Specifies the qualified name of the file from which the software license key information is taken. This input file must be in the format of QSYS/QALZAKEY, and can be created by using the LICKEYFILE parameter on the Display License Key Information (DSPLICKEY) command.

The name of the license key file can be qualified by one of the following library values:

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched.

library-name
Specify the name of the library to be searched.

The possible value is:

license-key-file
Specify the name of the file that contains the software license key information.

Top

License key member (LICKEYMBR)

Specifies the name of the member in the file specified on the LICKEYFILE parameter from which the software license key information is taken.

***FIRST**
The oldest member in the file is used.

***LAST**
The newest member in the file is used.

license-key-member
Specify the name of the member from which to get information.

Top

Device (DEV)

Specifies the name of the tape device holding the tape from which the software license key information is copied.

Top

Volume identifier (VOL)

Specifies the volume identifier of the tape from which the software license key information is copied.

***MOUNTED**
The volume currently placed in the device is used.

volume-identifier
Specify the identifier of the volume from which the software license key information is copied.

Top

Sequence number (SEQNBR)

Specifies the sequence number of the data file from which the software license key information is copied.

***SEARCH**
A search is made for a data file with an identifier that matches the label QFILEPGMKEY.

***NEXT**
The next sequence is used if that sequence is for a file with the label QFILEPGMKEY.

sequence-number
Specify the sequence number of the data file with the label QFILEPGMKEY to use for copying the software license key information.

Top

End of tape option (ENDOPT)

Specifies what positioning operation is done automatically on the tape volume after the software license key information is copied.

***REWIND**
The tape is automatically rewound, but not unloaded, after the operation has ended.

***LEAVE**
The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

***UNLOAD**
The tape is automatically rewound and unloaded after the operation ends.

Top

Examples

Example 1: Adding License Key Information from Prompt Input

```
ADDLICKEY   LICKEYINP(*PROMPT)  PRDID(1MYPROD)
            LICTRM(V3)  SERIAL(1234567)
            PRCGRP(P20)  LICKEY(123456 7890AB CDEF12)
            USGLMT(30)  EXPDATE(*NONE)  VNDDTA(12345678)
```

This command uses prompting to add the software license key information for feature 5001 of the product 1MYPROD to the license repository. The license term is Version 3. The license allows 30 users to use the system with serial number 1234567. That system is in the processor group of 20 or less. There is no expiration date on the license. Because the product is installed on a system with the specified serial number, the license also is installed.

Example 2: Adding All License Key Information from File Input

```
ADDLICKEY   LICKEYINP(*LICKEYFILE)  SERIAL(*ALL)
            LICKEYFILE(*LIBL/MYKEYFILE)  LICKEYMBR(*LAST)
```

This command adds the software license key information for all of the systems in the newest member of the file MYKEYFILE to the license repository. If the product is installed on the system, and the license is for this system, the license also is installed.

Example 3: Adding Local License Key Information from a License Key File

```
ADDLICKEY   LICKEYINP(*LICKEYFILE)  SERIAL(*LOCAL)
            LICKEYFILE(*LIBL/MYKEYFILE)  LICKEYMBR(*FIRST)
```

This command adds the software license key information found in the oldest member of the file MYKEYFILE to the license repository for this system only. If the product is installed on this system, the license is also installed.

Example 4: Adding Local License Key Information from Tape

```
ADDLICKEY   LICKEYINP(*TAPE)  DEV(TAP01)
```

This command searches the mounted volume on device TAP01 for the label QFILEPGMKEY. This data file is used and all software license keys for the local system are added to the repository. The tape is rewound after the operation.

Top

Error messages

***ESCAPE Messages**

CPF9E2D
Usage limit cannot be less than current usage.

CPF9E56
&1 license key information records added, &2 not added.

CPF9E6C
The license key cannot be used for processor group &2.

CPF9E6E
Product identifier &1 not valid.

CPF9E69
License key information not found in license key file.

CPF9E80
Error occurred during restoring license keys from tape.

CPF9E83
Expiration date &2 is not valid.

Top

Add Link (ADDLNK)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Link (ADDLNK) command adds a link to an object. The **New link (NEWLNK)** parameter specifies the name for the new link. The **Object (OBJ)** parameter specifies the current name of the object that is to receive the new link. After the link is established the object may be referred to by either the old name or the new name.

For more information about integrated file system commands, see the Integrated file system topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Restrictions:

- This command works on only one object. If a pattern is specified on the OBJ parameter and more than one object matches the pattern, the user can select the object from a list in an interactive job. If this is a batch job, the command will fail with error message CPFA08E "More than one name matches pattern.". A pattern is not allowed if the value for the **Link type (LNKTYPE)** parameter is *SYMBOLIC, because the object is not required to exist.
- The user must have write (*W) and execute (*X) authority to the directory that contains the new link. If a hard link is to be added, the user must also have object existence (*OBJEXIST) authority to the existing object and *X authority to each of the path name prefixes of the OBJ name.
- A hard link cannot be created to a symbolic link. When LNKTYPE(*HARD) is specified and the OBJ parameter names a symbolic link, the link is created to the resolved object (which must exist).
- A hard link cannot be created to a directory.
- A hard link cannot be created to an object in another file system.
- No links can be created in the QSYS.LIB, independent ASP QSYS.LIB or QDLS portion of the name space.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	*Path name*	Required, Positional 1
NEWLNK	New link	*Path name*	Required, Positional 2
LNKTYPE	Link type	***SYMBOLIC**, *HARD	Optional, Positional 3

Top

Object (OBJ)

Specifies the path name of the object the user wants to add a link to. This object must exist unless a symbolic link is to be added.

For more information on specifying path names, refer to ″Object naming rules″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Note: This parameter is Unicode-enabled. See ″Unicode support in CL″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for additional information.

Top

New link (NEWLNK)

Specifies the new path name that can be used to refer to the object. The new name must not exist.

For more information on specifying path names, refer to ″Object naming rules″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Note: This parameter is Unicode-enabled. See ″Unicode support in CL″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for additional information.

Top

Link type (LNKTYPE)

Specifies whether the link is hard or symbolic.

***SYMBOLIC**

The link to the object is a representation of a path name. This representation is in the form of a path contained in a file. The actual path is determined by doing a path search based on the contents of the file. A symbolic link is also called a soft link.

Symbolic links can cross file systems. An object need not exist. An existing object can be deleted without removing the symbolic link.

***HARD**

The link to the object is an actual path to an existing object. A hard link is established by creating a directory entry.

Hard links cannot cross file systems. When all hard links to an object are removed, the space occupied by the object is freed and the object can no longer be accessed. An object cannot be removed while a hard link to it exists.

Top

Examples

Example 1: Adding a Symbolic Link

```
ADDLNK   OBJ('DECEMBER-1994-MONTHLY-PAYROLL-FILE')
         NEWLNK('PAY')
```

This command adds a symbolic link named PAY to DECEMBER-1994-MONTHLY-PAYROLL-FILE.

Example 2: Adding a Symbolic Link to a Source File

```
ADDLNK   OBJ('/QSYS.LIB/MYLIB.LIB/F1.FILE/P1.MBR')
         NEWLNK('PGM1')  LNKTYPE(*SYMBOLIC)
```

This command adds a symbolic link named PGM1 from the user's directory (not in QSYS.LIB, independent ASP QSYS.LIB, or QDLS) to member P1 in source file F1 in QSYS.LIB.

Example 3: Adding a Hard Link

```
ADDLNK   OBJ('/QOpenSys/MYDIR/FILE1')  NEWLNK('FILE2')
         LNKTYPE(*HARD)
```

This command adds a hard link from the user's current directory, with the name FILE2, to FILE1 in /QOpenSys/MYDIR.

Top

Error messages

***ESCAPE Messages**

CPFA085
Home directory not found for user &1.

CPFA089
Pattern not allowed in path name.

CPFA08E
More than one name matches pattern.

CPFA093
Name matching pattern not found.

CPFA09C
Not authorized to object. Object is &1.

CPFA0A1
An input or output error occurred.

CPFA0A7
Path name too long.

CPFA0B0
Request not allowed to operate from one file system to another.

Top

Add Mounted FS (ADDMFS)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Mounted File System (ADDMFS) command makes the objects in a file system accessible to the integrated file system name space. The file system to be made accessible can be either a user-defined file system (*UDFS) on the local system or a remote file system accessed through a local Network File System client (*NFS). The directory that is the destination for the mount, the **Directory to mount over (MNTOVRDIR)** parameter, must exist.

This command can also be issued using the following alternative command name:

- MOUNT

For more information about Network File System commands, see i5/OS Network File System Support book, SC41-5714

Restrictions:

- The user must have input/output (I/O) system configuration (*IOSYSCFG) special authority to use this command.
- The user must have write (*W) authority to the directory to be mounted over.
- The user must have execute (*X) authority to each directory in the path.

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Type of file system	*NFS, *UDFS	Required, Key, Positional 1
MFS	File system to mount	*Path name*	Required, Key, Positional 2
MNTOVRDIR	Directory to mount over	*Path name*	Required, Key, Positional 3
OPTIONS	Mount options	*Character value*, ***DFT**	Optional
CCSID	Coded character set ID	*Element list*	Optional
	Element 1: Data file CCSID	1-65533, *ASCII, *JOBCCSID, ***BINARY**	
	Element 2: Path name CCSID	1-65533, ***ASCII**, *JOBCCSID	
CODEPAGE	Code page	*Element list*	Optional
	Element 1: Data file code page	1-32767, *ASCII, *JOBCCSID, ***BINARY**	
	Element 2: Path name code page	1-32767, ***ASCII**, *JOBCCSID	

Top

Type of file system (TYPE)

Specifies the type of file system to be mounted. The type of mount determines the correct form for the **File system to mount (MFS)** parameter.

***NFS** The file system specified for the MFS parameter is a Network File System. The MFS parameter must be of the form *hostname:pathname* where *hostname* can either be the name of a system or an IP address, and pathname must be an absolute path name.

***UDFS**

The file system specified for the MFS parameter is a user-defined file system. The MFS parameter must be in one of the two following forms:

- */dev/qaspXX/udfsname.udfs* where *XX* is one of the valid system or basic user auxiliary storage pool (ASP) numbers on the system, and *udfsname* is the name of the user-defined file system. All other parts of the name must appear as in the example above.
- */dev/aspname/udfsname.udfs*, where *aspname* is one of the valid independent ASP names on the system, and *udfsname* is the name of the user-defined file system. All other parts of the name must appear as in the example above.

The name part of the path must be unique within the specified *qaspXX* or *aspname* directory.

This is a required parameter.

Top

File system to mount (MFS)

Specifies the path name of the file system to be mounted. It can be the path to a local Block Special File (*BLKSF) or a remote NFS path name. See the **Type of file system (TYPE)** parameter to determine the correct format for the MFS parameter.

This is a required parameter.

Note: This parameter is Unicode-enabled. See ″Unicode support in CL″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for additional information.

Top

Directory to mount over (MNTOVRDIR)

Specifies the path name of the existing directory that the file system will be mounted over. This directory gets ’covered’ by the mounted file system. This directory must exist.

Multiple file systems can be mounted over the same directory, one on top of the other. However, only the topmost mounted file system is accessible, and the file systems must later be unmounted in the opposite order from which they were mounted (last-in first-out order).

This is a required parameter.

Note: This parameter is Unicode-enabled. See ″Unicode support in CL″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for additional information.

Top

Mount options (OPTIONS)

The options list contains a character string of mount options. The options are separated by commas. For some options, an equal '=' and a value follow the option. If an option is not specified, the default value for that option will be used. The options list may contain spaces.

***DFT** The default value for the options string for the mount of a Network File System (*NFS) is:

```
'rw,suid,retry=5,rsize=32768,wsize=32768,timeo=20,retrans=5,
acregmin=30,acregmax=60,acdirmin=30,acdirmax=60,hard,async,sec=sys,
vers=3:2,cachetimeo=60,nocache'
```

The default value for the options string for the mount of a user-defined file system (*UDFS) is:

```
'rw,suid'
```

For the mount of a Network File System, all of the following options are valid. For the mount of a user-defined file system, only the *ro*, *rw*, *suid* and *nosuid* options are valid. If options are specified that are not valid for the file system type to be mounted, they are ignored.

options-list

The following are the available options and their descriptions:

rw | ro This option specifies the protection for the mounted file system. Either *ro* (read-only) or *rw* (read-write) may be specified. If neither is specified, *rw* is assumed.

suid | nosuid

For the mount of a user-defined file system or a Network File System, if *suid* is specified, setuid execution is allowed. This means that bits other than the permission bits may be set. If *nosuid* is specified, setuid execution is not allowed.

hard | soft

For the mount of a Network File System, specifies whether NFS file systems are hard or soft mounted. Hard mounted means that operations on them are retried until they are acknowledged by the server. Soft mounted means that a timeout error is returned if a remote operation fails the number of times specified on the *retrans* option. If neither is specified, *hard* is assumed.

rsize=n

For the mount of a Network File System, specifies the size of the read buffer in bytes. The read buffer is used for data transfer between the NFS client and the remote NFS server on an NFS read request. The allowed range is 512 to 32768. If *rsize* is not specified, the default value of 32768 is assumed. For better performance, the read buffer should be a multiple of the the application buffer size.

wsize=n

For the mount of a Network File System, specifies the size of the write buffer in bytes. The write buffer is used for data transfer between the NFS client and the remote NFS server on an NFS write request. The allowed range is 512 to 32768. If *wsize* is not specified, the default value of 32768 is assumed. For better performance, the write buffer should be a multiple of the application buffer size.

timeo=n

For the mount of a Network File System, specifies the amount of time, in tenths of seconds, to wait for the client to respond on each try. The allowed range is 0 to 10000. If *timeo* is not specified, the default value of 20 tenths of a second (2 seconds) is assumed.

retry=n

For the mount of a Network File System, specifies the number of times to retry the mount operation. The allowed range is 0 to 10000. If *retry* is not specified, the default value of 5 retransmission attempts is assumed.

retrans=n

For the mount of a Network File System, specifies the number of times to retry the transmission to the server. The allowed range is 0 to 10. If *retrans* is not specified, the default value of 5 retransmission attempts is assumed.

acregmin=n

For the mount of a Network File System, specifies the minimum number of seconds to hold locally stored file attributes after file updates. The allowed range is 1 to 3600. If *acregmin* is not specified, the default value of 30 seconds is assumed.

acregmax=n

For the mount of a Network File System, specifies the maximum number of seconds to hold locally stored file attributes after file updates. The allowed range is 1 to 2,000,000,000. If *acregmax* is not specified, the default value of 60 seconds is assumed.

acdirmin=n

For the mount of a Network File System, specifies the minimum number of seconds to hold locally stored directory attributes after a directory update. The allowed range is 1 to 3600. If *acdirmin* is not specified, the default value of 30 seconds is assumed.

acdirmax=n

For the mount of a Network File System, specifies the maximum number of seconds to hold locally stored directory attributes after a directory update. The allowed range is 1 to 2,000,000,000. If *acdirmax* is not specified the default value of 60 seconds is assumed.

nocto For the mount of a Network File System, specifies whether to force the refresh of remote attributes when opening a file. If this option is specified, attributes are not refreshed from the server when opening a file, and changes are not sent to the server on the last close. If *nocto* is not specified, the default value of no suppression is assumed.

noac For the mount of a Network File System, specifies whether to suppress local storage of attributes and names. If this option is specified, local storage of attributes and names is suppressed. If *noac* is not specified, the default value of no suppression is assumed. If *noac* is specified, values specified for *agregmin*, *agregmax*, *agdirmin*, and *agdirmax* may be specified but are not used.

async For the mount of a Network File System, specifies whether to allow asynchronous write requests. If *async* is not specified, only synchronous write requests will be used.

cachetimeo=n

For the mount of a Network File System, specifies the amount of time in seconds, for the client to revalidate data cache consistency. The allowed range is 1 to 10000. If *cachetimeo* is not specified, the default value of 60 seconds is assumed.

nocache

For the mount of a Network File System, specifies whether to disable the data cache. If *nocache* is not specified, data cache will be enabled by default.

sec=flavor[:flavor...]

For the mount of a Network File System, specifies an ordered list of security flavors that may be used to access the mount point. Allowable flavor values are:

sys UNIX-like (user ids, group ids).

krb5 Kerberos 5, no integrity or privacy. Only valid when NFS version 4 specified.

krb5i Kerberos 5, with integrity. Only valid when NFS version 4 specified.

krb5p Kerberos 5, with privacy. Only valid when NFS version 4 specified.

vers=version[:version...]
For the mount of a Network File System, specifies NFS versions allowed for the mount. If this option is specified, only the specified NFS versions will be used to attempt the mount, in the order they are provided. The default used if this option is not specified is vers=3:2. Allowable version numbers are: 2:3:4.

Top

Coded character set ID (CCSID)

Specifies, for Network File Systems, a pair of coded character set identifiers (CCSIDs) to identify a specific character representation to be used. The first CCSID specifies what encoding scheme should be assumed for data files on the remote system. The second CCSID specifies what encoding scheme should be assumed for path names on the remote system.

This parameter is only valid if mounting a Network File System.

Element 1: Data file CCSID

***BINARY**
No conversion is used.

***ASCII**
The ASCII equivalent of the default job CCSID associated with the current job is used.

***JOBCCSID**
The CCSID from the default job CCSID is used.

1-65533
Specify a CCSID to be assumed for data files on the remote system.

Element 2: Path name CCSID

***ASCII**
The ASCII equivalent of the default job CCSID associated with the current job is used.

***JOBCCSID**
The CCSID from the default job CCSID is used.

1-65533
Specify a CCSID to be assumed for path names on the remote system. Only CCSIDs that can be converted into UCS-2 level 1 (1200) are supported. See Globalization information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for a list of supported conversions.

Top

Code page (CODEPAGE)

Specifies, for Network File Systems, a pair of code pages. The first code page specifies what code page should be assumed for data files on the remote system. The second code page specifies what code page should be assumed for path names on the remote system.

This parameter is only valid if mounting a Network File System.

Note: This parameter is replaced by **Coded character set ID (CCSID)** but the CODEPAGE parameter can still be used. However, because this parameter may be removed in a later release, whenever possible use the CCSID parameter.

Element 1: Data file code page

Note: A code page that has the same number of bytes per character as the original data should be specified.

***BINARY**
No conversion is used.

***ASCII**
The ASCII equivalent of the default job coded character set identifier (CCSID) associated with the current job is used.

***JOBCCSID**
The default job coded character set identifier (CCSID) associated with the current job is used.

1-32767
Specify a code page to be assumed for data files on the remote system. Only code pages that correspond to single-byte or double-byte encoding schemes are supported. Code pages that correspond to mixed-byte encoding schemes are not supported.

Element 2: Path name code page

***ASCII**
The ASCII equivalent of the default job coded character set identifier (CCSID) associated with the current job is used.

***JOBCCSID**
The default job coded character set identifier (CCSID) associated with the current job is used.

1-32767
Specify a code page to be assumed for path names on the remote system. Only code pages whose CCSIDs can be converted into UCS-2 level 1 (1200) are supported. See Globalization information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for a list of supported conversions.

Top

Examples

Example 1: Mounting a User-Defined File System

```
ADDMFS   TYPE(*UDFS)  MFS('/DEV/QASP03/PROD1')
         MNTOVRDIR('DIRB')
```

This command mounts a user-defined file system PROD1 over the directory, DIRB. It uses the defaults for the other parameters.

Example 2: Mounting a Network File System

```
ADDMFS   TYPE(*NFS)  MFS('RAINFALL:/QSYS.LIB/RAY.LIB')
         MNTOVRDIR('/mystuff')
```

This command mounts the **/qsys.lib/ray.lib** file system from the remote system RAINFALL into the directory **/mystuff**.

Example 3: Mounting a Network File System with OPTIONS

```
ADDMFS   TYPE(*NFS)   MFS('RAINFALL:/QSYS.LIB/RAY.LIB')
         MNTOVRDIR('/mystuff')
         OPTIONS('ro,nosuid,rsize=256, retrans=10')
         CODEPAGE(*ASCII *JOBCCSID)  CCSID(*ASCII *JOBCCSID)
```

This command mounts the **/qsys.lib/ray.lib** file system from the remote system RAINFALL into the directory **/mystuff**. In addition it specifies to mount as read-only, not allow setuid execution, set the read buffer to 256 bytes, and the retransmission attempts to 10. The job CCSID is used to determine the coded character set identifier to use for remote path names.

Top

Error messages

***ESCAPE Messages**

CPDBCC2
A non-recoverable error occurred when attempting to resolve the name.

CPF3BCB
Encoding scheme &1 of CCSID &2 not supported.

CPFA09C
Not authorized to object. Object is &1.

CPFA0A2
Information passed to this operation was not valid.

CPFA0A9
Object not found. Object is &1.

CPFA0D0
CCSID conversion error occurred.

CPFA0D9
Character string not converted.

CPFA1B0
Unrecognized option found on options list.

CPFA1B8
*IOSYSCFG authority required to use &1.

CPFA1C6
The value supplied for the File system to mount (MFS) parameter does not have the correct form.

CPFA1CE
Cannot find an address for the specified system name.

Top

Add Message Description (ADDMSGD)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Message Description (ADDMSGD) command describes a message and stores it in a message file for later use. The message description remains in the message file until the file is deleted or until the Remove Message Description (RMVMSGD) command is used to remove it from the file. To change any of the attributes of the message description, such as its message text or severity code, use the Change Message Description (CHGMSGD) command.

Note: A description of how to print a single message or a group of messages is in the section entitled *Handling Messages* in the Basic system operations topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Substitution variables can be embedded both in the first-level and second-level message text. They can be replaced later by message data fields specified in the Retrieve Message (RTVMSG), Send User Message (SNDUSRMSG), and Send Program Message (SNDPGMMSG) commands.

Note: The *type* of message being defined is *not* specified in the ADDMSGD command. The type is specified in the command that actually sends the message.

If the message second-level text exceeds 512 characters, it will not fit because of the i5/OS Prompter limit. In this case, enter the command on the Command Entry panel or in a CL program.

Restrictions:

- To add a message description to a message file, you must have use (*USE) and add (*ADD) authorities for the message file.

Top

Parameters

Keyword	Description	Choices	Notes
MSGID	Message identifier	*Name*	Required, Positional 1
MSGF	Message file	*Qualified object name*	Required, Positional 2
	Qualifier 1: Message file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MSG	First-level message text	*Character value*	Required, Positional 3
SECLVL	Second-level message text	*Character value*, ***NONE**	Optional, Positional 4
SEV	Severity code	0-99, **00**	Optional, Positional 5

Keyword	Description	Choices	Notes
FMT	Message data fields formats	Single values: ***NONE** Other values (up to 99 repetitions): *Element list*	Optional
	Element 1: Data type	*QTDCHAR, *CHAR, *HEX, *SPP, *DEC, *BIN, *UBIN, *CCHAR, *UTC, *UTCD, *UTCT, *DTS, *SYP, *ITV	
	Element 2: Length	*Integer*, ***VARY**	
	Element 3: *VARY bytes or dec pos	*Integer*, **0**	
TYPE	Reply type	***CHAR**, *DEC, *ALPHA, *NAME, *NONE	Optional
LEN	Maximum reply length	Single values: ***TYPE**, *NONE Other values: *Element list*	Optional
	Element 1: Length	*Integer*	
	Element 2: Decimal positions	*Integer*	
VALUES	Valid reply values	Single values: ***NONE** Other values (up to 20 repetitions): *Character value*	Optional
SPCVAL	Special reply values	Single values: ***NONE** Other values (up to 20 repetitions): *Element list*	Optional
	Element 1: Original from-value	*Character value*	
	Element 2: Replacement to-value	*Character value*	
RANGE	Range of reply values	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Lower value	*Character value*	
	Element 2: Upper value	*Character value*	
REL	Relationship for valid replies	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Relational operator	*EQ, *LE, *GE, *GT, *LT, *NE, *NL, *NG	
	Element 2: Value	*Character value*	
DFT	Default reply value	*Character value*, ***NONE**	Optional
DFTPGM	Default program to call	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Default program to call	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DMPLST	Data to be dumped	Single values: *NONE Other values (up to 102 repetitions): 1-99, ***JOB**, *JOBINT, *JOBDMP	Optional
LVL	Level of message	*Element list*	Optional
	Element 1: Creation date	*Date*, ***CURRENT**	
	Element 2: Level number	1-99, **1**	
ALROPT	Alert options	*Element list*	Optional
	Element 1: Alert type	*IMMED, *DEFER, *UNATTEND, ***NO**	
	Element 2: Resource name variable	1-99, ***NONE**	
LOGPRB	Log problem	***NO**, *YES	Optional
CCSID	Coded character set ID	1-65535, ***JOB**, *HEX	Optional

Top

Message identifier (MSGID)

Specifies the message identifier under which the message is stored in the message file. Every message must have an identifier, and every identifier in the message file must be unique.

This is a required parameter.

The message identifier must be 7 characters in length and in the following format: *pppnnnn*

The first 3 characters must be a code consisting of an alphabetic character followed by two alphanumeric (alphabetic or decimal) characters; the last 4 characters must consist of numbers ranging from 0 through 9 and characters ranging from A through F.

Top

Message file (MSGF)

Specifies the message file where the message is to be stored.

This is a required parameter.

Qualifier 1: Message file

name Specify the name of the message file where the message is to be stored.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the message file. If no current library entry exists in the library list, QGPL is used.

name Specify the library where the message file is located.

Top

First-level message text (MSG)

Specifies the text of the message being defined. This text is the message that is initially shown or printed, or sent to a program or log. A maximum of 132 characters enclosed in apostrophes can be specified, but the limitations of the display stations (their screen size) should be considered. The entire message must be enclosed in apostrophes if blanks are included in the message. To code an apostrophe for use in the message, enter a double apostrophe.

This is a required parameter.

Double-Byte Character Set Considerations

When entering double-byte characters on this parameter, several combinations of characters may cause errors to occur on this command. If double-byte characters contain the string, X'50Fn' (where n is a 1-digit number ranging from 0 through 9), error messages CPF2424 or CPF2431 may result.

Coded Character Set Identifier (CCSID) Considerations

The text supplied for the MSG parameter is assumed to be in the CCSID of the job running this command unless the CCSID parameter is coded. If the CCSID parameter is coded, the text is assumed to

be in the CCSID specified. For more information about the message handler and its use of CCSIDs, see the i5/OS globalization topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Top

Second-level message text (SECLVL)

Specifies the message help that is shown to a display station user to further explain the message specified for the **First-level message text (MSG)** parameter. The user presses the Help key to request the message help. Message help can also be written to the job log if *SECLVL is specified for the **Log in service log (LOG)** parameter of the job commands.

***NONE**
There is no message help for this message description.

'second-level-text'
Specify the message help that is shown when it is requested by the user. No more than 3000 characters enclosed in apostrophes can be specified, but display limitations must be considered.

Message help can be formatted for the work station using three format control characters. Each must be followed by a blank.

- **&N** Forces the message help to a new line (column 2). If the help is longer than one line, the next lines are indented to column 4 until the end of the help or until another format control character is found.
- **&P** Forces the message help to a new line, indented to column 6. If the help is longer than one line, the next lines start in column 4 until the end of the help or until another format control character is found.
- **&B** Forces the message help to a new line, starting in column 4. If the help is longer than one line, the next lines are indented to column 6 until the end of the help or until another format control character is found.

Double-Byte Character Set Considerations

When entering double-byte characters on this parameter, several combinations of characters may cause errors to occur on this command. The double-byte characters should not contain the string, X'50Fn' (where n is a 1-digit number, 0-9) or error messages CPF2424 or CPF2431 may result. Examples are: X'50F0', X'50F4', X'50F9'.

Coded Character Set Identifier (CCSID) Considerations

The text supplied for the SECLVL parameter will be assumed to be in the CCSID of the job running this command unless the CCSID parameter is coded. If the CCSID parameter is coded, the text will be assumed in the CCSID specified. For more information about message handler and its use of CCSIDs, see the i5/OS globalization topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Top

Severity code (SEV)

Specifies the severity code of the message being defined. The severity code indicates the severity level of the condition that causes the message to be sent. (99 is the most important severity.)

00 The severity code assigned to this message is 00. The message is an informational message.

severity-code
Specify a value, ranging from 00 through 99, as the severity level associated with this message.

Message data fields formats (FMT)

Specifies the formats of up to 99 message data fields. Each field is described in this parameter by a list of attributes. These attributes specify the type of data in the field, the total length of the field, and, optionally, the number of decimal digits to the right of the decimal point. Certain data types do not require a length field. Boundary alignment requirements must be considered (for example, pointers are always aligned on 16-byte boundaries).

All 99 of the message data fields can be used as substitution values in the message and message help defined in this message description. They can also be specified for the **Data to be dumped (DMPLST)** parameter and the **Alert options (ALROPT)** parameter of this command.

Single values

***NONE**
No format is being described for message fields. If *NONE is specified, or if this parameter is omitted, no message data fields can be referred to in the **First-level message text (MSG)**, **Second-level message text (SECLVL)**, **Data to be dumped (DMPLST)**, or **Alert options (ALROPT)** parameters.

type [length [decimal-positions]]

Element 1: Data type

The first element specifies the type of data the substitution field contains and how the data is formatted when substituted in the message text. The contents of the second and third elements vary depending on the type specified. One of the following types can be specified for each field described on this parameter:

***QTDCHAR**
A character string formatted with enclosing apostrophes ('Monday, the 1st') is specified.

***CHAR**
A character string formatted without enclosing apostrophes is specified. It is an alphanumeric string that can be used, for example, to specify a name (BOB). Trailing blanks are truncated.

***HEX** A string of bytes formatted as a hexadecimal value (X'C0F4') is specified.

***DEC** A packed decimal number that is formatted in the message as a signed decimal value with a decimal point is specified. Values for length (required) and decimal positions (optional) are specified for this type (*DEC) to indicate the number of decimal digits and the number of digits to the right of the decimal point. Zeros to the left of the first significant digit are suppressed, and leading blanks are truncated (removed). If a decimal position other than zero is specified, a decimal point is shown in the result even if the decimal precision in the result is zeros; examples are 128.00 and 128.01 if FMT(*DEC 5 2) is specified. If the number of decimal positions is not specified, zero is assumed. The following gives two examples:

- If FMT(*DEC 2) is specified for a substitution field and the message data is a packed decimal value of X'058C', the message text contains a positive value of 58 with no decimal point indicated.
- If FMT(*DEC 4 2) is specified and the packed value is specified as X'05810C' (3 bytes long), the text contains the formatted decimal value of 58.10.

***BIN** A binary value that is either 2, 4 or 8 bytes long (B'0000 0000 0011 1010') and is formatted in the message as a signed decimal value (58) is specified.

***UBIN**
A binary value that is either 2, 4 or 8 bytes long (B'0000 0000 0011 1010') and is formatted in the message as an unsigned decimal value (58) is specified.

***CCHAR**
A character string that can be converted. If data of this type is sent to a message queue that has a CCSID tag other than 65535 or 65534, the data is converted from the CCSID specified by the send function to the CCSID of the message queue. Conversions can also occur on data of this type when the data is obtained from the message queue using a receive or display function. See the Message Handler section of the i5/OS globalization topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for more details on CCSID conversions.

***UTC** An 8-byte field that contains a system date time stamp in Coordinated Universal Time (UTC) is specified. The output formatted date time stamp contains the date followed by one blank separator and the time. The date and time in the output message are adjusted from UTC using the time zone specified for the job. The date is formatted using the job definition attributes for date format and date separator. The time is formatted using the job definition attribute for time separator. When the 8-byte field is passed as hex zero (X'0000000000000000') the value will be formatted as *N.

***UTCD**
An 8-byte field that contains a system date time stamp in Coordinated Universal Time (UTC) is specified. The date in the output message is adjusted from UTC by the time zone specified for the job. The date is formatted using the job definition attributes for date format and date separator. When the 8-byte field is passed as hex zero (X'0000000000000000') the value will be formatted as *N.

***UTCT**
An 8-byte field that contains a system date time stamp in Coordinated Universal Time (UTC) is specified. The time in the output message is adjusted from UTC by the time zone specified for the job and is formatted using the job definition attribute for time separator. When the 8-byte field is passed as hex zero (X'0000000000000000') the value will be formatted as *N.

***DTS** An 8-byte field that contains a system date time stamp is specified. The date time stamp contains the date followed by one blank separator and the time. The date is formatted in the output message using the job definition attributes for date format and date separator. The time is formatted using the job definition attribute for time separator.

***ITV** An 8-byte binary field that contains the time interval (in seconds) for wait time-out conditions is specified. The time interval is formatted in the message as a zero-suppressed zoned decimal value (15 0) representing the number of seconds to wait.

The following formats are valid only in IBM-provided message descriptions and should not be used for other messages.

***SPP** A 16-byte space pointer to data in a space object is specified. When referred to in the DMPLST parameter, the data in the space object (from the offset indicated by the pointer) for the length specified, is dumped. *SPP is not valid as a replacement field in message text.

***SYP** A 16-byte system pointer to a system object is specified. When referred to in message text, the simple name of the system object is formatted as described in the name type, *CHAR. When referred to on the **Data to be dumped (DMPLST)** parameter, the object itself is dumped.

Element 2: Length

Following the type specification, a second element can be specified to indicate the number of characters or digits that are passed in the message data. How the second element is used depends on the type specified in the first element.

1. If a length is not specified for *QTDCHAR, *CHAR, *HEX, or *SPP, then *VARY is assumed for the length. If *VARY is specified or assumed, the message data field passed by the SNDUSRMSG or

SNDPGMMSG commands must be preceded by a 2-byte or 4-byte binary field that indicates the actual number of bytes of data being passed. However, when *SPP is specified, the length field is contained in the first bytes pointed to by the space pointer. Therefore, the 2-byte or 4-byte field must precede the data pointed to by the space pointer, and *not* precede the space pointer that is passed as part of the message data.

2. If the type *DEC is specified, the total number of decimal digits (including the fraction) *must* be specified as the second value; the number of digits in the fraction optionally can be specified (optional) as the third value.
3. If the type *BIN or *UBIN is specified, the message data field can be only 2, 4 or 8 bytes long; the default is 2 bytes.
4. If the type *CCHAR is specified, the message data length field can only be *VARY. A variable length field is required because as the data in this field gets converted to different coded character set identifiers (CCSIDs), its length may change.

Element 3: *VARY bytes or dec pos

The third element is used in one of two ways, depending on the type specified in the first element: (1) If *QTDCHAR, *CHAR, *CCHAR, *HEX, or *SPP is specified, and if *VARY is specified or assumed for the second element, the third element is used with *VARY to indicate the size of the length field actually passed. The third element can be either a 2 or a 4, which is the number of bytes used to specify the length (in binary) of the passed value. (2) If *DEC is specified, the third element indicates the number of decimal positions in the decimal value. If not specified, the default is 0 decimal positions.

Note: If an object has been damaged or deleted, the substitution variable is not replaced by the object name when it is displayed; instead, the variable appears as &n, where n = number. Also, if the length of the message data that is passed to the substitution variable is shorter than the length specified, the substitution value becomes a null field.

Top

Reply type (TYPE)

Specifies the type of reply that can be made to an inquiry or notify message.

***CHAR**
Any character string is valid. If it is a quoted character string, the apostrophes are passed as part of the character string.

***NONE**
No reply type is specified. *NONE must also be specified for the **Maximum reply length (LEN)** parameter.

***DEC** Only a decimal number is a valid reply.

***ALPHA**
Only an alphabetic string is valid. Blanks are not allowed.

***NAME**
Only a simple name is a valid reply. The name does not have to be an object name, but it must start with an alphabetic character; the remaining characters must be alphanumeric.

Top

Maximum reply length (LEN)

Specifies the maximum length of a reply to an inquiry or notify message. The values specified apply *only* if one or more of the other validity checking parameters are specified. If none of the validity checking parameters are specified, the reply can contain up to 132 characters.

Single values

***TYPE** The maximum length is determined by the type of reply specified for the **Reply type (TYPE)** parameter The maximum length for each type of reply is:

- Up to 32 characters can be specified for types *CHAR and *ALPHA (132 characters if no additional validity checking is being performed).
- Up to 15 digits are specified for *DEC, of which a maximum of 9 digits can be to the right of the decimal point.
- Up to 10 alphanumeric characters are specified for *NAME.

***NONE**
No reply length is specified. *NONE must also be specified for the **Reply type (TYPE)** parameter.

Element 1: Length

length Specify the maximum length allowed for the message reply.

Element 2: Decimal positions

decimal-positions
If *DEC is specified for the **Reply type (TYPE)** parameter, specify the number of decimal positions allowed for the message reply.

Top

Valid reply values (VALUES)

Specifies a list of values of which one can be received as a valid reply to an inquiry or notify message. No more than 20 values can be specified in the list. Each value in the list must meet the requirements specified for message replies for the **Reply type (TYPE)** parameter and the **Maximum reply length (LEN)** parameter. If this parameter is specified, the **Range of reply values (RANGE)** parameter and the **Relationship for valid replies (REL)** parameter cannot be specified.

Single values

***NONE**
No list of reply values is specified.

Other values

compare-value
Specify a list of up to 20 values to compare with a reply value that is sent in response to the message defined in this message description. The maximum length of each value is 32 characters.

Top

Special reply values (SPCVAL)

Specifies a list of up to 20 sets of special values of which one set is used as the reply to an inquiry or notify message. The reply sent is compared to the from-value in each set; if a match is found, and a to-value was specified in that set, the to-value is sent as the reply. If no to-value was specified, the from-value is sent as the reply.

The to-value must meet the requirements specified on the **Reply type (TYPE)** parameter and the **Maximum reply length (LEN)** parameter.

Single values

***NONE**
No special values are specified for the replies to this message.

Element 1: Original from-value

from-value
Specify a from-value to compare to a message reply value.

Element 2: Replacement to-value

to-value
Specify a to-value that the from-value will be mapped to before the reply is sent.

Top

Range of reply values (RANGE)

Specifies the upper and lower value limits for valid replies sent to an inquiry or notify message. These values must meet the requirements specified on the **Reply type (TYPE)** parameter and the **Maximum reply length (LEN)** parameter, and both values must be of the same type.

If this parameter is specified, the **Valid reply values (VALUES)** parameter and the **Relationship for valid replies (REL)** parameter cannot be specified.

Single values

***NONE**
No range values are specified for the replies to this message.

Element 1: Lower value

lower-value
Specify the lower limit value for valid replies to this message.

Element 2: Upper value

upper-value
Specify the upper limit value for valid replies to this message.

Top

Relationship for valid replies (REL)

Specifies the relationship that must be met for a valid reply to an inquiry or notify message. The value specified must meet the requirements specified for replies on the **Reply type (TYPE)** parameter and the **Maximum reply length (LEN)** parameter.

If this parameter is specified, **Valid reply values (VALUES)** parameter and the **Range of reply values (RANGE)** parameter cannot be specified.

Single values

***NONE**
No relationship values are specified for the replies to this message.

Element 1: Relational operator

operator-value
Specify one of the relational operators and the value against which the message reply is validity checked.

- *LT — Less than
- *LE — Less than or equal to
- *GT — Greater than
- *GE — Greater than or equal to
- *EQ — Equal to
- *NL — Not less than
- *NG — Not greater than
- *NE — Not equal to

Element 2: Value

value Specify the value to be used to compare against a message reply value.

Top

Default reply value (DFT)

Specifies, if the message is sent as an inquiry or notify message, the default reply that is used when the receiver of the message has indicated that all incoming messages are to receive default replies, or when a message is deleted from a message queue and no reply is specified. The default reply must meet the requirements specified for replies on the **Reply type (TYPE)** parameter and the **Maximum reply length (LEN)** parameter.

***NONE**
No default reply is specified for the replies to this message.

'default-reply'
Specify the default reply to send (enclosed in apostrophes if it contains special characters) to inquiry or notify messages.

Top

Default program to call (DFTPGM)

Specifies the qualified name of any default program called to take default action if this message is sent as an escape message to a program or procedure that is not monitoring for it. This parameter is ignored if the message is *not* sent as an escape message. If the message *is* sent as an escape message, the following parameters are passed to the specified default program:

- The name of the program or procedure to which the message is sent (277 characters). The program name, module name, procedure name, and program type of the call message queue to which the message is sent. This is the same name as the program or procedure that did not monitor for the escape message.

Characters 1 through 10 are the name of the program to which the message is sent.

Characters 11 through 20 are the name of the module to which the message is sent. If the message is not sent to an ILE procedure, the value *N is returned in this field padded on the right with blanks.

Characters 21 through 276 are the name of the procedure to which the message is sent. If the message is not sent to an ILE procedure, the value *N is returned in this field padded on the right with blanks.

Character 277 is set to the value 1 if the message is sent to an ILE procedure, or to the value 0 if the message is not sent to an ILE procedure.

- Message reference key (4 characters). The message reference key of the escape message on the program message queue.

Single values

***NONE**
No default program is specified for this message.

Qualifier 1: Default program to call

name Specify the name of the default program that is called when an escape message is sent.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the program. If no current library entry exists in the library list, the QGPL library is used.

name Specify the library where the program is located.

Top

Data to be dumped (DMPLST)

Specifies the data that is dumped when this message is sent as an escape message to a program that is not monitoring for it. This parameter can specify that data related to the job be dumped, that data from message data fields be dumped, or that a combination of these be dumped.

Single values

***NONE**
There is no dump list for this message. No dump occurs.

Other values

***JOB** This value is the equivalent of specifying * for the **Job name (JOB)** parameter and *PRINT for the **Output (OUTPUT)** parameter of the Display Job (DSPJOB) command.

***JOBDMP**
The data areas of the job are dumped as specified by the Dump Job (DMPJOB) command. *JOBDMP can be specified by itself, with *JOB, with *JOBINT, or with a list of message data field numbers.

***JOBINT**
The internal machine data structures, related to the machine process in which the job is running, are dumped to the machine error log. *JOBINT can be specified by itself, with *JOBDMP, *JOB, or with a list of message data field numbers.

message-data-field-number
Specify the numbers of the message data fields that identify the data that is dumped when this escape message is sent but not monitored. A maximum of 99 data field numbers can be specified.

Top

Level of message(LVL)

Specifies the level identifier of the message description being defined. The level identifier is made up of the date on which the message is defined and a 2-digit number that makes the identifier unique.

Element 1: Creation date

***CURRENT**
The current date is used as the first part of the message description level identifier.

creation-date
Specify the date on which the message description is being defined.

Element 2: Level number

1 The number 1 is used as the second part of the message description level identifier.

1-99 Specify a number (ranging from 1 through 99) that makes the level identifier of the message description unique.

Top

Alert options (ALROPT)

Specifies the alert option associated with messages sent to the system operator message queue (QSYSOPR). Alerts can be used to send a message to the host system indicating that an error has occurred on this system.

Element 1: Alert type

***NO** No alert is sent.

***IMMED**
An alert is sent immediately, simultaneous with sending the message to QHST or QSYSOPR.

***UNATTEND**
An alert is sent immediately only when *UNATTEND is specified for the **Alert status (ALRSTS)** parameter of the Change Network Attributes (CHGNETA) command.

***DEFER**
The alert is sent after local problem analysis. *DEFER should be specified only for those messages against which problem analysis can be run. An alert is sent at the first exit from problem analysis for the problem referred to by the message. All alerts set to *DEFER are treated as *IMMED if:

- *UNATTEND is specified for the **Alert status (ALRSTS)** parameter of the Change Network Attributes (CHGNETA) command.
- An error log ID is not available for a problem that might be resolved using problem analysis.
- *NO is specified for the **Log problem (LOGPRB)** parameter (problem analysis is not available for the condition reported by the message).

Element 2: Resource name variable

***NONE**
No message data field format number is passed with the alert identifier.

1-99 Specify the message data field format number that is passed with the alert identifier.

Top

Log problem (LOGPRB)

Specifies, for IBM-supplied messages, whether an entry is put into the problem log. If there is an error log ID for the message and *YES is specified for this parameter, the user can request problem analysis by pressing F14 from the system operator message queue display (by running the DSPMSG *SYSOPR command).

***NO** An entry is not put in the problem log.

***YES** An entry is put in the problem log if there is an error log ID associated with this message.

Top

Coded character set ID (CCSID)

Specifies the coded character set identification (CCSID) that the text supplied for the MSG and SECLVL parameters is in. If the message file that this message description is being added to is not 65534 or 65535, the text supplied is converted from the CCSID specified to the CCSID of the message file. Otherwise, the text is not converted but the CCSID is saved in case a conversion is needed during a retrieve or display function. For more information about the message handler and its use of CCSIDs, see the i5/OS globalization topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

***JOB** The text for this message description is assumed to be in the CCSID of the job running this command.

***HEX** The text for this message description is not converted and is tagged 65535.

coded-character-set-identifier
Specify the CCSID you want the text to be considered in. Valid values range form 1 through 65535. See the Globalization information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for a list of valid CCSID values. Only CCSIDs that a job can be changed to are accepted.

Top

Examples

Example 1: Defining a Message

```
ADDMSGD   MSGID(UIN0115)  MSGF(INV)
          MSG('Enter the name of user''s department')
          SECLVL('Valid departments:  &B X12 &B X13 &B X14')
          TYPE(*CHAR)  LEN(3)  DFT('ZZZ')
```

This command defines a message and stores it in a file named INV under the identifier UIN0115. The message supplies second-level message text by using the &B formatting character to show the three valid department names (X12, X13, and X14) each on a separate line. The reply requires validity checking so that a valid reply can only be a 3-character identifier. A default reply of ZZZ is also provided.

Example 2: Defining a Message Description

```
ADDMSGD   MSGID(UPY0047)  MSGF(PAYLIB/TIMECARD)
          MSG('For week of &1, &2 time cards. Are there more?')
          FMT((*CHAR 8) (*CHAR 3))  TYPE(*ALPHA)  LEN(1)
          VALUES(N  Y)  SPCVAL((YES Y)(NO N))  DFT(N)
```

This command defines a message description that is stored in the TIMECARD message file in the PAYLIB library. The program that processes the time cards can send a message (as an inquiry type message) telling how many time cards (in &2) have been processed for the week (specified in &1). To send this message to a user via a message queue, the program must use the SNDPGMMSG or SNDUSRMSG commands. In this example, the command specifies:

- The message identifier of this message (UPY0047)
- The file (TIMECARD) that contains this message
- The time card date in 8 characters (such as 09/15/88); this must be the first value in the MSGDTA parameter
- The number of time cards in no more than 3 digits (such as 125)

If a reply of YES is sent, it is accepted as a Y (SPCVAL parameter). If NO is sent, it is accepted as an N. If neither YES nor NO is sent, the reply is checked for validity by the TYPE, LEN, and VALUES parameters. If the user chooses, no reply is sent and the default reply (N) is assumed.

Example 3: Defining an Escape Message

```
ADDMSGD   MSGID(UPY1234)  MSGF(PAYLIB/TIMECARD)
          MSG('Tax for employee &1 exceeds gross salary.')
          SEV(75)  FMT((*CHAR  6) (*DEC  9  2) (*CHAR  8))
          DFTPGM(PAYLIB/BADTAX)  DMPLST(1  2  3 *JOB)
```

This command defines an escape message. The sender of the message passes three data values, the first of which (employee serial number) is used as replacement text in the message. If this message is sent as an escape message and the program to which the message is sent does not monitor for message UPY1234, default system action is taken. This includes dumping the three data values that were passed and the job structure. After the dump is taken, program BADTAX is called.

See the Monitor Message (MONMSG) command for more about monitoring for messages.

Top

Error messages

***ESCAPE Messages**

CPF2401
Not authorized to library &1.

CPF2407
Message file &1 in &2 not found.

CPF2411
Not authorized to message file &1 in &2.

CPF2412
Message ID &1 already exists in message file &2 in &3.

CPF2430
Message description not added to message file

CPF2461
Message file &1 could not be extended.

CPF2483
Message file currently in use.

CPF2510
Message file &1 in &2 logically damaged.

CPF9830
Cannot assign library &1.

CPF9838
User profile storage limit exceeded.

Top

Add Master Key Part (ADDMSTPART)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The Add Master Key Part (ADDMSTPART) command loads a key part for the specified master key by hashing the specified passphrase and adding it into the new master key version.

You should securely store your passphrase outside the system so you can recover the master key should it become damaged.

For more information on master keys, refer to the Cryptographic Services Master Keys article in the Cryptographic Services section of the APIs topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Restrictions:

- You must have all object (*ALLOBJ) and security administrator (*SECADM) special authorities to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
MSTKEY	Master key	1-8, *ASP, *SAVRST	Required, Positional 1
PASSPHRASE	Passphrase	*Character value*	Required, Positional 2
PASSLEN	Length of passphrase	1-256, ***CALC**	Optional

Top

Master key (MSTKEY)

Specifies the master key on which to perform the action.

This is a required parameter.

The action will be performed on:

***ASP** The master key used for encrypting data stored on auxiliary storage pool (ASP) disk storage.

***SAVRST**
The master key used for encrypting all the other master keys on a SAVSYS operation.

1-8 One of the eight general purpose master keys.

Top

Passphrase (PASSPHRASE)

Specifies a text string. The text string will be hashed and then added into the new version of the master key using an exclusive-OR operation.

Note: If a specific passphrase is added twice, the value is actually removed from the new version because a value exlusive-OR'ed with itself results in a value of 0.

This is a required parameter.

character-value
Specify the passphrase. Each character may contain any valid binary value.

Top

Length of passphrase (PASSLEN)

Specifies the number of bytes of the value specified for **Passphrase (PASSPHRASE)** that are to be used to create the key part which will be added to the master key.

***CALC**
Allow the system to calculate the passphrase length by removing all trailing blanks.

1-256 Specify the length of passphrase typed above. If the length specified is greater than the length of the passphrase, the passphrase will be padded with binary zeros to the length specified. If the length specified is less than the length of the passphrase, the passphrase will be truncated.

Top

Examples

Example 1: Add a Key Part for a Master Key Specifying Length of Passphrase

```
ADDMSTPART   MSTKEY(3)
             PASSPHRASE('Twas brillig, and the slithy toves')
             PASSLEN(30)
```

This command takes the first 30 bytes from the PASSPHRASE parameter, hashes it down to 20 bytes, and adds it into the new version of Master Key 3.

Example 2: Add a Key Part for a Master Key Allowing the System to Calculate the Passphrase Length

```
ADDMSTPART   MSTKEY(*SAVRST)
             PASSPHRASE('Twas brillig, and the slithy toves')
             PASSLEN(*CALC)
```

This command removes all trailing blanks from the PASSPHRASE parameter value, takes the resulting text string, hashes it down to 20 bytes, and adds it into the new version of the Save/Restore Master Key.

Top

Error messages

***ESCAPE Messages**

CPF222E
&1 special authority is required.

CPF3CF2
Error(s) occurred during running of &1 API.

CPF9872
Program or service program &1 in library &2 ended. Reason code &3.

CPF9DDA
Unexpected return code &1 from cryptographic service provider &2.

Top

Add Nickname (ADDNCK)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Nickname (ADDNCK) command is used to add a nickname to the system distribution directory. The nickname must be unique if it is a public nickname. The nickname must be unique only for the owner if it is a private nickname.

A **nickname** is a short version of either a directory entry or a distribution list name. More information about nicknames is in the SNA Distribution Services book, SC41-5410.

Top

Parameters

Keyword	Description	Choices	Notes
NCK	Nickname	*Element list*	Required, Positional 1
	Element 1: Nickname	*Character value*	
	Element 2: Access	***PRIVATE**, *PUBLIC	
USRID	User identifier	*Element list*	Optional
	Element 1: User ID	*Character value*	
	Element 2: Address	*Character value*	
LSTID	List identifier	*Element list*	Optional
	Element 1: List ID	*Character value*	
	Element 2: List ID qualifier	*Character value*	
TEXT	Nickname description	*Character value*, ***DFT**	Optional

Top

Nickname (NCK)

Specifies the nickname to be added and the ability of users to access the nickname.

The possible nickname value is:

nickname
Specify the nickname you are adding.

The possible nickname access values are:

***PRIVATE**
The nickname cannot be shared with other users. It can be accessed and changed only by the owner.

***PUBLIC**
The nickname can be shared with other users. It can be accessed by any user on the local system, but it can be changed only by a user with security administrator (*SECADM) authority or by the owner.

This is a required parameter.

Top

User identifier (USRID)

Specifies the two-part user identifier for which a user nickname is being added. Both the user ID and address elements must be specified. More information about specifying the user ID and address is in the SNA Distribution Services book, SC41-5410.

Note: This parameter cannot be specified when the LSTID parameter is specified.

The possible user ID value is:

user-ID
Specify the user ID for this nickname. A maximum of 8 characters can be specified.

The Possible user address value is:

user-address
Specify the address for this nickname. A maximum of 8 characters can be specified.

This is a required parameter.

Top

List identifier (LSTID)

Specifies the two-part list identifier of the distribution list for which a list nickname is being added. Both the list identifier and qualifier elements must be specified.

Note: This parameter cannot be specified when the USRID parameter is specified.

The possible list identifier value is:

list-ID
Specify the list identifier (ID) of the distribution list.

The possible list qualifier value is:

list-ID-qualifier
Specify the list ID qualifier of the distribution list.

Note: The distribution list identifier has two parts, the ID and the qualifier, separated by at least one space. If lowercase characters are specified, the system changes them to uppercase.

The naming rules for the two-part list ID are identical to the rules for the user ID and address. A complete description of these rules is in the SNA Distribution Services book, SC41-5410.

This is a required parameter.

Top

Nickname description (TEXT)

Specifies the description of the nickname.

***DFT** The default description is used for the text. The default description is the first description associated with the specified USERID or LISTID parameter.

'nickname-description'
Specify a description to further identify the nickname. A maximum of 50 characters can be specified and must be enclosed in apostrophes.

Top

Examples

```
ADDNCK   NCK(SEC44A *PUBLIC)  USRID(XZWS44A RCHAS1)
         TEXT('Secretary for Department 44A')
```

This command adds a public user nickname which is a short version of the User ID and Address XZWS44A&rbl.RCHAS1. If this nickname is unique it is added to the directory.

Top

Error messages

***ESCAPE Messages**

CPF8360
Not enough storage for commitment control operation.

CPF8AA1
Library QUSRSYS not completely installed.

CPF905C
Error occurred trying to find a translation table.

CPF9838
User profile storage limit exceeded.

CPF9A89
Nickname function not successful.

Top

Add Network Job Entry (ADDNETJOBE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Network Job Entry (ADDNETJOBE) command adds a network job entry to the network job table on the system. The network job entry is used to determine the action that is taken when a input stream is sent to a user on this system using the Submit Network Job (SBMNETJOB) command.

This entry determines whether the input stream is automatically submitted, placed on the queue of network files for a user, or rejected. The entry also specifies the user profile that is used for checking the authority to the job description referred to by the input stream. There must be one entry for each user or distribution group who intends to submit jobs to this system. There is a network attribute, JOBACN (Job Action), that provides overall control of network job submission. Its value must be *SEARCH before the network job table is searched for an action. If the network attribute is *REJECT, all incoming jobs are rejected. If the network attribute is *FILE, all incoming network jobs are saved in the user's queue of network files regardless of any network job entry. The network attribute can be changed with the Change Network Attributes (CHGNETA) command.

Each network job entry is identified by the two-part user ID of the sender. When an input stream arrives, the user ID of the sending user is used to find a network job entry. If no entry is found, the second part of the user ID is used to find an entry, using *ANY for the first part. If this search fails, a search is made using *ANY for both parts of the user ID. If no entry is found, the job is rejected.

For additional information on the network job table, refer to the SNA Distribution Services book, SC41-5410.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The user must have *ALLOBJ (all object) authority.
3. The internal value for a node identifier may differ from the characters shown by the ADDNETJOBE command depending on the type of work station (language) being used. If the byte-string value specified for the FROMUSRID command parameter does not match the rules for an internal node identifier value, or if it does not match the internal value for any defined node (ignoring case differences), an error may be reported.

Top

Parameters

Keyword	Description	Choices	Notes
FROMUSRID	User ID	*Element list*	Required, Positional 1
	Element 1: User ID	*Character value*	
	Element 2: User ID qualifier	*Character value*	
ACTION	Network job action	*FILE, *REJECT, *SUBMIT	Required, Positional 2
SBMUSER	User profile	*Name*, **QUSER**	Optional, Positional 3

Keyword	Description	Choices	Notes
MSGQ	Message queue	Single values: ***USRPRF**, *NONE Other values: *Qualified object name*	Optional
	Qualifier 1: Message queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
JOBQ	Job queue	*Qualified object name*	Optional
	Qualifier 1: Job queue	*Name*, **QBATCH**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

User ID (FROMUSRID)

Specifies the two-part user ID of the user who submits an input stream to this system. Any input streams received from the user are handled as specified in this network job entry. Both parts of the user ID are required. A special value of *ANY can be entered for the first part or for both parts of the user ID.

Note: Depending on the type of work station being used, the internal value for a user identifier may differ from the characters shown by the Display Network Job Entry (DSPNETJOBE) command. If the byte-string value specified for the FROMUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.

This is a required parameter.

Top

Network job action (ACTION)

Specifies the action that is taken for an input stream controlled by this entry if the network attribute JOBACN is *SEARCH.

This is a required parameter.

***REJECT**
The input stream is rejected.

***FILE** The input stream is placed on the queue of network files for the user to whom the input stream is sent.

***SUBMIT**
The input stream is submitted to a batch job queue. The user profile specified in the network job entry is used to check for the required authority to the job queues.

Top

User profile (SBMUSER)

Specifies the user profile name under which the jobs are submitted. This user profile is used to check the authority to the job queues and job descriptions specified in the input stream. The value specified for this parameter will be effective if ACTION(*SUBMIT) is specified either on this command or on the Change Network Job Entry (CHGNETJOBE) command.

QUSER
The IBM-supplied user profile QUSER is used to submit the jobs.

user-profile
Specify the name of the user profile that is used to submit the jobs.

Top

Message queue (MSGQ)

Specifies the name and library of the message queue to which messages are sent after an input stream arrives. A message is also sent to the history log (QHST) when a input stream arrives.

***USRPRF**
The message queue of the user profile to whom the job was sent is used. This user is specified on the TOUSRID parameter of the SBMNETJOB command; this may or may not be the same user as is specified on the SBMUSER parameter of this command.

***NONE**
No message is sent to a user; however, a message is sent to the history log (QHST).

message-queue-name
Specify the name of the message queue that is used to receive messages.

The possible library values are:

***LIBL** The library list is used to locate the message queue.

***CURLIB**
The current library for the job is used to locate the message queue. If no library is specified as the current library for the job, QGPL is used.

library-name
Specify the library where the message queue is located.

Top

Job queue (JOBQ)

Specifies the job queue on which the job entries are placed. A job entry is placed on this queue for each job in the input stream that has JOBQ(*RDR) specified on the Batch Job (BCHJOB) command. If *RDR is not specified on the BCHJOB command, the job queue specified on the BCHJOB command or in the job description is used. (The job queue for each job in the input stream can be different.) This parameter is valid only if ACTION(*SUBMIT) is specified on this command, in the existing network job entry, or in a subsequent Change Network Job Entry (CHGNETJOBE) command.

The possible values are:

QBATCH
The job entry is to use job queue QBATCH.

job-queue-name
Specify the name of the job queue that is used.

The possible library values are:

***LIBL** The library list is used to locate the job queue.

***CURLIB**
The current library for the job is used to locate the job queue. If no library is specified as the current library for the job, QGPL is used.

library-name
Specify the library where the job queue is located.

Top

Examples

Example 1: Submitting Input Streams Automatically

```
ADDNETJOBE   FROMUSRID(JOHN SMITH)  ACTION(*SUBMIT)
             SBMUSER(ANDERSON)  JOBQ(QGPL/QPGMRL)
```

This command adds a network job entry that is used to determine the action that is taken for any input stream received from a user with a user ID of JOHN SMITH. The input streams are submitted automatically. User profile ANDERSON is used to check the authority to the job queues and job descriptions specified in the input stream. Messages are sent to the message queue specified in the user profile of the user to whom the input stream was sent. If no job queue is specified in either the received // BCHJOB command or the referenced job description, the jobs are placed on job queue QPGMRL in the QGPL library.

Example 2: Sending Messages to Specific Message Queue

```
ADDNETJOBE   FROMUSRID(*ANY JONES)  ACTION(*FILE)
             MSGQ(BROWN) SBMUSER(ANDERSON)
```

This command adds a network job entry that is used to determine the action taken for any input stream received from any user with JONES as the second part of the user ID for whom there is not a specific network job entry. The input stream is placed on the queue of received files for the user to whom the job was sent, and a message is sent to message queue BROWN.

Top

Error messages

***ESCAPE Messages**

CPF8050
Network job table could not be accessed.

CPF8051
*ANY not correct for second part of user ID.

CPF8052
Network job entry &1 &2 not added.

CPF9040
Wrong characters used in User ID or address, or List identifier &1 &2.

Top

Add Network Table Entry (ADDNETTBLE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Network Table Entry (ADDNETTBLE) command is used to add a network entry to the network table. You can use the network table to manage a list of your networks and their associated internet addresses.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
NETWORK	Network	*Character value*	Required, Positional 1
INTNETADR	Internet address	*Character value*	Required, Positional 2
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional
ALIAS	Aliases	Single values: ***NONE** Other values (up to 4 repetitions): *Character value*, ***NONE**	Optional

Top

Network (NETWORK)

Specifies the network to be added to the table.

Note: The combination of the values on the NETWORK and INTNETADR parameters must be unique.

character-value
Specify the name of the network to be added.

Top

Internet address (INTNETADR)

Specifies the internet address of the network. Internet addresses are expressed in the decimal form
`nnn.nnn.nnn.nnn`

where *nnn* is a number ranging from 0 through 255.

character-value
Specify the internet address of the network.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the network entry.

***BLANK**
Text is not specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Aliases (ALIAS)

Specifies the alternate name for the network. You can specify a maximum of 4 aliases. No checking is done to ensure that an alias is unique.

***NONE**
The network has no alternate name.

character-value
Specify an alternate network name.

Top

Examples

```
ADDNETTBLE   NETWORK(NETONE)  INTNETADR(9.5.0.0)
```

This command adds an entry for the network NETONE to the network table. The internet address for NETONE is 9.5.0.0.

Top

Error messages

***ESCAPE Messages**

TCP1901
Internet address &2 not valid.

TCP290C
Network entry already exists in table. Entry was not added.

TCP2916
Network entry contains characters that are not valid. Entry was not added.

TCP8050
*IOSYSCFG authority required to use &1.

Top

Add Node List Entry (ADDNODLE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Node List Entry (ADDNODLE) command adds a new entry to an existing node list object.

Top

Parameters

Keyword	Description	Choices	Notes
NODL	Node list	*Qualified object name*	Required, Positional 1
	Qualifier 1: Node list	*Name*	
	Qualifier 2: Library	*Name*, *CURLIB, ***LIBL**	
RMTLOCNAME	Remote location	*Element list*	Optional
	Element 1: Name or address	*Character value*	
	Element 2: Type	***SNA**, *IP	
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional
CPNAME	SNA node name	Single values: ***RMTLOC** Other values: *Element list*	Optional
	Element 1: Network identifier	*Communications name*, *NETATR	
	Element 2: Control point	*Communications name*	

Top

Node list (NODL)

Specifies the qualified name of the node list object to which the entry is added.

The node list name can be qualified by one of the following library values:

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name
Specify the name of the library to be searched.

The possible values are:

node-list-name
Specify the name of the node list to which the entry is added.

Top

Remote location (RMTLOCNAME)

Specifies the name and address type of the system to add to the list object. The name can be an SNA network ID and control point name, an internet protocol host name, or an internet address.

An SNA node name is specified using the format nnnnnnnn.cccccccc, where nnnnnnnn is the network ID and cccccccc is the control point name. If only the control point name is specified, the local network ID (LCLNETID) network attribute is used as the value of the network identifier (ID) of the system being added to the node list. If the LCLNETID network attribute is changed, the new value does not affect the existing entries.

A host name must follow these conventions:

- The first character must be either an English alphabetic character or a numeric character.
- The last character must be either an English alphabetic character or a numeric character.
- Blanks ( ) are not allowed.
- The special characters, period (.) and minus (-), are allowed.
- Parts of the name separated by periods (.) cannot exceed 63 characters in length.
- Internet address names (in the form nnn.nnn.nnn.nnn) are not allowed.
- Names must be from 1 to 255 characters in length.

The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

The possible Name or Address value is:

remote-location-name
Specify the remote location name to add to the node list.

The possible Address Type values are:

***SNA** The node name has a Systems Network Architecture (SNA) address type.

***IP** The node name has an Internet Protocol (IP) address type.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the node list entry.

The possible values are:

***BLANK**
Text is not specified.

'description'
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

SNA node name (CPNAME)

Specifies the SNA node name that is being added to the node list object. This system is specified as two elements: the network ID and the control point name.

Notes:

1. The RMTLOCNAME parameter is recommended for use in specifying the network ID and the control point name.
2. When the RMTLOCNAME parameter is used to specify the name of a system to add to the node list, *RMTLOC must be specified for this parameter.

The possible values are:

***RMTLOC**
The network ID and control point name are specified using the RMTLOCNAME parameter.

The possible Network ID values are:

***NETATR**
The local network ID (LCLNETID) network attribute is used as the value of the network identifier (ID) of the system being added to the node list. If the LCLNETID network attribute is changed, the new value does not affect the existing entries.

network-ID
Specify the network ID of the system being added to the node list.

The possible Control Point Name value is:

control-point-name
Specify the control point name of the system being added to the node list.

Note: This field is left blank when *RMTLOC is specified as the network ID.

Top

Examples

Example 1: Adding a System in the Local Network to a Node List

```
ADDNODLE   NODL(MYLIB/NODL02)  RMTLOCNAME(AS400A01 *SNA)
           TEXT('THE FIRST ISERIES 400 SERVER IN MY NETWORK')
```

This command adds an entry for system AS400A01, which is in the local network, to the node list NODL02 in library MYLIB. The entry has an address type of SNA. The text description for the entry is 'THE FIRST ISERIES 400 SERVER IN MY NETWORK'.

Example 2: Adding a Host Name to a Node List

```
ADDNODLE   NODL(MYLIB/NODL02)  RMTLOCNAME(MYSYS.NET1.LOCAL *IP)
           TEXT('SYSTEM AT HEADQUARTERS')
```

This command adds an entry for host name MYSYS.NET1.LOCAL to the node list NODL02 in library MYLIB. The entry has an address type of IP. The text description for the entry is 'SYSTEM AT HEADQUARTERS'.

Example 3: Adding an Internet Address to a Node List

```
ADDNODLE   NODL(MYLIB/NODL02)  RMTLOCNAME('9.13.156.8' *IP)
           TEXT('MINNEAPOLIS OFFICE')
```

This command adds an entry for internet address 9.13.156.8 to the node list NODL02 in library MYLIB. The entry has an address type of IP. The text description for the entry is 'MINNEAPOLIS OFFICE'.

Top

Error messages

***ESCAPE Messages**

CPF7AD4
Network ID &1 not in correct format.

CPF7B18
Control point &1 not in correct format.

CPF813E
Node list &4 in &9 damaged.

CPF96B3
Node list entry already exists.

CPF96B5
Remote location name not in correct format.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9830
Cannot assign library &1.

Top

Add Server Storage Link (ADDNWSSTGL)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Network Server Storage Link (ADDNWSSTGL) command is used to add a network storage space link to a network server description.

When *IXSVR is specified for the Network server type (TYPE) parameter in CRTNWSD command:

1. A maximum of 16 network server storage spaces can be linked to a network server description when using standard linking (DYNAMIC *NO specified).

   An additional 32 storage spaces may be dynamically linked to a Windows 2000 server or later with 16 of the 32 links being linked with *UPDATE access (read/write to only one server). The remaining 16 links are available for *SHRUPD access (read/write to multiple clustered Windows Server 2000 or later servers). The first of these *SHRUPD access links must be to a quorum resource disk (format *NTFSQR) at DRVSEQNBR *QR.

   This is only valid when ***WIN32 or WIN64** is specified for the **Network server type (TYPE)** parameter.
2. A maximum of 64 network server storage spaces can be linked to a network server description when using dynamic linking (DYNAMIC *YES specified). The first two storage spaces (1 and 2) must be linked with *UPDATE access (read/write). This is only valid when ***LINUX32** is specified for the **Network server type (TYPE)** parameter.

When *ISCSI is specified for the Network server type (TYPE) parameter in CRTNWSD command:

A maximum of 64 network server storage spaces can be linked to a network server description when using dynamic linking. The first two storage spaces (1 and 2) must be linked with *UPDATE access (read/write).

1. Storage spaces 3-63 can be linked with *UPDATE or *SHRUPD access mode. The last of the 64 storage spaces (64th) must be *SHRUPD access to a quorum resource (format *NTFSQR) at DRVSEQNBR *QR when ***WIN32 or WIN64** is specified for the **Network server type (TYPE)** parameter.
2. Storage spaces 3-64 can be linked with *UPDATE or *SHRUPD access mode when ***LINUX32** is specified for the **Network server type (TYPE)** parameter.

When *GUEST is specified for the Network server type (TYPE) parameter in CRTNWSD command:

1. A maximum of 64 network server storage spaces can be linked in any access mode to a network server description when using dynamic linking.

For more information about adding a link to a specific server type, see the following:

- For **Server connection** specified *IXSVR or *ISCSI for the **Network server type (TYPE)** parameter in NWSD object - System i integration category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ book.

Top

Parameters

Keyword	Description	Choices	Notes
NWSSTG	Network server storage space	*Name*	Required, Positional 1
NWSD	Network server description	*Communications name*	Required, Positional 2
DYNAMIC	Dynamic storage link	***NO**, *YES	Optional, Positional 3
TYPE	Network server type	*Character value*, ***NWSD**	Optional
ACCESS	Access	***UPDATE**, *READ, *SHRUPD	Optional
DRVSEQNBR	Drive sequence number	1-64, ***CALC**, *QR	Optional
STGPTHNBR	Storage path number	1-4, ***DFTSTGPTH**, *MLTPTHGRP	Optional

Top

Network server storage space (NWSSTG)

Specifies the name of the network server storage space.

This is a required parameter.

Top

Network server description (NWSD)

Specifies the name of the network server description to which this link is to be added.

This is a required parameter.

Top

Dynamic storage link (DYNAMIC)

Specifies whether this storage is to be linked as dynamic storage using the next available location, or linked as static storage using the drive sequence number specified by the DRVSEQNBR parameter.

***NO** The storage space is linked in the standard linking method using the DRVSEQNBR parameter. The NWSD must be in a VARIED OFF state to perform this operation.

***YES** The storage space is linked using dynamic linking. It will use the next available sequence number.

Note: If the Network server description (NWSD) was created with *IXSVR or *ISCSI specified for the **Network server type (TYPE)** parameter, DYNAMIC parameter will be ignored and considered *YES when the server is installed with Windows 2000 or later, or Linux. If the NWSD was created with *GUEST specified for the **Network server type (TYPE)** parameter, DYNAMIC parameter will be ignored and considered *YES.

Top

Network server type (TYPE)

Describes the type of network server description to which this link is added.

Note: This parameter is present only for compatibility with previous releases. The value specified is not syntax checked and no verification is done to ensure that the network server description matches the specified TYPE value.

Top

Access (ACCESS)

Specifies the server's access method to the storage space.

***UPDATE**
The storage space is accessed in READ/WRITE mode. The storage space can be written to by the server.

***READ**
The storage space is accessed in READ-only mode. Write requests to the storage space are not allowed from this server. Multiple servers may be linked to this storage space.

***SHRUPD**
The storage space is accessed in shared update mode. The storage space can be written to by the server. Multiple servers may be linked to this storage space. Disks linked to servers in a Windows cluster must be linked as *SHRUPD.

Note: DYNAMIC(*YES) must be specified when ACCESS(*SHRUPD) is specified.

Top

Drive sequence number (DRVSEQNBR)

Specifies the order in which the network storage spaces are presented to the server. Each storage space must be given a unique sequence number.

***CALC**
The system will assign the lowest available sequence number.

***QR** The special value used for linking the quorum disk for a Windows cluster.

This value must be used when linking a *NTFSQR format disk as the quorum disk for a Windows cluster. The quorum disk must be linked with ACCESS(*SHRUPD) and it must be the first *SHRUPD link when *IXSVR is specified for the **Network server type (TYPE)** parameter, or must be the last (64th) when *ISCSI is specified for the **Network server type (TYPE)** parameter in the CRTNWSD command. Each server in a Windows cluster must be linked to one and only one quorum resource disk. The quorum must also be linked to each server with DYNAMIC(*YES) specified.

1-64 When network server description created with **Server connection *IXSVR** and **Server operating system *WIN32 or WIN64**:

1. When DYNAMIC(*YES) and ACCESS(*SHRUPD) is specified, a value between 1 and 15 must be specified.
2. When DYNAMIC(*YES) and ACCESS is *UPDATE, a value between 1 and 16 must be specified.
3. When DYNAMIC(*NO) is specified, a value between 3 and 18 must be specified.

When network server description created with **Server connection *ISCSI** and **Server operating system *WIN32 or WIN64**:

1. Only ACCESS(*UPDATE) is valid for DRVSEQNBR values 1 and 2.
2. When ACCESS *UPDATE or *SHRUPD is specified, a value between 3 and 63 must be specified.

When network server description created with **Server connection *IXSVR** or ***ISCSI** and **Server operating system *LINUX32**:

1. Only ACCESS(*UPDATE) is valid for DRVSEQNBR values 1 and 2.
2. When ACCESS *UPDATE or *SHRUPD is specified, a value between 3 and 64 must be specified.

When network server description created with **Server connection *ISCSI** and **Server operating system *LINUX32**:

1. Drive sequence number 64 is not valid.

When network server description created with **Server connection *GUEST**:

1. Any access mode can be specified for a value between 1 and 64.

Note: If a non-sequential number is specified for a Network server description (NWSD) created with **Server connection** specified *IXSVR or *ISCSI for the **Network server type (TYPE)** parameter, the number may not match the logical unit number on Windows or Linux and unexpected results may occur on drive assignments and applications.

Top

Storage path number (STGPTHNBR)

Specifies the storage path number in the network server description (NWSD) that is used by this storage space.

Note: This parameter is only valid when **Server connection** specified *ISCSI for the **Network server type (TYPE)** parameter in the corresponding **CRTNWSD** command.

See **Storage path (STGPTH)** parameter of the network server description on the **CRTNWSD** or **CHGNWSD** commands.

***DFTSTGPTH**
The storage space is linked to the relative storage path of the file server defined in the **Default storage path (DFTSTGPTH)** parameter of the network server description.

***MLTPTHGRP**
The storage space is linked using a multiple path group as defined on the **Multi-path group (MLTPTHGRP)** parameter of the network server description.

1-4 The storage space is linked using a specific file server storage path number. Use the **DSPNWSD** command with OPTION(*STGPTH) to see which storage paths are valid.

Top

Examples

Example 1:Linking Storage Spaces to a Network Server Description

```
ADDNWSSTGL   NWSSTG(PARTS)  NWSD(WNTSVR)  DRVSEQNBR(3)
ADDNWSSTGL   NWSSTG(DATA)  NWSD(WNTSVR)  DRVSEQNBR(*CALC)
```

These commands link storage space PARTS to the Windows network server description at drive sequence 3, and storage space DATA at the next available sequence number.

Example 2:Linking a Dynamic Storage Space

```
ADDNWSSTGL  NWSSTG(DATAUPD)  NWSD(W2KSVR)  DYNAMIC(*YES)
```

This command dynamically links the storage to the Windows network server description to the next available dynamic storage sequence.

Example 3: Linking Quorum and Shared Storage Spaces

```
ADDNWSSTGL   NWSSTG(QUORUM)  NWSD(DNETSVR1)  DYNAMIC(*YES)
             ACCESS(*SHRUPD)  DRVSEQNBR(*QR)
ADDNWSSTGL   NWSSTG(QUORUM)  NWSD(DNETSVR2)  DYNAMIC(*YES)
             ACCESS(*SHRUPD)  DRVSEQNBR(*QR)
ADDNWSSTGL   NWSSTG(DATASHR)  NWSD(DNETSVR1)  DYNAMIC(*YES)
             ACCESS(*SHRUPD)  DRVSEQNBR(*CALC)
ADDNWSSTGL   NWSSTG(DATASHR)  NWSD(DNETSVR2)  DYNAMIC(*YES)
             ACCESS(*SHRUPD)  DRVSEQNBR(*CALC)
```

These commands allow two Windows network server descriptions DNETSVR1 and DNETSVR2 to share storage space DATASHR with read/write access.

Example 4: Linking a Storage Path

```
ADDNWSSTGL  NWSSTG(DATAUPD)  NWSD(W2KSVR)
            STGPTHNBR(*DFTSTGPTH)
```

This command links the storage to the Windows network server description using the default storage path identified in the NWSD.

Top

Error messages

***ESCAPE Messages**

CPF26BA

Add network server storage link command failed.

Top

Add Optical Cartridge (ADDOPTCTG)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Optical Cartridge (ADDOPTCTG) command adds an optical disk cartridge and its volume(s) to an optical library or stand alone device. An optical cartridge can be either single or double sided. Each side represents an optical volume.

A 12-character date and time stamp is assigned by the system to an uninitialized volume when it is added to the optical device. This stamp is used as the volume identifier to track each volume until it is initialized. You can keep the system-generated volume identifier or you can specify a new volume identifier when the volume is initialized.

Restriction: The user must have *USE authority to use this command. The command is shipped with *EXCLUDE public authority.

Top

Parameters

Keyword	Description	Choices	Notes
MLB	Optical device	*Name*	Required, Positional 1
AUTL	Authorization list	*Name*, ***PRV**, QOPTSEC, *NONE	Optional, Positional 2
DIR	Rebuild directory index	***NO**, *YES	Optional
MEDLOC	Media location	***IOSTATION**, *MAGAZINE	Optional

Top

Optical device (MLB)

Specifies the name of the optical device where the optical media is to be added.

Top

Authorization list (AUTL)

Specifies the authorization list used to verify authority to the optical cartridge and its volume(s).

***PRV** The previous authorization list is used. If no previous authorization list was saved or if the previous authorization list does not exist, the default authorization list, QOPTSEC, is used.

QOPTSEC
The default authorization list for the optical volume(s), QOPTSEC, is used.

***NONE**
No security checking is performed for the optical volume(s).

authorization-list-name
Specify the name of the authorization list used.

Top

Rebuild directory index (DIR)

Specifies whether the optical directory index is built for the optical volume(s) being added. The directory index is required to run Work with Optical Directories (WRKOPTDIR) and Display Optical (DSPOPT) when displaying directories. If not built now, the index will be built later the first time one of these commands is issued.

***NO** The optical directory index will not be built for the optical volume(s) being added but instead will be built at a later time if needed. Using this option may result in better performance for the command than if DIR(*YES) were specified.

***YES** The optical directory index is created now for each volume.

Top

Media location (MEDLOC)

Specifies the location of the optical volume(s) to be added to an optical library..

***IOSTATION**
Add a single optical disk cartridge to an optical library from the input/output station.

***MAGAZINE**
Add multiple optical disk cartridges to an optical library from the bulk load magazine.

Note: Not all optical libraries are equipped with a bulk load magazine.

Top

Examples

Example 1: Specify an Authorization List

```
ADDOPTCTG   MLB(OPTMLB01)  AUTL(MYAUTH)
```

This command adds an optical cartridge and its volumes to the system in optical media library OPT01. The optical cartridge is secured with the authorization list MYAUTH.

Example 2: Specify a Media Location of *MAGAZINE

```
ADDOPTCTG   MLB(OPTMLB01)  AUTL(QOPTSEC) MEDLOC(*MAGAZINE)
```

This command adds all of the optical cartridges and volumes from the bulk load magazine to the system in optical media library OPTMLB01. All of the volumes added to optical media library OPTMLB01 will be secured by authorization list QOPTSEC.

Top

Error messages

***ESCAPE Messages**

OPT1245
Error processing directories for optical volume &1.

OPT1480
Add optical disk cartridge failed.

OPT1530
&1 does not represent a valid optical device.

OPT1555
Optical device &1 in use.

OPT1652
Device &1 is not an optical media library.

OPT1671
Feature not installed for optical device &1.

OPT1672
Error accessing bulk load magazine for optical device &1.

OPT1675
Bulk load magazine is empty for optical device &1.

OPT1677
Bulk load magazine is open for optical device &1.

OPT1790
Operation not allowed or conflicts with another request.

OPT1815
Internal program error occurred.

OPT1860
Request to optical device &1 failed.

OPT1861
No device description configured for resource &1.

OPT1862
No active device description for resource &1.

OPT1863
Optical libraries need to be reclaimed.

OPT2040
Error accessing backup control file.

OPT2301
Internal system object in use.

OPT2410
Authorization list &1 for volume &2 was not found.

OPT7740
User not authorized to object &2 in library &3 type &4.

Top

Add Optical Server (ADDOPTSVR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Optical Server (ADDOPTSVR) command enables access, using the hierarchical file system (HFS) APIs, to a remotely attached optical server. This command retrieves a list of optical volumes in each server and adds them to the optical index database allowing the volumes to be accessed using the HFS APIs. If a remote optical server is already enabled, you can use this command to refresh the volume list for that server.

Restrictions:

1. You must have *USE authority to use this command. It is shipped with *EXCLUDE public authority.
2. To use a remote optical server, the users must have the library that contains the communications side information for that destination in their library list.

Top

Parameters

Keyword	Description	Choices	Notes
CSI	Side information	Values (up to 16 repetitions): *Qualified object name*	Required, Positional 1
	Qualifier 1: Side information	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

Side information (CSI)

Specifies the communications side information (*CSI) object of the optical server to be added to the optical configuration. The communications side information object name is also referred to as the optical server name or the optical destination name. A maximum of 16 qualified names of servers can be specified.

Qualifier 1: Side information

communications-side-information-object-name
Specify the name of the communications side information object representing the optical server.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name
Specify the name of the library to be searched.

Top

Examples

```
ADDOPTSVR   CSI((QGPL/LAN01))
```

This command enables access to the optical server defined by the communications side information object name LAN01 in library QGPL.

Top

Error messages

***ESCAPE Messages**

OPT0125
Command &1 completed with errors, more information in job log.

OPT6712
Remote optical server volume list rebuild failed.

Top

Add OSPF Area (ADDOSPFARA)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The Add OSPF Area (ADDOSPFARA) command adds an OSPF area to the routing domain. An OSPF area is a collection of subnetted networks in which the OSPF interfaces are attached. Each OSPF area has an unique 32-bit identifier. The main purpose of an OSPF area is to reduce the routing traffic and be hidden to other areas, thereby reducing the size of the Link-State database.

This command can add an OSPF areas in either the IPv4 or IPv6 routing domain.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
AREA	Area identifier	*Character value*	Required, Positional 1
AUTHTYPE	Authentication type	***NONE**, *MD5, *PASSWORD	Optional
STUB	Stub area	***NO**, *YES	Optional
STUBCOST	Stub cost	1-65535, **1**	Optional
IMPORT	Import	***NO**, *YES	Optional
IPVERSION	IP version	***IPV4**, *IPV6	Optional

Top

Area identifier (AREA)

Specifies the OSPF area identifier to be configured.

This is a required parameter.

character-value
Specify the internet address for the OSPF area in the form *a.a.a.a* where *a* is a decimal number between 1 and 255.

Top

Authentication type (AUTHTYPE)

Specifies the default security schema to be used in this area. This parameter is only valid for OSPF IPv4.

***NONE**
No authentication is necessary to pass packets.

***MD5** MD5 cryptographic authentication is used.

***PASSWORD**
Simple password authentication is used.

Top

Stub area (STUB)

Specifies whether or not this area is a stub area. A stub area is a small OSPF area that will not import autonomous system external link advertisements.

***NO** This area is not a stub area.

***YES** The area is a stub area. The area will not receive any autonomous system external link advertisements, reducing the size of your database and decreasing memory usage for routers in the stub area. You cannot configure virtual links through a stub area. You cannot configure a router within the stub area as an autonomous system boundary router.

Top

Stub cost (STUBCOST)

Specifies the cost of the default summary_LSA that the system should advertise into the area, if the area has been configured as a stub area and the system is participating as a area border router.

1 The lowest stub cost value is used.

1-65535
Specify the stub default cost value.

Top

Import (IMPORT)

Specifies whether this stub area will import either a routing summary (IPv4) or prefixes from (IPv6) a neighbor area.

***NO** Specifies this area will not import routing information.

***YES** Specifies this area will import routing information.

Top

IP version (IPVERSION)

Specifies whether the area is part of the IPv4 or IPv6 routing domain.

***IPV4** This area will be part of an OSPF IPv4 routing domain.

***IPV6** This area will be part of an OSPF IPv6 routing domain.

Top

Examples

Example 1:Adding an OSPF IPv4 Area

```
ADDOSPFARA   AREA('1.1.1.1')
```

This commands adds an OSPF IPv4 area with an area identifier of 1.1.1.1.

Example 2:Adding an OSPF IPv6 Area

```
ADDOSPFARA   AREA('67.67.67.67')  IPVERSION(*IPV6)
```

This commands adds an OSPF IPv6 area with an area identifier of 67.67.67.67.

Top

Error messages

***ESCAPE Messages**

TCP8050
*IOSYSCFG authority required to use &1.

TCP6515
OSPF Area &1 has not been added to the configuration file. Fails with reason code &2.

TCP9999
Internal system error in program &1.

Top

Add OSPF Interface (ADDOSPFIFC)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The Add OSPF Interface (ADDOSPFIFC) command defines an OSPF interface that the OMPROUTED server will add into the OSPF routing protocol implementation and set the different parameters needed for each interface.

With this command both IPv4 and IPv6 OSPF Interfaces can be configured.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
IFC	Interface identifier	*Character value*, *PPPCNNPRF	Required, Positional 1
PPPCNNPRF	Connection profile	*Character value*, ***NONE**	Optional
AREA	Area identifier	*Character value*, ***BACKBONE**	Optional
RETMSITV	Retransmission interval	1-65535, **5**	Optional
TMSDLY	Transmission delay	1-65535, **1**	Optional
RTRPTY	Router priority	1-255, **1**	Optional
ITVHELLO	Interval for hello	1-255, **10**	Optional
DBEXCHTIMO	Database exchange timeout	2-65535, **40**	Optional
INACTTIMO	Inactive router timeout	2-65535, **40**	Optional
COST	Cost	1-65535, **1**	Optional
IFCUSG	Interface usage	*PRIMARY, ***BACKUP**	Optional
SUBNET	Subnet	***NO**, *YES	Optional
AUTHTYPE	Authentication type	***AREA**, *NONE, *MD5, *PASSWORD	Optional
AUTHVAL	Authentication values	*Element list*	Optional
	Element 1: Authentication key identifier	1-255	
	Element 2: Authentication key	*Character value*	
DMDCCT	Demand circuit	***NO**, *YES	Optional
SUPHELLO	Suppression of hello	***ALLOW**, *DISABLE, *REQUEST	Optional
PTPPOLLITV	Point-to-point poll interval	0-65535, **60**	Optional
NBCLNK	Nonbroadcast link	***DFT**, *NO, *YES	Optional
NBCPOLLITV	Nonbroadcast interval	1-65535, **120**	Optional
NGHRTRL	Neighbor router list	Values (up to 10 repetitions): *Element list*	Optional
	Element 1: Internet address	*Character value*	
	Element 2: Eligible	***NO**, *YES	

Keyword	Description	Choices	Notes
INSTANCE	Instance	*Integer*, **0**	Optional

Top

Interface identifier (IFC)

Specifies the internet address of a configured logical interface in the system. The interface identifier can be an IPv4 or IPv6 address.

***PPPCNNPRF**
The point-to-point connection profile specified for the **Connection profile (PPPCNNPRF)** parameter is used to send and receive OSPF routing traffic over a point-to- point link.

character-value
Specify the internet address of the OSPF interface to add. An IPv4 internet address is specified in the form *nnn.nnn.nnn.nnn* where *nnn* is a decimal number between 0 and 255. An IPv6 internet address is specified in the form *n:n:n:n:n:n:n:n* where *n* is a hexadecimal number in the range from 0 through X'ffff'. The value "::" indicates that one or more groups of 16 bits are zero.

The interface identifier can also be an alias name of an logical interface.

Top

Connection profile (PPPCNNPRF)

Specifies the profile created for point-to-point connection to a host at the remote end. This profile will be used later when the point-to-point link is being active. This parameter is only valid for OSPF IPv4.

***NONE.**
Specifies that this interface will not work over a point-to-point link.

character-value
Specify the connection profile to be used. This profile must have been configured previuosly using Remote Access Services.

Top

Area identifier (AREA)

Specifies the OSPF area to which this interface attaches. You can specify the backbone area (internet address 0.0.0.0) or any area previously defined by running the Add OSPF Area (ADDOSPFARA) command.

***BACKBONE**
The backbone area is used (internet address 0.0.0.0).

character-value
Specify the internet address for the OSPF area in the form *a.a.a.a* where *a* is a decimal number between 1 and 255.

Top

Retransmission interval (RETMSITV)

Specifies the frequency of retransmitting link-state update packets, link-state request packets, and database description packets.

Note: If this parameter is set too low, needless retransmissions will occur that could affect performance and interfere with neighbor adjacency establishment. It should be set to a higher value for a slower machine.

5 Retransmit packets every 5 seconds.

1-65535
Specify the number of seconds to use for the retransmission interval value.

Top

Transmission delay (TMSDLY)

Specifies the estimated number of seconds that it takes to transmit link-state information over the interface.

1 This interface is on a low-latency network.

1-65535
Specify the number of seconds to use for the transmission delay value.

Top

Router priority (RTRPTY)

Specifies the value for a broadcast and non-broadcast multi-access networks in order to elect a designated router. The router with the highest priority value will be elected.

1 This system should not normally be used as a Designated Router.

1-255 Specify the router priority value to use.

Top

Interval for hello (ITVHELLO)

Specifies the number of seconds between OSPF hello packets to be sent out on this interface. This value must be the same for all routers attached to a common network.

10 Send hello packets every 10 seconds.

1-65535
Specify the number of seconds to use for the hello interval value.

Top

Database exchange timeout (DBEXCHTIMO)

Specifies the length of time that the database exchange process cannot exceed. If the interval elapses, the procedure will be restarted. This value must be larger than the interval for hello.

40 Database exchanges should be completed within 40 seconds.

1-65535
Specify the number of seconds to use for the database exchange process timeout value.

Top

Inactive router timeout (INACTTIMO)

Specifies the length of time after not having received OSPF hello packets that an OSPF neighbor is declared to be down.

40 The router will be considered down after not having received hello packets in 40 seconds.

1-65535
Specify the number of seconds to use for the inactive router timeout value.

Top

Cost (COST)

Specifies the cost of sending a data packet on the interface, expressed in the link state metric. This is advertised as the link cost for this interface in the router-LSA. The cost of an interface must be greater than zero.

1 This interface has the lowest possible cost.

1-65535
Specify the cost value to be used.

Top

Interface usage (IFCUSG)

Specifies whether this interface is designated as primary or backup by OSPF. When there is more than one local interface defined on the same IPv4 subnet, IPv6 subnet, or link, OSPF sends all routing traffic on the designated primary interface.

***BACKUP**
This OSPF interface is defined as a backup interface. If the primary interface is not specified, OSPF chooses one of the backup interfaces to carry out routing traffic.

***PRIMARY**
This OSPF interface is defined as the primary interface. OSPF sends all routing traffic over this interface.

Top

Subnet (SUBNET)

Specifies, for Point-to-Point links, if this interfaces can have a connection to a network/subnet or host. This parameter only applies to IPv4.

***NO** This interface cannot have a connection to a network/subnet or host. Implements option 1 of section 12.4.1.1 of RFC 2328. It assumes that neighbor router's IP address is known.

***YES** This interface can have a connection to a network/subnet or host. Implements option 2 of section 12.4.1.1 of RFC 2328 if a subnet has been assigned to the point-point link.

Top

Authentication type (AUTHTYPE)

Specifies the security schema used on the network where this interface is attached. If this parameter is not specified the security schema will be taken from the area's security schema. This parameter is ignored for OSPF IPv6 interfaces.

***AREA**
Use the security schema defined in the OSPF area to which this interface is attached.

***NONE**
No authentication is necessary to pass packets.

***MD5** MD5 cryptographic authentication is used.

***PASSWORD**
Simple password authentication is used.

Top

Authentication values (AUTHVAL)

Specifies the combination of parameters needed for cryptographic authentication.

Element 1: Authentication key identifier

0-255 Specify the constant value to use as the key identifier. This value is only meaningful when the authentication type *MD5 is specified and is used with other routers that require identification of an authentication type.

Element 2: Authentication key

character-value
Specify the authentication key for this interface. This value must be the same in all routers that are attached to a common network. This value depends on the authentication type used for this interface. This parameter is ignored for OSPF IPv6.

For *NONE authentication type, this parameter element is not required and is ignored if specified.

For authentication type *PASSWORD, this parameter element must specify a password for OSPF routers that are attached to this subnet. Valid values are any character string up to 8 characters in length coded within single quotation marks, or any hexadecimal string up to 8 bytes (16 hexadecimal characters) long beginning with X followed by the hexadecimal value enclosed in single quotation marks.

For authentication type *MD5, this parameter must specify the 16-byte MD5 authentication key for OSPF routers attached to this subnet. The standard method is with a 16-byte hexadecimal string beginning with X followed by the hexadecimal value enclosed in single quotation marks.

Top

Demand circuit (DMDCCT)

Specifies whether link state advertisements (LSAs) are periodically refreshed or not over this interface.

Only LSAs with real changes will be advertised. This parameter is ignored by IPv6 interfaces.

***NO** Link state advertisements are periodically refreshed.

***YES** Link state advertisements are not periodically refreshed.

Top

Suppression of hello (SUPHELLO)

Specifies whether to allow suppression of hello packets. This support is only used for point-to-point-interface and point-to-multipoint interfaces types that are on demand circuits. This parameter only applies to IPv4.

***ALLOW**
If both sides specify this option, hello suppression is disabled.

***DISABLE**
If either or both sides specify this option, hello suppression is disabled.

***REQUEST**
If both sides specify this option, hello suppression is enabled.

Top

Point-to-point poll interval (PTPPOLLITV)

Specifies the interval time that OMPROUTED will try to re-establish a relationship with a neighboring router that has failed but the interface is still active. This parameter is used only if the interface is Point-to-Point type and suppression for hello has been enabled. This parameter is ignored for OSPF IPv6 interfaces.

60 Attempt re-establishing a relationship every 60 seconds.

0-65535
Specify the number of seconds to use for the point-to-point poll interval value.

Top

Nonbroadcast link (NBCLNK)

Specifies if the router is connected to a non-broadcast or multi-access (NBMA) network.

***DFT** Determined by the interface type.

***NO** Router is not connected to a NBMA network. Neighbors are discovered by sending a broadcast packet on the network. If this option is specified for an interface on a network which does not support broadcast, neighbor discovery will fail.

***YES** The interface is connected to a non-broadcast network. This option can help to discover its neighbors.

Top

Nonbroadcast interval (NBCPOLLITV)

Specifies the time interval for hello packets to be sent from this interface to neighbors that are inactive. It is only valid if *YES is specified for the **Nonbroadcast link (NBCLNK)** parameter.

120 Attempt to send hello packets every 120 seconds.

1-65535
Specify the number of seconds to be used for the non-broadcast poll interval value.

Top

Neighbor router list (NGHRTRL)

Specifies the list of routers eligible to be treated as neighbors for this system over this interface.

Up to 10 neighboring routers can be defined.

Note: It is not necessary or recommended to define neighbors on nonbroadcast or multicast-capable media. If you do define neighbors on these media, OMPROUTED will be able to communicate OSPF information only with those neighbors that are defined (it will not form adjacencies with any additional neighbors).

Element 1: Internet address

Specifies the neighboring router's internet address.

character-value
Specify the internet address of the router to be eligible to be a neighbor router. An IPv4 internet address is specified in the form *nnn.nnn.nnn.nnn* where *nnn* is a decimal number between 0 and 255. An IPv6 internet address is specified in the form *n:n:n:n:n:n:n:n* where *n* is a hexadecimal number in the range from 0 through X'ffff'. The value "::" indicates that one or more groups of 16 bits are zero.

Element 2: Authentication key identifier

Specifies whether or not the neighbor router may be eligible to be a designated router.

***NO** This router cannot be a designated router.

***YES** This router can be elected as a designated router.

Top

Instance (INSTANCE)

Specifies the IPv6 protocol instance number for this interface. This value should be the same as the instance value of other IPv6 OSPF hosts or routers that OMPROUTED communicates with on the link. This value will be set in all the outgoing IPv6 OSPF packets. All the incoming OSPF packets whose instance value does not match with the instance value set for this interface will be ignored. This characteristic permits multiples instance of IPv6 OSPF to be run on this link. OMPROUTED supports only one instance per link, however if this parameter is configured in order to interact with router that can supports multiples IPv6 OSPF instances on a link.

0 Recommended default value for instance.

1-255 Specify the instance value to be used.

Top

Examples

Example 1:Adding an OSPF IPv4 Interface

```
ADDOSPFIFC   IFC('9.67.107.7')
```

This command adds an OSPF IPv4 interface. The referenced logical interface must exist on the system.

Example 2:Adding an OSPF IPv6 Interface

```
ADDOSPFIFC   IFC('1000::5678:9abc:def')
```

This command adds an OSPF IPv6 interface. The referenced logical interface must exist on the system.

Example 3:Adding an Interface that Will Work Over a Point-to-Point Link

```
ADDOSPFIFC   IFC(*PPPCNNPRF)  PPPCNNPRF(PROFILE1)
```

This command adds an OSPF IPv4 interface using a connection profile. The connection profile PROFILE1 must have already been configured using Remote Access Services.

Top

Error messages

***ESCAPE Messages**

TCP8050
*IOSYSCFG authority required to use &1.

TCP6525
Internet address &1 not valid

TCP1902
Internet address &1 not valid.

TCP1908
Internet address &1 not valid.

TCP6524
OSPF interface &1 has not been added successfully in configuration file. Fails with reason code &2.

TCP9999
Internal system error in program &1.

Top

Add OSPF Virtual Link (ADDOSPFLNK)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add OSPF Virtual Link (ADDOSPFLNK) command configures a virtual link between two area border routers. To maintain backbone connectivity you must have all of your backbone routers interconnected either by permanent or virtual links. Virtual links are considered to be separate router interfaces connecting to the backbone area. Therefore, you are asked to specify many of the interface parameters when configuring a virtual link.

Virtual links can be configured between any two backbone routers that have an interface to a common non-backbone, non-stub area. Virtual links are used to maintain backbone connectivity and must be configured at both endpoints.

IPv4 and IPv6 OSPF virtual links can be added. The **Neighbor router (LNKTMSARA)** parameter value will determine whether the virtual link is an IPv4 or IPv6 OSPF virtual link.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
NGHRTR	Neighbor router	*Character value*	Required, Positional 1
LNKTMSARA	Link transmission area	*Character value*	Required, Positional 2
RETMSITV	Retransmission interval	1-65535, **10**	Optional
TMSDLY	Transmission delay	1-65535, **5**	Optional
ITVHELLO	Interval for hello	1-255, **30**	Optional
DBEXCHTIMO	Database exchange timeout	2-65535, **180**	Optional
INACTTIMO	Inactive router timeout	2-65535, **40**	Optional
AUTHTYPE	Authentication type	***BACKBONE**, *NONE, *MD5, *PASSWORD	Optional
AUTHVAL	Authentication values	*Element list*	Optional
	Element 1: Authentication key identifier	1-255	
	Element 2: Authentication key	*Character value*	

Top

Neighbor router (NGHRTR)

Specifies the router identifier of the neighbor (other endpoint).

This is a required parameter.

character-value
Specify the internet address of the router in the form *a.a.a.a* where *a* is a decimal number between 1 and 255.

Top

Link transmission area (LNKTMSARA)

Specifies a non-backbone, non-stub area through which the virtual link is configured. Virtual links can be configured between any two area border routers that have an interface to a common non-backbone and non-stub area. Virtual links must be configured in each of the link's two endpoints.

This is a required parameter.

character-value
Specify the internet address of the area in the form *a.a.a.a* where *a* is a decimal number between 1 and 255.

Top

Retransmission interval (RETMSITV)

Specifies the frequency (in seconds) of retransmitting link-state update packets, link-state request packets, and database description packets.

Note: If this parameter is set too low, needless retransmissions will occur that could affect performance and interfere with neighbor adjacency establishment. It should be set to a higher value for a slower machine.

10 Retransmit packets every 10 seconds.

1-65535
Specify the number of seconds to use for the retransmission interval value.

Top

Transmission delay (TMSDLY)

Specifies the estimated number of seconds that it takes to transmit link-state information over the virtual link. Each link-state advertisement has a finite lifetime of 1 hour. As each link-state advertisement is sent out from this virtual link, it will be aged by this configured transmission delay.

5 This virtual link is on a low-latency network.

1-65535
Specify the number of seconds to use for the transmission delay value.

Top

Interval for hello (ITVHELLO)

Specifies the number of seconds between OSPF hello packets being sent out from this virtual link. The interval for hello should be set higher than the same value used on the intervening OSPF interfaces.

30 Send hello packets every 30 seconds.

1-65535
Specify the number of seconds to use for the hello interval value.

Top

Database exchange timeout (DBEXCHTIMO)

Specifies the length of time that the database exchange process cannot exceed. If the interval elapses, the procedure will be restarted. This value must be larger than the interval for hello.

180 Database exchanges should be completed within 180 seconds.

2-65535
Specify the number of seconds to use for the database exchange timeout value.

Top

Inactive router timeout (INACTTIMO)

Specifies the interval, in seconds, after not having received an OSPF hello, that the neighbor is declared to be down. This value must be larger than the interval for hello. The inactive router timeout should be set higher than the same value used on the intervening OSPF interfaces.

40 The router will be considered down after not having received hello packets in 40 seconds.

2-65535
Specify the number of seconds to use for the inactive router timeout value.

Top

Authentication type (AUTHTYPE)

Specifies the security scheme to be used over the virtual link. This parameter is ignored for OSPF IPv6 virtual links.

***BACKBONE**
Use the security schema defined in the OSPF backbone area.

***NONE**
No authentication is needed to pass OSPF packets.

***MD5** MD5 cryptographic authentication is used.

***PASSWORD**
Simple password authentication is used.

Top

Authentication values (AUTHVAL)

Specifies the combination of parameters needed for cryptographic authentication.

Element 1: Authentication key identifier

0-255 Specify the constant value to use as the key identifier. This value is only meaningful when the authentication type *MD5 is specified and is used with other routers that require identification of an authentication type.

Element 2: Authentication key

character-value

Specify the authentication key for this virtual link. This value must be the same in all routers that are attached to a common network. This value depends on the authentication type used for this interface. This parameter is ignored for OSPF IPv6.

For *NONE authentication type, this parameter element is not required and is ignored if specified.

For authentication type *PASSWORD, this parameter element must specify a password for OSPF routers that are attached to this subnet. Valid values are any character string up to 8 characters in length coded within single quotation marks, or any hexadecimal string up to 8 bytes (16 hexadecimal characters) long beginning with X followed by the hexadecimal value enclosed in single quotation marks.

For authentication type *MD5, this parameter must specify the 16-byte MD5 authentication key for OSPF routers attached to this subnet. The standard method is with a 16-byte hexadecimal string beginning with X followed by the hexadecimal value enclosed in single quotation marks.

Top

Examples

Example 1:Adding an OSPF IPv4 Virtual Link

```
ADDOSPFLNK   NGHRTR('2.2.2.2')  LNKTMSARA('3.3.3.3')
```

This command adds a virtual link for neighbor router 2.2.2.2 with a link transmission area identifier of 3.3.3.3. To add a virtual link, you need the router identifier over which the link will be established and the area identifier of the link transmission area. The system must be an area border router. The link transmission area should have been configured for IPv4 previously.

Example 2:Adding an OSPF IPv6 Virtual Link

```
ADDOSPFLNK   NGHRTR('62.62.62.62')  LNKTMSARA('63.63.63.63')
```

This command adds a virtual link for neighbor router 62.62.62.62 with a link transmission area identifier of 63.63.63.63. To add a virtual link, you need the router identifier over which the link will be established and the area identifier of the link transmission area. The system must be an area border router. The link transmission area should have been configured for IPv6 previously.

Top

Error messages

***ESCAPE Messages**

TCP8050
*IOSYSCFG authority required to use &1.

TCP6531
OSPF virtual link &1 has not added from configuration file. Fails with reason code &2.

TCP9999
Internal system error in program &1.

Top

Add OSPF Range (ADDOSPFRNG)

Where allowed to run: All environments (*ALL)
Threadsafe: Yes

Parameters
Examples
Error messages

The Add OSPF Range (ADDOSPFRNG) command adds a range to an OSPF areas. OSPF areas are defined in terms of address ranges. External to the area, a single route is advertised for each address range. Ranges can be defined to control which routes are advertised external to an area. When OSPF is configured not to advertise the range, no inter-area routes are advertised for routes that fall within the range. Ranges cannot be used for areas that serves as transit areas for virtual links.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
AREA	Area identifier	*Character value*	Required, Positional 1
IPADRRNG	IP address range	*Character value*	Required, Positional 2
SUBNETMASK	Subnet mask	*Character value*, *HOST	Optional
PFXLEN	Prefix length	1-128, **64**	Optional
ADVERTISE	Advertise	***YES**, *NO	Optional

Top

Area identifier (AREA)

Specifies the area identifier for which the range is to be added. The area must have been defined by running the Add OSPF Area (ADDOSPFARA) command.

This is a required parameter.

character-value
Specify the internet address for the OSPF area in the form *a.a.a.a* where *a* is a decimal number between 1 and 255.

Top

IP address range (IPADRRNG)

Specifies the common subnet portion of internet addresses in this range.

This is a required parameter.

character-value
An IPv4 internet address is specified in the form *nnn.nnn.nnn.nnn* where *nnn* is a decimal number between 0 and 255.

An IPv6 internet address is specified in the form *n:n:n:n:n:n:n:n* where *n* is a hexadecimal number in the range from 0 through X'ffff'. The value "::" indicates that one or more groups of 16 bits are zero.

Top

Subnet mask (SUBNETMASK)

Specifies the subnet mask with respect to the network range specified for the **IP address range (IPADRRNG)** parameter. Subnet mask is ignored for IPv6 ranges.

***HOST**
The subnet mask value used should be 255.255.255.255.

character-value
Specify the mask for the network subnet field and host address field of the internet address that defines a subnetwork. The subnet mask is in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. The subnet mask must mask off all bits of the network class's network ID portion of the internet address.

For example, 255.255.255.0 could define a subnet mask for an interface with a Class B internet address. In this example, the first two octets must be 1 bits because these octets define the network ID portion of the Class B internet address. The third octet of this subnet mask defines the actual subnet mask ID portion of the interface's internet address. It is also all 1 bits. This leaves the fourth octet to define the host ID portion of the interface's internet address.

Top

Prefix length (PFXLEN)

Specifies how much of the leftmost portion of an IPv6 address is the subnet prefix. This parameter only is valid is the internet address range is an IPv6 address.

64 The default is 64 since most of the IPv6 address have a 64-bit interface ID.

1-128 Specify the number of bits of the IPv6 internet address to use as the subnet prefix.

Top

Advertise (ADVERTISE)

Specifies whether this range will be advertised to other areas or not.

***YES** This range will be advertised to other areas.

***NO** This range will not be advertised to other areas.

Top

Examples

Example 1:Adding a Range to an OSPF IPv4 Area

```
ADDOSPFRNG   AREA('1.1.1.1')  IPADRRNG('128.185.0.0')
             SUBNETMASK('255.255.0.0')
```

The command adds a range to an OSPF IPv4 area. If the OSPF area were to consist of all subnets of class B network 128.185.0.0, it would be defined as consisting of a single address range. The address range would be specified as an address of 128.185.0.0 with a mask of 255.255.0.0. Outside of the area, the entire subnetted network would be advertise as a single route to network 128.185.0.0.

Example 2:Adding a Range to an OSPF IPv6 Area

```
ADDOSPFRNG   AREA('67.67.67.67')  IPADRRNG('2001:0db8:1:2::')
```

The command adds a range to an OSPF IPv4 area. If an IPv6 OSPF area were to consist of the prefix 2001:0db8:1:2::/64, all addresses falling within that prefix would be defined as consisting of a single address range. The address range would be specified as an address of 2001:0db8:1:2:: together with a prefix length of 64. Outside of the area, all addresses that fall within that prefix would be advertised as a single route to prefix 2001:0db8:1:2::/64.

Top

Error messages

***ESCAPE Messages**

TCP8050
*IOSYSCFG authority required to use &1.

TCP6525
Internet address &1 not valid

TCP1902
Internet address &1 not valid.

TCP1908
Internet address &1 not valid.

TCP652B
OSPF range &1 has not been added in configuration file. Fails with reason code &2.

TCP9999
Internal system error in program &1.

Top

Add Protocol Table Entry (ADDPCLTBLE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Protocol Table Entry (ADDPCLTBLE) command is used to add a protocol entry to the protocol table. You can use the protocol table to manage a list of protocols used in the Internet. The **Internet** is a collection of networks functioning as a single, cooperative, and virtual network using Transmission Control Protocol/Internet Protocol (TCP/IP) to support peer-to-peer connectivity.

The protocol table is shipped with a list of some valid protocols. Current protocol values are available to the Internet community in the assigned numbers **RFC** (Request for Comments) document, a formal specification of proposals and standards for a portion of TCP/IP.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
PROTOCOL	Protocol	*Character value*	Required, Positional 1
PCLNBR	Protocol number	*Integer*	Required, Positional 2
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional
ALIAS	Aliases	Single values: ***NONE** Other values (up to 4 repetitions): *Character value*, ***NONE**	Optional

Top

Protocol (PROTOCOL)

Specifies the protocol to be added to the table. A protocol can be added to the table only once.

character-value
Specify the protocol name.

Top

Protocol number (PCLNBR)

Specifies the number that represents the protocol.

integer
Specify the protocol number.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the protocol entry.

***BLANK**
Text is not specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Aliases (ALIAS)

Specifies the alternate name for the protocol. You can specify a maximum of 4 aliases. No checking is done to ensure that an alias is unique.

Single values

***NONE**
The protocol has no alternate name.

Other values (up to 4 repetitions)

character-value
Specify an alternate protocol name.

Top

Examples

```
ADDPCLTBLE   PROTOCOL(TCP)  PCLNBR(6)
```

This command adds an entry for the TCP protocol to the protocol table. The protocol number for the TCP entry is 6.

Top

Error messages

***ESCAPE Messages**

TCP290B
Protocol entry already exists in table. Entry was not added.

TCP2915
Protocol entry contains characters that are not valid. Entry was not added.

TCP8050
*IOSYSCFG authority required to use &1.

Top

Add PEX Definition (ADDPEXDFN)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The Add Performance Explorer Definition (ADDPEXDFN) command adds a new Performance Explorer definition to the system. Each definition is stored as a member in the QAPEXDFN file in library QUSRSYS. A Performance Explorer definition identifies the performance data that is to be collected during a Performance Explorer session. A session can be started using the STRPEX (Start Performance Explorer) command. When starting a new session, a Performance Explorer definition name must be provided.

Additional information about the Performance Explorer tool can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. You must have *EXECUTE authority to the library of each program specified on the PGM parameter.
3. To use this command you must have *SERVICE special authority, or be authorized to the Service Trace function of i5/OS through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
4. The following user profiles have private authorities to use the command:
   - QPGMR
   - QSRV
5. Two threads within the same job will not be allowed to run ADDPEXDFN at the same time. The thread that issued ADDPEXDFN first will run the command to completion while the second ADDPEXDFN waits.

Top

Parameters

Keyword	Description	Choices	Notes
DFN	Definition	*Name*	Required, Positional 1
TYPE	Type	***STATS**, *TRACE, *PROFILE	Optional
PRFTYPE	Profile type	***PGM**, *JOB	Optional
JOB	Jobs	Single values: *, *NONE Other values (up to 8 repetitions): *Element list*	Optional
	Element 1: Job name	*Qualified job name*	
	Qualifier 1: Job name	*Generic name, name*, ***ALL**	
	Qualifier 2: User	*Generic name, name*, ***ALL**	
	Qualifier 3: Number	000001-999999, ***ALL**	
	Element 2: Thread identifier	Single values: ***ALL**, *SELECT Other values (up to 20 repetitions): 00000001-FFFFFFFF, *INITIAL	
	Element 3: Subsystem	*Generic name, name*, ***ALLSBS**	

Keyword	Description	Choices	Notes
TASK	Task name	Single values: ***NONE**, *ALL Other values (up to 8 repetitions): *Generic name, name*	Optional
PGM	Program to monitor	Values (up to 16 repetitions): *Element list*	Optional
	Element 1: Program	*Qualified object name*	
	Qualifier 1: Program	*Name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**	
	Element 2: Module	*Name*, ***ALL**	
	Element 3: Procedure	*Character value*, ***ALL**	
	Element 4: Type	***PGM**, *SRVPGM	
	Element 5: Pane size	**4**, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096	
DTAORG	Data organization	***FLAT**, *HIER	Optional
MAXSTG	Maximum storage to use	1024-250000000, **10000**	Optional
TRCFULL	Trace full	***STOPTRC**, *WRAP	Optional
MRGJOB	Merge job data	***YES**, *NO	Optional
ADDTHDOPT	Add threads/tasks option	***ALL**, *NEW, *CURRENT	Optional
LSTALLJOB	List all jobs/tasks	***NO**, *YES	Optional
INTERVAL	Sampling interval	*Element list*	Optional
	Element 1: Milliseconds	0.1-200.0, ***NONE**	
	Element 2: Randomize	***FIXED**, *VARY	
PGMBKTEVT	Program bracketing events	Single values: ***DFT** Other values (up to 3 repetitions): *MISTREND, *MIENTRYEXIT, *JVA, *PRC	Optional
TRCTYPE	Trace type	Single values: *SLTEVT Other values (up to 12 repetitions): ***CALLRTN**, *BASIC, *DSKIO1, *DSKIO2, *DSKSVR, *DSKSTG, *VRTADR, *PGMACT, *FILEOPEN, *PRFDTA, *TASKSWT, *HEAP	Optional
SLTEVT	Specific events	***NO**, *YES	Optional
MCHINST	Machine instructions	Single values: ***ALL**, *NONE Other values (up to 50 repetitions): *Character value*	Optional
PGMEVT	Program events	Single values: ***NONE** Other values (up to 10 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *MIENTRY, *MIEXIT, *MISTR, *MIEND, *JVAENTRY, *JVAEXIT, *PRCENTRY, *PRCEXIT, *PASEPRCENTRY, *PASEPRCEXIT	
	Element 2: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
BASEVT	Base events	Single values: ***NONE** Other values (up to 27 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *PRCCRT, *PRCDLT, *TASKCRT, *TASKDLT, *TASKSWTIN, *TASKSWTOUT, *TASKSWTOUTQ, *TASKSWTOUTINT, *TASKAVAIL, *PMCO, *PGMDBG, *ACTGRPACTPGM, *ACTGRPCRT, *ACTGRPDLT, *EXCP, *MIEXCP, *DCRINTSTR, *DCRINTEND, *CPUVRYON, *CPUVRYOFF, *CPUSWTIN, *CPUSWTOUTY, *CPUSWTOUTP, *CPUAVAIL, *CPUSWT, *SETPRF, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	

Keyword	Description	Choices	Notes
STGEVT	Storage events	Single values: ***NONE** Other values (up to 13 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *CRTSEG, *DLTSEG, *EXDSEG, *FNDSEGSIZ, *TRUNCSEG, *SYSHEAP, *RESHEAP, *LCLHEAP, *USRHEAP, *ACTGRPHEAP, *HDLHEAP, *CLEHEAP, *STGPTC, *TSADRFALSETRAP, *SPCADRFALSETRAP, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
DSKEVT	Disk events	Single values: ***NONE** Other values (up to 14 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *ALLSTR, *READSTR, *READEND, *WRTSTR, *WRTEND, *PGREADSTR, *PGREADEND, *PGWRTSTR, *PGWRTEND, *RMTWRTSTR, *RMTWRTEND, *RMTPGWRTSTR, *RMTPGWRTEND, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
FAULTEVT	Fault events	Single values: ***NONE** Other values (up to 4 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *STR, *ENDOK, *ENDERR, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
JOBEVT	Job events	Single values: ***NONE** Other values (up to 10 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *LWSTR, *LWEND, *INELIGIBLE, *ACTIVE, *INTERRUPT, *MPLPOOLCHG, *TOBCHMPLPOOL, *TSLEND, *MPLLEAVE, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
LCKEVT	Lock events	Single values: ***NONE** Other values (up to 5 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *LWEND, *OBJLOCK, *DBLOCK, *SPCLOCK, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
SAREVT	SAR events	Single values: ***NONE** Other values (up to 18 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *ALLSTR, *CLR, *CLRPIN, *EXCHCLR, *READ, *READASYNC, *READASYNCCALL, *READPIN, *EXCHREAD, *EXCHREADASYNC, *WRT, *WRTASYNC, *WRTRMV, *WRTPGOUT, *RMV, *UNPINRMV, *ENDOK, *ENDERR, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
EXPCCHEVT	Expert cache events	Single values: ***NONE** Other values (up to 3 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *TUNE, *TUNEDB, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	

Keyword	Description	Choices	Notes
DSKSVREVT	Disk server events	Single values: ***NONE** Other values (up to 11 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *CTLEND, *RQSACP, *RQSRJT, *SVRTASKSTR, *SVRTASKEND, *RQSRCV, *RQSENDOK, *RQSENDERR, *TAPSNDCMD, *TAPDONE, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
OSEVT	Operating System events	Single values: ***NONE** Other values (up to 54 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *DBIO, *DBOPEN, *DBSVRCNN, *DBSVRREQ, *DIRSRV, *DTAARA, *DTAQ, *HOSTSVRCNN, *IFSCOMP, *IFSIO, *IFSOPEN, *MIEV8, *MGTC, *TRCCOMP, *TRCDTA, *USRTNS, *SAVRST, *ADDTHD1, *ARMTRC, *MIEV25, *MIEV26, *MIEV27, *MIEV28, *MIEV29, *MIEV30, *MIEV35, *MIEV36, *MIEV37, *MIEV38, *MIEV39, *MIEV40, *MIEV41, *MIEV42, *MIEV43, *MIEV44, *MIEV45, *MIEV46, *MIEV47, *MIEV48, *MIEV49, *MIEV50, *MIEV51, *MIEV52, *MIEV53, *MIEV54, *MIEV55, *MIEV56, *MIEV57, *MIEV58, *MIEV59, *MIEV60, *MIEV61, *MIEV62	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
JVAEVT	Java events	Single values: ***NONE** Other values (up to 20 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *OBJCRT, *GBGCOL, *GBGCOLSWEEP, *THDCRT, *THDDLT, *THDWAIT, *THDNFY, *THDNFYALL, *CLSLOAD, *CLSUNLOAD, *LIBOPR, *TFMSTR, *LCKSTR, *UNLCK, *JVAEXCP, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
CMNEVT	Communications events	Single values: ***NONE** Other values (up to 6 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *SOCKETSAPI, *SOCKETS, *IP, *TCP, *TCPIN, *TCPOUT, *UDP, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
APPEVT	Application events	Single values: ***NONE** Other values (up to 7 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *DOMTRCDTA, *WAS, *CONNECT, *APPEVT1, *APPEVT2, *APPEVT3, *APPEVT4	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
PASEEVT	PASE events	Single values: ***NONE** Other values (up to 22 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *SYSCALLSTR, *SYSCALLEND, *FORKPARENTSTR, *FORKCHILDSTR, *FORKEND, *PRCINITSTR, *PRCINITEND, *EXECSTR, *EXECEND, *EXIT, *THDINITSTR, *THDINITEND, *THDEND, *LOADSTR, *LOADEND, *UNLOADSTR, *UNLOADEND, *SIGPRCSND, *SIGTHDSND, *SIGRCV, *TRCHOOK, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	

Keyword	Description	Choices	Notes
FILSVREVT	File server events	Single values: ***NONE** Other values (up to 9 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *IFSBSF, *IFSDIR, *IFSLFS, *LIC, *NETSVR, *NFS, *OS, *VNODMGMT, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
SYNCEVT	Synchronization events	Single values: ***NONE** Other values (up to 13 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *CONDSET, *CONDWAIT, *HDLMTXRLS, *HDLMTXWAIT, *PTRMTXLOCK, *PTRMTXUNLOCK, *PTRSEMPOST, *PTRSEMWAIT, *NAMSEMPOST, *NAMSEMWAIT, *TKNLOCK, *TKNUNLOCK, *MTXCLEANUP, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
JRNEVT	Journal events	Single values: ***NONE** Other values (up to 10 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *STRCOMMIT, *ENDCOMMIT, *STROBJFORCE, *ENDOBJFORCE, *STRROLLBACK, *ENDROLLBACK, *STRCYCLE, *STREVAL, *ENDEVAL, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
DBEVT	Database events	Single values: ***NONE** Other values (up to 9 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
RSCAFNEVT	Resources affinity events	Single values: ***NONE** Other values (up to 9 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *CHGTASK, *CHGGRP, *CHGCFG, *BALCFG, *CHKBAL, *VFYBAL, *ANZBAL, *CHGBALSTT, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
VRTIOEVT	Virtual I/O events	Single values: ***NONE** Other values (up to 12 repetitions): *Element list*	Optional
	Element 1: Event identifier	*ALL, *SCSICLIENT, *SCSISERVER, *ISCSI, *SCSICMD, *SCSITSKMGMT, *DISKSTR, *DISKEND, *OPTSTR, *OPTEND, *TAPSTR, *TAPEND, *ETHADPT, *SERVICE	
	Element 2: Event counter	***NONE**, 1, 2, 3, 4	
	Element 3: Event format	***FORMAT1**, *FORMAT2, *FORMAT3, *FORMAT4	
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Top

Definition (DFN)

Specifies the name of the Performance Explorer definition being added. If the specified definition already exists in the QAPEXDFN file in library QUSRSYS, an error condition will occur. The user can either change the definition name or remove the existing definition using the Remove Performance Explorer Definition (RMVPEXDFN) command, and try this command again.

This is a required parameter.

name Specify the name of the new Performance Explorer definition.

Top

Type (TYPE)

Specifies the type of performance data to be collected.

***STATS**
General performance program statistics are collected to help identify problem areas. This mode is mainly used as a map to help determine if and where more detailed information should be collected and analyzed.

In addition to collecting the information specified above, TYPE(*STATS) also provides the option of counting the occurrences of specific types of events. These can be counted in any of 4 counters provided.

To count the occurrences of a particular type of event, specify SLTEVT(*YES) and then choose the event-identifier(s) to be counted from the various event categories.

For example, TYPE(*STATS) SLTEVT(*YES) JOBEVT((*ALL 1)) would count all job events in counter 1.

***TRACE**
Detailed trace information is collected. This is the most detailed type of performance data collection available.

***PROFILE**
Specific programs are sampled to identify sections of code that are using larger amounts of resources. This information is very valuable when the user wants to improve the performance of a specific program or application.

Top

Profile type (PRFTYPE)

Specifies the type of profile to collect. This parameter is only valid if TYPE(*PROFILE) is specified.

***PGM** Specific programs are sampled to identify sections of code that are using larger amounts of resources. This information is very valuable when the user wants to improve the performance of a specific program or application.

***JOB** Specific jobs are sampled to identify programs and procedures that are using larger amounts of resources. This mode can provide a view of all the programs and procedures in the entire system, and is equivalent to *TRACE mode of *PMCO events with a specified interval.

Top

Job name (JOB)

Specifies which jobs are included in the Performance Explorer data collection session. A job will be added to the collection if it matches this specification.

Single values

***** Only the job that issues the STRPEX (Start Performance Explorer) command is included. All threads for that job are included.

***NONE**
No jobs are included in the data. The TASK parameter should be specified.

Element 1: Job name

Qualifier 1: Job name

***ALL** All jobs on the system are included.

name Specify the name of the job to be included in the Performance Explorer data collection session.

generic-name
Specify the generic name of the job to be included. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name. For more information about generic object names, see the i5/OS objects topic under Programming->CL in the iSeries Information Center at the Web site: http://www.ibm.com/eserver/iseries/infocenter

Qualifier 2: User

***ALL** All jobs that match the specified job name are included.

name Specify the name of the user of the job to be included.

generic-name
Specify the generic user name of the jobs to be included.

Qualifier 3: Number

***ALL** All jobs that match the specified job name and user name are included.

job-number
Specify the job number to further qualify the job name and user name.

Element 2: Thread identifier

***ALL** All threads of the specified job are included.

***INITIAL**
Only the initial thread of the specified job is included.

***SELECT**
Select the threads from a list of threads for the specified job. This value is only valid if the command is run in an interactive job.

thread-identifier
Specify the thread identifier of the job to be included. This is the thread ID as shown by the WRKJOB command.

Element 3: Subsystem

***ALLSBS**
All subsystems are searched to find jobs matching the specified job name, user, and number.

name Specify the subsystem name of the jobs to be included.

generic-name
Specify the generic subsystem name of the jobs to be included.

Top

Task name (TASK)

Specifies which licensed internal code (LIC) tasks are included in the Performance Explorer data collection session. Up to 8 task names may be specified.

Note: LIC tasks can be obtained from the Performance Tools reports and WRKSYSACT command. There is no guarantee that LIC task names will remain the same or exist from system to system or release to release.

Single values

***NONE**
No LIC tasks on the system are included. The JOB parameter should be specified.

***ALL** All LIC tasks on the system are included.

Other values

name Specify the name of the task to be included in the Performance Explorer data collection session.

generic-name
Specify the generic name of the task to be included. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name.

Top

Program to monitor (PGM)

Specifies the program whose performance profile data is collected. This parameter is only valid if TYPE(*PROFILE) is specified. Up to 16 programs may be specified.

Element 1: Program

Qualifier 1: Program

***ALL** All programs in the specifed library are sampled.

name Specify the name of the program to be sampled.

Qualifier 2: Library

***LIBL** The library list of the job that issues the STRPEX command is searched to find the specified program or service program.

name Specify the library which contains the program or service program.

Element 2: Module

***ALL** All modules in the program or service program will be sampled. If sampling an OPM (Original Program Model) program, specify *ALL for this element.

name Specify the name of the module within the program or service program that is to be sampled.

Element 3: Procedure

***ALL** All procedures in the specified module are sampled.

name Specify the name of a procedure within the specified module that is to be sampled. Specify the procedure name within single quotes if the procedure name contains lower case characters.

Element 4: Type

***PGM** The program being specified is a program (*PGM) object.

***SRVPGM**
The program being specified is a service program (*SRVPGM) object.

Element 5: Pane size

The pane size is the number of consecutive program instruction addresses assigned to each counter. The smaller the pane size, the more fine-grained the program profile information will be.

4 The default pane size is 4.

number
Specify the pane size to use for the program. Valid values are 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096.

Top

Data organization (DTAORG)

Specifies how the data is organized.

Note: This parameter is only valid if TYPE(*STATS) is specified.

***FLAT** The Performance Explorer tool will not collect data for a parent-child relationship.

***HIER** The Performance Explorer tool will collect data for a parent-child relationship.

Top

Maximum storage to use (MAXSTG)

Specifies the maximum amount of storage, in kilobytes (K), that the Performance Explorer tool uses for collecting trace data.

Note: This parameter is only valid if TYPE(*TRACE) is specified.

10000 Up to 10000 kilobytes of storage is used.

maximum-K-bytes
Specifies the requested maximum amount of storage, in kilobytes (K), to use for the collected trace records. The system calculates the minimum amount of storage that is necessary for the trace. This minimum storage size calculation depends on the system's processor configuration. The minimum amount of storage may be significantly larger than the size specified on the MAXSTG parameter. The system uses the larger of the two values.

Top

Trace full (TRCFULL)

Specifies whether the trace records wrap (replace oldest records with new records) or whether the trace stops when all of the storage specified by the MAXSTG parameter has been used.

Note: This parameter is only valid if TYPE(*TRACE) is specified.

***STOPTRC**
Tracing stops when the trace file is full of trace records.

***WRAP**
When the trace file is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

Top

Merge job data (MRGJOB)

Specifies whether the data from different jobs should be merged together or kept separate.

Note: This parameter is only valid if DTAORG(*FLAT) is specified.

***YES** The data from individual jobs is merged.

***NO** The data from individual jobs is kept separate.

Top

Add threads/tasks option (ADDTHDOPT)

Specifies what types of threads and tasks should be included in the Performance Explorer session based on the creation time of the threads and tasks relative to the start time of the Performance Explorer session.

***ALL** The collection will include all threads and tasks that satisfy the criteria specified by the JOB and TASK parameters.

***NEW** The collection will include only new threads and tasks that satisfy the criteria specified by the JOB and TASK parameters and are created after the Performance Explorer session is started. Any threads or tasks that are active at the time the Performance Explorer session is started will not be included.

***CURRENT**
The collection will include only threads and tasks that satisfy the criteria specified by the JOB and TASK parameters and are active at the time the Performance Explorer session is started. Any threads or tasks created after the start of the Performance Explorer session will not be included.

Top

List all jobs/tasks (LSTALLJOB)

Specifies whether summary data should be listed for all jobs and tasks in the system, or only those jobs and tasks specified on the JOB or TASK parameter.

***NO** Data for a job or task is listed in the task information file only if that job or task is specified on the JOB or TASK parameter.

***YES** Data for all jobs and tasks is listed in the task information file, even if the job or task is not specified on the JOB or TASK parameter.

Top

Sampling interval (INTERVAL)

Specifies the interval at which samples are taken of the program. A small interval will cause a high number of samples to be taken, and will also cause higher overhead. The smaller the sampling interval, the larger the amount of trace data collected.

Note: This parameter is valid only when TYPE(*PROFILE) or TYPE(*TRACE) is specified.

Element 1: Milliseconds

***NONE**
Interval is not used.

0.1-200.0
Specify the interval at which samples of the program are taken. Valid values range from 0.1 to 200.0 milliseconds.

Element 2: Randomize

***FIXED**
The sampling interval will always be the value specified for element 1 of this parameter.

***VARY**
The sampling interval will be approximately the value specified for element 1 of this parameter. The specified interval will be changed each time by adding or subtracting a small random percentage of the interval. This may be necessary to eliminate harmonics when the set of threads/tasks in a collection have settled into a steady state sequence of machine cycles that do not vary in order or length during the collection.

Top

Program bracketing events (PGMBKTEVT)

Specifies which program call flow events are included in the TYPE(*STATS) definition.

Note: This parameter is only valid if TYPE(*STATS) is specified.

***DFT** Statistics are to be collected on *MISTREND events, *MIENTRYEXIT events for programs and service programs, and *JVA events.

***MISTREND**
Statistics are to be collected on all machine instructions.

***MIENTRYEXIT**
Statistics are to be collected on programs and procedures. This includes any program that has been compiled at optimization level 30 or below. Optimization level 40 programs are also enabled, but only for procedures that stack a frame on the invocation stack when called (non-leaf procedures).

***JVA** Statistics are to be collected on Java methods. This includes interpreted Java and Java programs running in JIT mode where the property 'os400.enbpfrcol=1' has been set. This is only for classic VM, not IBM Technology for Java.

***PRC** As of V6R1 *PRC will behave exactly like *MIENTRYEXIT.

Top

Trace type (TRCTYPE)

Specifies what type of trace performance data to be collected. Additional information about the TRCTYPE options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if TYPE(*TRACE) is specified.

Single values (up to 12 repetitions)

***SLTEVT**
Only selected individual events (xxxEVT parameters) and machine instructions (MCHINST parameter) are included in the trace definition.

Note: If TRCTYPE(*SLTEVT) is specified, SLTEVT(*YES) must also be specified.

Other values (up to 12 repetitions)

***CALLRTN**
Specifies that call return events are included in the trace definition. Call return events occur when a program is entered and exited as well as when certain machine instructions are started and completed.

***BASIC**
Specifies that events relative to general performance analysis are included in the trace definition. This option should be used when it is unclear as to what type of performance problem determination is necessary.

***DSKIO1**
Specifies that events associated with disk input/output operations are included in the trace definition.

***DSKIO2**
Specifies that events associated with disk input/output operations plus higher level requests to do input/output operations are included in the trace definition.

***DSKSVR**
Specifies that events associated with disk server operations are included in the trace definition.

***DSKSTG**
Specifies that events associated with disk storage consumption are included in the trace definition.

***VRTADR**
Specifies that events associated with virtual address assignment are included in the trace definition.

***PGMACT**
Specifies that events associated with program activations and deactivations are included in the trace definition.

***FILEOPEN**
Specifies that events associated with file (*FILE) opens are included in the trace definition.

***PRFDTA**
Specifies that events associated CPU instruction profiling are included in the trace definition.

***TASKSWT**
Specifies that events associated with tasking are included in the trace definition.

***HEAP**
Specifies that events associated with heap storage are included in the trace definition.

Top

Specific events (SLTEVT)

For trace mode (TYPE(*TRACE)) collections, SLTEVT allows individual machine instructions and events to be specified in addition to the categories of events available with the TRCTYPE parameter. For statistics mode (TYPE(*STATS)) collections, SLTEVT allows individual events to be counted.

Note: This parameter is only valid if TYPE(*TRACE) or TYPE(*STATS) is specified.

***NO** Do not allow selection of specific events.

***YES** Allow selection of specific events.

Top

Machine instructions (MCHINST)

Specifies which machine instructions are included in the Performance Explorer data collection session. Use this parameter in conjunction with the PGMEVT parameter. MCHINST controls which of the possible machine instructions are to be included in the collection. PGMEVT(*MISTR and *MIEND) controls whether any machine instructions are included in the collection.

Note: This parameter is only valid if TYPE(*TRACE) and SLTEVT(*YES) is specified.

Single values

***ALL** All machine instructions that are available for collection are included.

***NONE**
No machine instructions available for collection are included.

Other values (up to 50 repetitions)

name Specify the name of the machine instruction to be included in the Performance Explorer data collection session.

Note: Specifying machine instructions should be done only under the direction of your iSeries service representative.

Top

Program events (PGMEVT)

Specifies which program call flow events are included in the trace definition. Additional information about the PGMEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if TYPE(*TRACE) and SLTEVT(*YES) are specified.

Single values

***NONE**
No program call flow events are included in the trace definition.

***ALL** All program call flow events are included in the trace definition.

Element 1: Event identifier

event-identifier
Specify the call flow event identifier to be included in the trace definition.

Note: The *MIENTRY, *MIEXIT, *PRCENTRY and *PRCEXIT events are always available in programs that have been compiled at optimization level 30 or below. For optimization level 40, these events are available only for the procedures that stack a frame on the invocation stack when called (non-leaf procedures).

The *JVAENTRY and *JVAEXIT are available in interpreted Java and Java programs running in JIT mode where the property 'os400.enbpfrcol=1' has been set. This is only for classic VM, not IBM Technology for Java.

Element 2: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Base events (BASEVT)

Specifies which base events are included in the definition. Additional information about the BASEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No base events are included in the definition.

***ALL** All base events are included in the trace mode definition, or counted in the statistics mode definition. If TYPE(*STATS), you can specify which event-identifiers are to be counted. Some of the base event event-identifiers cannot be counted. An informational message will be shown if a definition attempts to count all base events.

Element 1: Event identifier

event-identifier
Specify the base event identifier to be traced or counted.

Element 2: Event counter

***NONE**
No base events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events. Values other than *FORMAT1 are valid for only the *PMCO and *SWOQ events. For all other events, *FORMAT1 will be used regardless of what format is specified.

*FORMAT2 for the *PMCO event will collect the same data as *FORMAT1 plus an additional 16 levels of the call stack.

*FORMAT3 for the *PMCO event will collect the same data as *FORMAT1 plus 4 additional PMC registers.

*FORMAT2 for the *SWOQ event will collect the same data as *FORMAT1 plus an additional 16 levels of the call stack.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Storage events (STGEVT)

Specifies which auxiliary storage management events are included in the definition. Additional information about the STGEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No auxiliary storage management events are included in the definition.

***ALL** All auxiliary storage management events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the auxiliary storage management event identifier to be included.

Element 2: Event counter

***NONE**
No auxiliary storage management events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Disk events (DSKEVT)

Specifies which disk events are included in the definition. Additional information about the DSKEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No disk events are included in the definition.

***ALL** All disk events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

***ALLSTR**
All disk start events are traced, if TYPE(*TRACE), or counted, if TYPE(*STATS).

event-identifier
Specify the disk event identifier to be included.

Element 2: Event counter

***NONE**
No disk events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Fault events (FAULTEVT)

Specifies which page fault events are included in the definition. Additional information about the FAULTEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No page fault events are included in the definition.

***ALL** All page fault events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the page fault event identifier to be included.

Element 2: Event counter

***NONE**
No page fault events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Job events (JOBEVT)

Specifies which job or process related events are included in the definition. Additional information about the JOBEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No job or process related events are included in the definition.

***ALL** All job or process related events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the job or process related event identifier to be included.

Element 2: Event counter

***NONE**
No job or process related events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Lock events (LCKEVT)

Specifies which lock or seize events are included in the definition. Additional information about the LCKEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No lock or seize events are included in the definition.

***ALL** All lock or seize events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the lock or seize event identifier to be included.

Element 2: Event counter

***NONE**
No lock or seize events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

SAR events (SAREVT)

Specifies which segment address register events are included in the definition. Additional information about the SAREVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No segment address register events are included in the definition.

***ALL** All segment address register events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the segment address register event identifier to be included.

Element 2: Event counter

***NONE**
No segment address register events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Expert cache events (EXPCCHEVT)

Specifies which expert cache events are included in the definition. Additional information about the EXPCCHEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No expert cache events are included in the definition.

***ALL** All expert cache events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the expert cache event identifier to be included.

Element 2: Event counter

***NONE**
No expert cache events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Disk server events (DSKSVREVT)

Specifies which disk server events are included in the definition. Additional information about the DSKSVREVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No disk server events are included in the definition.

***ALL** All disk server events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the disk server event identifier to be included.

Element 2: Event counter

***NONE**
No disk server events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Operating System events (OSEVT)

Specifies which operating system events are included in the definition. Additional information about the OSEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No operating system events are included in the definition.

***ALL** All operating system events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the operating system event identifier to be included.

Element 2: Event counter

***NONE**
No operating system events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Java events (JVAEVT)

Specifies which Java events are included in the definition. These events are only for classic VM, not IBM Technology for Java. Additional information about the JVAEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No Java events are included in the definition.

***ALL** All Java events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the Java event identifier to be included.

Element 2: Event counter

***NONE**
No Java events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

*FORMAT2 for the *OBJCRT, *LCKSTR, and *UNLCK events will collect the same data as *FORMAT1 plus an additional 16 levels of the call stack.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Communications events (CMNEVT)

Specifies which communications events are included in the definition. Additional information about the CMNEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No communications events are included in the definition.

***ALL** All communications events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the communications event identifier to be included.

Element 2: Event counter

***NONE**
No communications events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events. Values other than *FORMAT1 are valid for only the *SOCKETS events. For all other events, *FORMAT1 will be used regardless of what format is specified.

*FORMAT2 for the *SOCKETS event will collect the same data as *FORMAT1 except the application data contains 128 bytes of data instead of 64.

*FORMAT3 for the *SOCKETS event will collect the same data as *FORMAT1 except the application data contains 256 bytes of data instead of 64.

*FORMAT4 for the *SOCKETS event will collect the same data as *FORMAT1 except the application data contains 500 bytes of data instead of 64.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Application events (APPEVT)

Specifies which application events are included in the definition. Additional information about the APPEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No application events are included in the definition.

***ALL** All application events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the application event identifier to be included.

Element 2: Event counter

***NONE**
No application events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

PASE events (PASEEVT)

Specifies which Portable Application Solution Environment (PASE) events are included in the definition. Additional information about the PASEEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No PASE events are included in the definition.

***ALL** All PASE events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the PASE event identifier to be included.

Element 2: Event counter

***NONE**
No PASE events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

File server events (FILSVREVT)

Specifies which iSeries NetServer, File Server and Network File System Server and Client events are included in the definition. Additional information about the FILSVREVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No iSeries NetServer, File Server and Network File System Server and Client events are included in the definition.

***ALL** All iSeries NetServer, File Server and Network File System Server and Client events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the iSeries NetServer, File Server and Network File System Server and Client event identifier to be included.

Element 2: Event counter

***NONE**
No iSeries NetServer, File Server and Network File System Server and Client events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Synchronization events (SYNCEVT)

Specifies which synchronization events are included in the definition. Additional information about the SYNCEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No synchronization events are included in the definition.

***ALL** All synchronization events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the synchronization event identifier to be included.

Element 2: Event counter

***NONE**
No synchronization events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Journal events (JRNEVT)

Specifies which journal events are included in the definition. Additional information about the JRNEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No journal events are included in the definition.

***ALL** All journal events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the journal event identifier to be included.

Element 2: Event counter

***NONE**
No journal events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Database events (DBEVT)

Specifies which database events are included in the definition. Additional information about the DBEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No database events are included in the definition.

***ALL** All database events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the database event identifier to be included.

Element 2: Event counter

***NONE**
No database events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Resources affinity events (RSCAFNEVT)

Specifies which resources affinity events are included in the definition. Additional information about the RSCAFNEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No resources affinity events are included in the definition.

***ALL** All affinity events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the affinity event identifier to be included.

Element 2: Event counter

***NONE**
No affinity events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Virtual I/O events (VRTIOEVT)

Specifies which virtual I/O events are included in the definition. Additional information about the VRTIOEVT options can be found in the Performance Management information at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html.

Note: This parameter is only valid if SLTEVT(*YES) is specified.

Single values

***NONE**
No virtual I/O events are included in the definition.

***ALL** All virtual I/O events are included in the trace mode definition, or counted in the statistics mode definition.

Element 1: Event identifier

event-identifier
Specify the virtual I/O event identifier to be included.

Element 2: Event counter

***NONE**
No virtual I/O events are counted.

event-counter
If TYPE(*STATS), events can be counted in one of four counters (1,2,3 or 4). The event counter value is ignored if TYPE(*TRACE).

Element 3: Event format

The event format describes what data is collected for this event. *FORMAT1 provides the data used for most data analysis. The other formats allow for collection of other data related to these events.

***FORMAT1**
The first event format is used to collect data for this event.

***FORMAT2**
The second event format is used to collect data for this event.

***FORMAT3**
The third event format is used to collect data for this event.

***FORMAT4**
The fourth event format is used to collect data for this event.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the Performance Explorer definition.

***BLANK**
Text is not specified.

'description'
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Examples

Example 1: Using TYPE(*TRACE)

```
ADDPEXDFN   DFN(TEST1)  TYPE(*TRACE)  JOB(*)  MAXSTG(5000)
```

This command adds a new performance definition named TEST1, which will result in a member named TEST1 being added to file QAPEXDFN in library QUSRSYS. When this definition is used to start a

performance explorer session (STRPEX command), detailed trace information will be collected for the job that invoked the STRPEX command. A maximum of 5000 kilobytes of trace data will be collected. When the trace record storage area is full no more trace records will be collected.

Example 2: Using TYPE(*PROFILE)

```
ADDPEXDFN   DFN(TEST2)  TYPE(*PROFILE)
            PGM((MYLIB/MYSRVPGM1 *ALL *ALL *SRVPGM))
```

This command adds a new performance explorer definition named TEST2. When this definition is used to start a performance explorer session (STRPEX command), performance profile information for service program MYSRVPGM1 in library MYLIB will be collected.

Example 3: Using TYPE(*TRACE)

```
ADDPEXDFN   DFN(TEST3)  TYPE(*TRACE)  JOB(*ALL)
            TRCTYPE(*CALLRTN *DSKIO1)
            TEXT('Trace definition example')
```

This command adds a new performance explorer definition named TEST3. When this definition is used to start a performance explorer session (STRPEX command), performance trace information for program call/return and disk input/output operation will be collected.

Example 4: Counting SAR Events

```
ADDPEXDFN   DFN(TEST4)  TYPE(*STATS)  SLTEVT(*YES)
            SAREVT((*ALL 1))
            TEXT('Count all SARs in counter 1')
```

This command adds a new performance explorer definition named TEST4. When this definition is used to start a session (STRPEX command), performance statistics for program and procedure call/return operations will be collected. In addition, all segment address register (SAR) events that occur will be counted in counter 1.

Top

Error messages

***ESCAPE Messages**

CPFAF10

Definition or filter already exists.

Top

Add PEX Filter (ADDPEXFTR)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The Add Performance Explorer Filter (ADDPEXFTR) command adds a new Performance Explorer (PEX) filter to the system. Each filter is stored as a member in the QAPEXFTR file in library QUSRSYS. A Performance Explorer filter identifies the performance data that is to be collected during a Performance Explorer session, and is meant to limit the amount of data collected by specifying a compare value for specific events. If the data in the event matches the compare value, then the data will be collected. If not, the data is discarded. The filter is specified on the STRPEX (Start Performance Explorer) command.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. To use this command you must have *SERVICE special authority, or be authorized to the Service Trace function of i5/OS through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
3. You must have *EXECUTE authority to the libraries for PGMTRG, PGMFTR, OBJFTR, and *X authority to the directories for JVATRG, JVAFTR, JVACLSFTR, and PATHFTR if these parameters are specified.
4. The following user profiles have private authorities to use the command:
   - QPGMR
   - QSRV
5. Two threads within the same job will not be allowed to run ADDPEXFTR at the same time. The thread that issued ADDPEXFTR first will run the command to completion while the second ADDPEXFTR waits.

Top

Parameters

Keyword	Description	Choices	Notes
FTR	Filter	*Name*	Required, Positional 1
PGMTRG	Program trigger	*Element list*	Optional
	Element 1: Program	*Qualified object name*	
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*	
	Element 2: Module	*Name*, *ALL	
	Element 3: Procedure	*Character value*, ***PEP**, *ALL	
	Element 4: Type	***PGM**, *SRVPGM	
	Element 5: Trigger option	***ENTRYEXIT**, *ENTRY, *ENTRYEXITNOPGM, *ENTRYNOPGM	

Keyword	Description	Choices	Notes
JVATRG	Java trigger	*Element list*	Optional
	Element 1: Java package	*Character value*, ***NONE**	
	Element 2: Java class	*Character value*, ***ALL**	
	Element 3: Java method	*Character value*, ***ALL**	
	Element 4: Trigger option	***ENTRYEXIT**, *ENTRY, *ENTRYEXITNOPGM, *ENTRYNOPGM	
PGMFTR	Program filter	*Element list*	Optional
	Element 1: Relational operator	***EQ**, *NE	
	Element 2:	Values (up to 16 repetitions): *Element list*	
	Element 1: Program	*Qualified object name*	
	Qualifier 1: Program	*Name*, *ALL	
	Qualifier 2: Library	*Name*	
	Element 2: Module	*Name*, ***ALL**	
	Element 3: Procedure	*Character value*, ***ALL**	
	Element 4: Type	***PGM**, *SRVPGM	
JVAFTR	Java method filter	*Element list*	Optional
	Element 1: Relational operator	***EQ**, *NE	
	Element 2:	Values (up to 5 repetitions): *Element list*	
	Element 1: Java package	*Character value*, ***NONE**	
	Element 2: Java class	*Character value*, ***ALL**	
	Element 3: Java method	*Character value*, ***ALL**	
JVACLSFTR	Java class filter	*Element list*	Optional
	Element 1: Relational operator	***EQ**, *NE	
	Element 2:	Values (up to 5 repetitions): *Element list*	
	Element 1: Java package	*Character value*, ***NONE**	
	Element 2: Java class	*Character value*, ***ALL**	
	Element 3: Array	*Character value*, ***NOARRAY**, *CLASSARRAY, *VOIDARRAY, *BOOLEANARRAY, *CHARARRAY, *FLOATARRAY, *DOUBLEARRAY, *BYTEARRAY, *SHORTARRAY, *INTARRAY, *LONGARRAY	
OBJFTR	Object filter	*Element list*	Optional
	Element 1: Relational operator	***EQ**, *NE	
	Element 2:	Values (up to 5 repetitions): *Element list*	
	Element 1: Object	*Qualified object name*	
	Qualifier 1: Object	*Name*, *ALL	
	Qualifier 2: Library	*Name*	
	Element 2: Member, if data base file	*Name*	
	Element 3: Object type	*DTAARA, *DTAQ, *FILE, *FLR, *JRN, *JRNRCV, *LIB, *MSGQ, *PGM, *SRVPGM, *USRPRF, *USRIDX, *USRQ, *USRSPC	
PATHFTR	Path filter	*Element list*	Optional
	Element 1: Relational operator	***EQ**, *NE	
	Element 2: Path	Values (up to 5 repetitions): *Path name*	

Keyword	Description	Choices	Notes
MEMFTR	Memory filter	*Element list*	Optional
	Element 1: Relational operator	***EQ**, *NE	
	Element 2: Pool identifier	Values (up to 5 repetitions): 1-64	
DSKFTR	Disk filter	*Element list*	Optional
	Element 1: Relational operator	***EQ**, *NE	
	Element 2: Disk identifier	Values (up to 5 repetitions): *Integer*	
DSKTIMFTR	Disk time filter	*Element list*	Optional
	Element 1: Relational operator	***GT**, *LT	
	Element 2: Disk time	Values (up to 5 repetitions): *Decimal number*	
ASPFTR	ASP filter	*Element list*	Optional
	Element 1: Relational operator	***EQ**, *NE	
	Element 2: ASP identifier	Values (up to 5 repetitions): *Integer*	
IPFTR	IP filter	*Element list*	Optional
	Element 1: Relational operator	***EQ**, *NE	
	Element 2:	Values (up to 5 repetitions): *Element list*	
	Element 1: Address family	1-255, ***INET**, *INET6, *UNIX	
	Element 2: Communication type	1-255, ***STREAM**, *DGRAM, *RAW, *SEQPACKET	
	Element 3: Local internet address	*Character value*, ***ALL**	
	Element 4: Remote internet address	*Character value*, ***ALL**	
	Element 5: Local port	1-65535, ***ALL**	
	Element 6: Remote port	1-65535, ***ALL**	
USRDFNFTR	User defined filter	Values (up to 10 repetitions): *Element list*	Optional
	Element 1: Relational operator	***EQ**, *NE, *LT, *GT	
	Element 2: Event type	1-31	
	Element 3: Event subtype	1-31	
	Element 4: Data offset	*Integer*	
	Element 5: Data type	***CHAR**, *HEX, *INT1, *INT2, *INT4, *INT8, *UINT1, *UINT2, *UINT4, *UINT8, *SECONDS, *MILLISEC, *MICROSEC	
	Element 6: Compare value	Values (up to 5 repetitions): *Character value*	
	Element 7: Trigger option	0-8, ***NONE**, *START, *END, *COLLSTART, *COLLEND	
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Top

Filter (FTR)

Specifies the name of the Performance Explorer filter to be added. If the specified filter already exists in the QAPEXFTR file in library QUSRSYS, an error condition will occur. The user can either change the filter name or remove the existing filter using the Remove Performance Explorer Filter (RMVPEXFTR) command, and try this command again.

This is a required parameter.

name Specify the name of the new Performance Explorer filter.

Top

Program trigger (PGMTRG)

If a procedure entry event (*PRCENTRY) or MI entry event (*MIENTRY) occurs that matches this trigger specification, then Performance Explorer will begin collecting all events specified in the Performance Explorer definition used for the active Performance Explorer session. The events will be collected only for the thread where the trigger occurs. When the procedure exit event (*PRCEXIT) or MI exit event (*MIEXIT) occurs that matches the trigger specification, Performance Explorer will stop collecting the events specified in the Performance Explorer definition.

Note: The *MIENTRY, *MIEXIT, *PRCENTRY and *PRCEXIT events are always available in programs that have been compiled at optimization level 30 or below. For optimization level 40, these events are available only for the procedures that stack a frame on the invocation stack when called (non-leaf procedures).

The *JVAENTRY and *JVAEXIT are available in interpreted Java and Java programs running in JIT mode where the property 'os400.enbpfrcol=1' has been set. This is only for classic VM, not IBM Technology for Java.

Element 1: Program

Qualifier 1: Program

name Specify the name of the trigger program.

Qualifier 2: Library

name Specify the library which contains the program or service program.

Element 2: Module

***ALL** All modules in the specified program will act as the trigger.

name Specify the module within the program or service program that contains the procedure that is to be the trigger. This is required only for ILE programs and service programs.

Element 3: Procedure

***PEP** The program entry procedure will act as the trigger. This is not valid for programs of type *SRVPGM.

***ALL** All procedures in the specified module will act as the trigger.

name Specify a specific procedure within the specified module that is to be the trigger. Specify the procedure name within single quotes if the procedure name contains lower case characters.

Element 4: Type

Indicate the type of program being specified.

***PGM** The program being specified is a program (*PGM) object.

***SRVPGM**
The program being specified is a service program (*SRVPGM) object.

Element 5: Trigger option

***ENTRYEXIT**
The specified trigger procedure enables the collection of events at procedure entry time. At procedure exit, the collection of events is disabled.

***ENTRY**
The specified trigger procedure enables the collection of events at procedure entry time. The collection of events is enabled for the duration of the Performance Explorer session.

***ENTRYEXITNOPGM**
The specified trigger procedure enables the collection of events (except program events) at procedure entry time. At procedure exit, the collection of events is disabled. This option is not valid for STATS mode.

***ENTRYNOPGM**
The specified trigger procedure enables the collection of events (except program events) at procedure entry time. The collection of events is enabled for the duration of the Performance Explorer session. This option is not valid for STATS mode.

Top

Java trigger (JVATRG)

If a Java method entry event (*JVAENTRY) occurs that matches this trigger specification, then Performance Explorer will begin collecting all events specified in the Performance Explorer definition used for the active Performance Explorer session. The events will be collected only for the thread where the trigger occurs. When the Java method exit event (*JVAEXIT) occurs that matches the trigger specification, Performance Explorer will stop collecting the events specified in the Performance Explorer definition.

The *JVAENTRY and *JVAEXIT hooks are always enabled in interpreted Java. You can also enable these hooks when running in JIT mode by specifying the property 'os400.enbpfrcol=1'. This is only for classic VM, not IBM Technology for Java.

Element 1: Java package

name Specify the name of the Java package that contains the method to use as the trigger.

Element 2: Java class

name Specify a class within the package that contains the method to use as the trigger.

Element 3: Java method

name Specify a method to use as the trigger.

Element 4: Trigger option

***ENTRYEXIT**
The specified trigger method enables the collection of events at Java method entry time. At Java method exit, the collection of events is disabled.

***ENTRY**
The specified trigger method enables the collection of events at Java method entry time. The collection of events is enabled for the duration of the Performance Explorer session.

***ENTRYEXITNOPGM**
The specified trigger method enables the collection of events (except program events) at method entry time. At method exit, the collection of events is disabled. This option is not valid for STATS mode.

***ENTRYNOPGM**
The specified trigger method enables the collection of events (except program events) at method entry time. The collection of events is enabled for the duration of the Performance Explorer session. This option is not valid for STATS mode.

Top

Program filter (PGMFTR)

Specifies the program comparisons to use for this filter.

Element 1: Relational operator

***EQ** Events having program data that matches the specified program are included in the data collected by Performance Explorer.

***NE** Events having program data that matches the specified program are excluded from the data collected by Performance Explorer. These events will not show up in the *MGTCOL object or the Performance Explorer database.

Element 2: Filter entry

Specify the elements for each filter entry.

Element 1: Program

Qualifier 1: Program

***ALL** All programs in the specified library will pass the filter.

name Specify the name of the program to be used as a compare value for the program filter.

Qualifier 2: Library

name Specify the library which contains the program or service program.

Element 2: Module

***ALL** All modules in the program or service program will pass the filter. If filtering an OPM (Original Program Model), specify *ALL for this element.

name Specify a specific module within the program or service program to be used as a compare value for the program filter. This is required only for ILE programs and service programs.

Element 3: Procedure

***ALL** All procedures in the specified module are used as a compare value for the program filter.

name Specify a procedure to use as the filter compare value. Specify the procedure name within single quotes if the procedure name contains lower case characters.

Element 4: Type

Indicate the type of program being specified.

***PGM** The program being specified is a program (*PGM) object.

***SRVPGM**
The program being specified is a service program (*SRVPGM) object.

Top

Java method filter (JVAFTR)

Specifies the Java package, class, and methods to be used as compare values for the Java filter.

Element 1: Relational operator

***EQ** Events having Java data that match the specified packages,classes, and methods are included in the data collected by Performance Explorer.

***NE** Events having Java data that matches the specified packages, classes, and methods are excluded from the collection and will not show up in the *MGTCOL object or the Performance Explorer database.

Element 2: Filter entry

Specify the elements for each filter entry.

Element 1: Java package

***NONE**
The Java class and methods are not in a package.

name Specify the name of the Java package to be used as a compare value for the filter.

Element 2: Java class

***ALL** All classes in the specified package will pass the Java filter.

name Specify a class within the package to be used as a compare value for the filter.

Element 3: Java method

***ALL** All methods in the specified class and package will pass the filter.

name Specify a method to use as the filter compare value.

Top

Java class filter (JVACLSFTR)

Specifies the Java package and class to be used as compare values for the Java class filter.

Element 1: Relational operator

***EQ** Events having Java data that match the specified packages and classes are included in the data collected by Performance Explorer.

***NE** Events having Java data that matches the specified packages and classes are excluded from the collection and will not show up in the *MGTCOL object or the Performance Explorer database.

Element 2: Filter entry

Specify the elements for each filter entry.

Element 1: Java package

***NONE**
The Java classes are not in a package.

name Specify the name of the Java package to be used as a compare value for the filter.

Element 2: Java class

***ALL** All classes in the specified package will pass the Java filter.

name Specify a class within the package to be used as a compare value for the filter.

Element 3: Array

***NOARRAY**
The specified class is not an array of classes.

***CLASSARRAY**
The specified class is an array of classes.

***VOIDARRAY**
The array of void primitive class is to be used as a compare value for the filter.

***BOOLEANARRAY**
The array of boolean primitive class is to be used as a compare value for the filter.

***CHARARRAY**
The array of char primitive class is to be used as a compare value for the filter.

***FLOATARRAY**
The array of float primitive class is to be used as a compare value for the filter.

***DOUBLEARRAY**
The array of double primitive class is to be used as a compare value for the filter.

***BYTEARRAY**
The array of byte primitive class is to be used as a compare value for the filter.

***SHORTARRAY**
The array of short primitive class is to be used as a compare value for the filter.

***INTARRAY**
The array of int primitive class is to be used as a compare value for the filter.

***LONGARRAY**
The array of long primitive class is to be used as a compare value for the filter.

Top

Object filter (OBJFTR)

Specifies the object comparisons to use for this filter.

Element 1: Relational operator

***EQ** Events whose object name, library name, member (if a file) and object type matches this specification will be included in the collected trace data.

***NE** Events whose object name, library name, member (if a file) and object type matches this specification will be excluded from the collected trace data. These events will not show up in the *MGTCOL object or the Performance Explorer database.

Element 2: Filter entry

Specify the elements for each filter entry.

Element 1: Object

Qualifier 1: Object

***ALL** All objects in the specified library will pass the filter. Object type must be blank if *ALL is specified for the object.

name The name of the object to be used as a compare value in the object filter.

Qualifier 2: Library

name The library which contains the object to use as a compare value.

Element 2: Member, if data base file

name The member name to use as the compare value. This value is valid for physical or logical *FILE objects only. Specify *N for other objects.

Element 3: Object type

***DTAARA**
The specified object is a data area.

***DTAQ**
The specified object is a data queue.

***FILE** The specified object is a physical or logical database file.

***FLR** The specified object is a folder.

***JRN** The specified object is a journal.

***JRNRCV**
The specified object is a journal receiver.

***LIB** The specified object is a library. Note: this filters on the library object only, not on the objects contained in the library. Use an object name of *ALL to filter on all objects contained in a library.

***MSGQ**
The specified object is a message queue.

***PGM** The specified object is a program.

***SRVPGM**
The specified object is a service program.

***USRPRF**
The specified object is a user profile.

***USRIDX**
The specified object is a user index.

***USRSPC**
The specified object is a user space.

***USRQ**
The specified object is a user queue.

Top

Path filter (PATHFTR)

Specifies the Integrated File System object path name comparisons to use for this filter.

Element 1: Relational operator

***EQ** All events that have an object path that matches the specified object path are included in the data collected by Performance Explorer.

***NE** All events that have an object path data that matches the specified object path are discarded. These events will not show up in the *MGTCOL object or the Performance Explorer database.

Element 2: Path

object-path
The object path to use as a compare value for this filter.

Top

Memory filter (MEMFTR)

Specifies the memory pool comparisons to use for this filter.

Element 1: Relational operator

***EQ** All events that have pool identifier data that matches the specified pool are included in the data collected by Performance Explorer.

***NE** All events that have pool identifier data that matches the specified pool are discarded. These events will not show up in the *MGTCOL object or the Performance Explorer database.

Element 2: Pool identifier

pool-identifier
The system pool ID to use as a compare value for this filter. This pool ID corresponds to the pool identifier as shown on the WRKACTJOB command or on the output of PRTPEXRPT of type *TRACE.

Top

Disk filter (DSKFTR)

Specifies the disk unit comparisons to use for this filter.

Element 1: Relational operator

***EQ** All events that have disk identifier data that matches the specified disk are included in the data collected by Performance Explorer.

***NE** All events that have disk identifier data that matches the specified disk will be discarded. These events will not show up in the *MGTCOL object or the Performance Explorer database.

Element 2: Disk identifier

disk-identifier
The disk identifier used as a compare value for this filter. This disk identifier corresponds to the disk unit as shown on the WRKDSKSTS command or the output of PRTPEXRPT of type *TRACE. If a disk is mirrored, this identifier applies to both disks in the mirrored pair.

Top

Disk time filter (DSKTIMFTR)

Specifies the disk response time compare value (in milliseconds) to use for this filter.

Element 1: Relational operator

***GT** All events that have disk response time data that is greater than the specified disk response time criteria are included in the data collected by Performance Explorer.

***LT** All events that have disk response time data that is less than the specified disk response time criteria are included in the data collected by Performance Explorer.

Element 2: Disk time

disk-time
The disk response time in milliseconds used as a compare value for this filter. This disk response time corresponds to the total elapsed time data that is included in each disk event that represents the end of a function. For example *READEND, *WRTEND, *PGREADEND, *PGWRTEND, *RMTWRTEND, and *RMTPGWRTEND.

Top

ASP filter (ASPFTR)

Specifies the ASP (auxiliary storage pool) comparisons to use for this filter.

Element 1: Relational operator

***EQ** All events with an ASP identifier that matches the specified ASP will be included in the data collected by Performance Explorer.

***NE** All events with an ASP identifier that matches the specified ASP will discarded. These events will not show up in the *MGTCOL object or the Performance Explorer database.

Element 2: ASP identifier

ASP-identifier
The ASP identifier to use as a compare value for this filter. This ASP identifier can be set to the name of an independent ASP or to the ASP number that corresponds to the ASP value as shown on the WRKDSKSTS command or in the output of PRTPEXRPT of type *TRACE.

Top

IP filter (IPFTR)

Specifies the IP (internet protocol) information to use as a compare value for this filter.

Element 1: Relational operator

***EQ** All events with IP data that match the filter compare values will be included in the data collected by Performance Explorer.

***NE** All events with IP data that match the filter compare values will be discarded. These events will not show up in the *MGTCOL object or the Performance Explorer database.

Element 2: Filter entry

Specify the elements for each filter entry.

Element 1: Address family

***INET** The Internet protocol will be used as part of the compare value.

***INET6**
The Internet protocol version 6 will be used as part of the compare value.

***UNIX**
The Unix protocol will be used as part of the compare value.

Element 2: Communication type

***STREAM**
A communication type of SOCK_STREAM will be used as the compare value.

***DGRAM**
A communication type of SOCK_DGRAM will be used as the compare value.

***RAW** A communication type of SOCK_RAW will be used as the compare value.

***SEQPACKET**
A communication type of SOCK_SEQPACKET will be used as the compare value.

Element 3: Local internet address

***ALL** All local IP addresses will pass this part of the IP filter.

local-IP-address
The local IP address to be used as part of the IP compare value.

Element 4: Remote internet address

***ALL** All remote IP addresses will pass this part of the IP filter.

remote-IP-address
The remote IP address to used as part of the IP compare value.

Element 5: Local port

***ALL** All local ports for the specified local address will pass this part of the IP filter.

local-port
The local port number to be used as a compare value.

Element 6: Remote port

***ALL** All remote ports for the specified address will pass this part of the IP filter.

remote-port
The remote port number to be used as the compare value.

Top

User defined filter (USRDFNFTR)

Specifies user-defined comparisons to use for this filter. This type of filter will require help from IBM service.

Element 1: Relational operator

***EQ** All events for the specified user filter will be collected if the event data matches the compare value.

***NE** All events for the specified user filter will be discarded if the event data matches the compare value. These events will not show up in the *MGTCOL object or the Performance Explorer database.

***GT** All events for the specified user filter will be collected if the event data is greater than the compare value.

***LT** All events for the specified user filter will be collected if the event data is less than the compare value.

Element 2: Event type

event-type
The event type (1-31) for the event to filter.

Element 3: Event subtype

event-subtype
The event subtype (1-31) for the event to filter.

Element 4: Data offset

data-offset
The offset into the event data to be compared against the compare value.

Element 5: Data type

Specifies how to compare the event data to the compare value.

***CHAR**
Compare as two character strings, left adjusted and padded on the right with blanks. The maximum length is 30 bytes.

***HEX** Compare as hexadecimal strings, left adjusted and padded on the right with hexadecimal zeros. The maximum length is 30 hexadecimal digits.

***INT1** The first byte of event data at the specified data offset and the compare value are compared as a signed 1-byte integers.

***INT2** The first two bytes of event data at the specified data offset and the compare value are compared as signed 2-byte integers.

***INT4** The first four bytes of event data at the specified data offset and the compare value are compared as signed 4-byte integers.

***INT8** The first eight bytes of event data at the specified data offset and the compare value are compared as signed 8-byte integers.

***UINT1**
The first byte of event data at the specified data offset and the compare value are compared as a unsigned 1-byte integers.

***UINT2**
The first two bytes of event data at the specified data offset and the compare value are compared as unsigned 2-byte integers.

***UINT4**
The first four bytes of event data at the specified data offset and the compare value are compared as unsigned 4-byte integers.

***UINT8**
The first eight bytes of event data at the specified data offset and the compare value are compared as unsigned 8-byte integers.

***SECONDS**
The specified time in seconds is converted to an internal timestamp to compare to the timestamp duration event data at the specified data offset.

***MILLISEC**
The specified time in milliseconds is converted to an internal timestamp to compare to the timestamp duration event data at the specified data offset.

***MICROSEC**
The specified time in microseconds is converted to an internal timestamp to compare to the timestamp duration event data at the specified data offset.

Element 6: Compare value

compare-value
The value used to compare against the event data. Up to five compare values can be specified. If multiple values are specified, the comparison will be made with each compare value. If any comparison is true, the event will be filtered.

Element 7: Trigger option

***NONE**
The information in this filter is not used as a trigger.

***START**
If an event occurs that meets the criteria in the filter information, subsequent events will be stored to the database. Previous to this, no events will be recorded. This option can be used to start collecting events once a certain condition has occurred. This option affects only the task or process where the trigger event occured.

***END** If an event occurs that meets the criteria in the filter information, subsequent events will not be stored to the database. This can be used to stop collecting data once a certain condition has occurred. This option affects only the task or process where the trigger event occured.

***COLLSTART**
If an event occurs that meets the criteria in the filter information, subsequent events will be stored to the database. Previous to this, no events will be recorded. This option can be used to start collecting events once a certain condition has occurred. This option affects all tasks and processes in the collection.

***COLLEND**
If an event occurs that meets the criteria in the filter information, subsequent events will not be stored to the database. This can be used to stop collecting data once a certain condition has occurred. This option affects all tasks and processes in the collection.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the Performance Explorer definition.

***BLANK**
Text is not specified.

'description'
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Examples

Example 1: Adding Disk and Memory Pool Filters

```
ADDPEXFTR   FTR(FILTER1)  DSKFTR(*EQ (1 2))  MEMFTR(*EQ 3)
```

This command adds a new performance explorer filter named FILTER1 which will result in a member named FILTER1 being added to file QAPEXFTR in library QUSRSYS. If this filter is used when starting a performance explorer session (STRPEX command), then events will be collected if they contain disk device identifier data of '1' or '2'. In addition, the pool data for that event must contain '3'. If either the disk data or the memory pool data do not match the specified filter, then that event will not be recorded.

If an event does not contain disk device or memory pool data, then the filter does not apply to that event and those events will be collected. For example, the base event *TASKSWTIN does not contain any disk or memory pool data, so this event would still be collected.

Example 2: Adding a Disk Filter

```
ADDPEXFTR   FTR(DISKFILTER)  DSKFTR(*NE (1 2))
```

This command adds a new performance explorer filter named DISKFILTER. If this filter is used when starting a performance explorer session (STRPEX command), then events will be collected if the event contains disk device name data that does not match '1' and does not match '2'.

If an event does not contain disk device name data, then the filter does not apply to that event and those events will be collected. For example, the base event *TASKSWTIN does not contain any disk data, so this event would still be collected.

Example 3: Adding an IP Filter

```
ADDPEXFTR   FTR(IPFILTER)
            IPFTR(*EQ (*INET *STREAM '1.2.3.4'))
```

This command adds a new performance explorer filter named IPFILTER. If this filter is used when starting a performance explorer session (STRPEX command), then events will be collected if a communications event has an address family of *INET, the communication type is *STREAM, and the local IP address is '1.2.3.4'.

Example 3: Adding a Java Trigger

```
ADDPEXFTR FTR(PERFSTRUTS) JVATRG('org/apache/struts/action'
        'ActionServlet' 'doGet' *ENTRYEXIT)
```

This command adds a new performance explorer filter named PERFSTRUTS. If this filter is used when starting a performance explorer session (STRPEX command), then events will start being collected when a *JVAENTRY or *JVAEXIT event occurs for the 'doGet' method in the class 'org.apache.struts.action.ActionServlet'.

Example 4: Adding a User-Defined Filter

```
ADDPEXFTR   FTR(USERFILTER)
            USRDFNFTR((*EQ 1 2 20 *CHAR ('BOB' 'SAM')))
```

This command adds a new performance explorer filter named USERFILTER. If this filter is used when starting a performance explorer session (STRPEX command), then events will be collected if the event type is '1', the event subtype is '2' and the data at offset 20 is either 'BOB' or 'SAM'.

Top

Error messages

***ESCAPE Messages**

CPFAF10

Definition or filter already exists.

Top

Add PF Constraint (ADDPFCST)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The Add Physical File Constraint (ADDPFCST) command can be used to add constraint relationships to a specified physical file. The four types of constraint relationships that you can add are referential constraints, unique constraints, primary key constraints and check constraints. All constraints are defined at the file level.

You can use constraint relationships to define dependencies between files. The relationships that you define are enforced by the system when changes occur to information in the files. When you define constraint relationships you control the **referential integrity** of the data being processed.

To define or establish a referential constraint, the parent file and the dependent file must exist. However, if the parent or dependent file has no members, the constraint only is defined (not established).

When a referential constraint is established, either an access path is created or an existing access path with matching attributes is shared. A maximum of 300 constraint relationships can be established for a file. However, only one primary key constraint can be established for a file.

You can remove a constraint by using the Remove Physical File Constraint (RMVPFCST) command. You can view all constraints for a dependent file by using the Display File Description (DSPFD) command.

Restrictions:

- You cannot add constraint relationships to system files or to program described files.
- You cannot add a constraint relationship to a file that your user job has open.
- Referential constraints cannot span auxiliary storage pools (ASPs).
- Constraints cannot be added to a file in the temporary library QTEMP.
- If a check or referential constraint is added with this command and the established referential or check constraint has records that are in check pending, the constraint is automatically changed to the disabled state.
- This command is conditionally threadsafe. In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File	*Qualified object name*	Required, Positional 1
	Qualifier 1: File	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TYPE	Constraint type	*REFCST, *UNQCST, *PRIKEY, *CHKCST	Required, Positional 2
KEY	Constraint key	Values (up to 120 repetitions): *Name*	Optional, Positional 3
CST	Constraint name	*Character value*, ***GEN**	Optional

Keyword	Description	Choices	Notes
PRNFILE	Parent file	*Qualified object name*	Optional
	Qualifier 1: Parent file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
PRNKEY	Parent key	Single values: ***PRNFILE** Other values (up to 120 repetitions): *Name*	Optional
DLTRULE	Delete rule	*CASCADE, ***NOACTION**, *SETDFT, *SETNULL, *RESTRICT	Optional
UPDRULE	Update rule	***NOACTION**, *RESTRICT	Optional
CHKCST	Check constraint	*Character value*	Optional

Top

File (FILE)

Specifies the file to which a constraint is being added. The file must be a physical file and it must allow a maximum of one member (MAXMBRS(1)).

If a referential constraint is being added, this parameter specifies the dependent file and the library containing the dependent file. The parent file is specified on the PRNFILE parameter.

This is a required parameter.

Qualifier 1: File

name Specify the name of the physical file.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Constraint type (TYPE)

Specifies the type of constraint being added to the physical file.

This is a required parameter.

***REFCST**
A referential constraint is being added.

Notes:

- Referential constraints cannot span multiple ASPs (auxiliary storage pools).
- Referential constraints cannot be added while either the parent or the dependent file is open.
- Duplicate and multiple referential constraints can be added between the same dependent and parent files if the constraint name is unique. However, the results may not match your

expectations. See the Database category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for advisory information on duplicate or multiple referential constraints.

***UNQCST**
A unique constraint is being added.

Note: Duplicate unique constraints are not allowed.

***PRIKEY**
A primary key constraint is being added. A primary key constraint is a special case of a unique constraint.

Note: Only one primary key constraint is allowed per physical file.

***CHKCST**
A check constraint is being added.

Top

Constraint key (KEY)

Specifies the constraint key, which is the definition of the access path for the type of constraint specified on the TYPE parameter. The constraint key is one or more fields that exist in the file specified on the file parameter. For referential and unique constraints, the fields specified can allow nulls (ALWNULL). For primary key constraints, the fields specified cannot allow nulls.

***REFCST**
The foreign key of a referential constraint is defined. If a referential constraint is established, a foreign key access path is added to the dependent file.

***UNQCST**
The key of a unique constraint is defined. If a unique constraint is established, a unique key access path is added to the physical file.

***PRIKEY**
The key of a primary key constraint is defined. If a primary key constraint is established, a primary key access path is added to the physical file.

***CHKCST**
The constraint key (KEY) parameter does not apply for check constraints. Check constraints use the CHKCST parameter to specify a check constraint expression.

name Specify the name of the field for the constraint key you are defining. Each field name must exist in the file specified on the FILE parameter. You can specify a maximum of 120 (but no duplicate) field names to define the constraint key, where:

- The field names are of the object type *NAME and are a maximum length of 10.
- The fields must be specified in ascending order.
- The maximum number of bytes in a key is 32768 bytes (see the Database category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for more information on this limitation).

Top

Constraint name (CST)

Specifies the name of the constraint being added.

***GEN** The system generates a constraint name.

character-value
Specify the name of the constraint. The constraint name must be unique to the library of the physical file specified on the FILE parameter. You can specify a maximum of 128 characters without delimiters, or 258 characters with quotation mark (″) delimiters.

Note: The case is preserved when lowercase characters are specified.

Top

Parent file (PRNFILE)

Specifies the parent file of a referential constraint. The file must be a physical file and it must allow a maximum of one member (MAXMBRS(1)).

Qualifier 1: Parent file

name Specify the name of the parent file of a referential constraint.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Parent key (PRNKEY)

Specifies the parent key, which is the definition of the access path on a parent file of a referential constraint. Only unique constraints or the primary key constraint of a parent file can be used to define a referential constraint. The parent key is one or more fields that exist in the file specified on the PRNFILE parameter.

Single values

***PRNFILE**
The access path of the parent file is used when the access path is either a primary key constraint or a unique constraint.

Other values (up to 120 repetitions)

name Specify the name of the field for the constraint key you are defining. Each field name must exist in the file specified on the PRNFILE parameter. You can specify a maximum of 120 (but no duplicate) field names to define the parent key, where:

- The fields can allow nulls.
- The field names are of the object type *NAME and are a maximum length of 10.
- The fields must be in ascending order.
- The fields must match the type and length attributes of the fields specified for the foreign key.
- The maximum number of bytes in a key is 32768 bytes (see the Database category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for more information on this limitation).

Top

Delete rule (DLTRULE)

Specifies the delete rule for a referential constraint between a parent file and dependent file. The **delete rule** restricts or defines the effect of deleting a record in both the parent file and the dependent file.

***NOACTION**
The no action delete rule is used. The delete rule is enforced at the end of the delete request. The following are attributes of the no action delete rule:

- Deleting a record in parent file is permitted (not restricted) if data for a non-null parent key does not match data for a foreign key.
- Deleting a record in a parent file is restricted (does not occur) if data for a non-null parent key matches data for a foreign key.

***RESTRICT**
The restrict delete rule is used. The delete rule is enforced at the beginning of the delete request. The following are attributes of the restrict delete rule:

- Deleting a record in a parent file is permitted if data for a non-null parent key does not match data for a foreign key.
- Deleting a record in a parent file is restricted if data for a non-null parent key matches data for a foreign key.

***CASCADE**
The cascade delete rule is used. Deleting a record in a parent file causes matching records in the dependent file to be deleted when data for a non-null parent key matches data for a foreign key.

***SETNULL**
The set null delete rule is used. Deleting a record in a parent file updates matching records in a dependent file if data for a non-null parent key matches data for a foreign key. If the matching foreign key field is null-capable, the value is set to null. If the matching foreign key field is not null-capable, the field is not updated.

Note: To use this rule, a minimum of one field in the foreign key access path must be null-capable.

***SETDFT**
The set default delete rule is used. The following are attributes of the set default delete rule:

- Deleting a record in the parent file updates matching records in the dependent file when a data for a non-null parent key matches data for a foreign key. The matching foreign key values are set to the default value as defined by the default.
- The default foreign key value must match the corresponding parent key value when there are no null-capable fields.

Top

Update rule (UPDRULE)

Specifies the update rule for a referential constraint between a parent file and dependent file. The **update rule** restricts or defines the effect of updating a record in both the parent file and the dependent file.

***NOACTION**
The no action update rule is used. The update rule is enforced at the end of the update request. The following are attributes of the no action update rule:

- Updating a record in parent file is permitted (not restricted) if data for a non-null parent key does not match data for a foreign key.
- Updating a record in a parent file is restricted (does not occur) if data for a non-null parent key matches data for a foreign key.

***RESTRICT**

The restrict update rule is used. The update rule is enforced at the beginning of the update request. The following are attributes of the restrict update rule:

- Updating a record in a parent file is permitted if data for a non-null parent key does not match data for a foreign key.
- Updating a record in a parent file is restricted if data for a non-null parent key matches data for a foreign key.

Top

Check constraint (CHKCST)

Specifies the check constraint expression that is being added.

character-value

Specify the check constraint expression. The expression has the same syntax as used for SQL check conditions. See the the DB2 for i5/OS SQL reference topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for more information on syntax rules.

Top

Examples

Example 1: Adding a Unique Constraint

```
ADDPFCST   FILE(MYLIB/LOCATIONS)  TYPE(*UNQCST)  KEY(REGION)
           CST(Personnel_by_REGION)
```

This command adds a unique constraint to the LOCATIONS file in the MYLIB library. The field that defines the access path is REGION. The name of the access path is Personnel_by_REGION.

Example 2: Adding a Referential Constraint

```
ADDPFCST   FILE(ADMN/PERSONNEL)  TYPE(*REFCST)  KEY(REGION)
           CST(1994Hires)  PRNFILE(MYLIB/LOCATIONS)
           PRNKEY(REGION)  DLTRULE(*CASCADE)
           UPDRULE(*RESTRICT)
```

This command adds a referential constraint to the PERSONNEL file in the ADMN library. The field that defines the access path is REGION, which is also the key for the parent file LOCATIONS in the MYLIB library. The name of the access path is 1994Hires. According to the delete rule of cascade, if a record in the LOCATIONS file is subsequently deleted, and that record matches a record in the PERSONNEL file, the record also will be deleted from the PERSONNEL file. According to the update rule of restrict, subsequent changes to the LOCATION file records defined in the constraint are restricted at the beginning of the update request.

Example 3: Adding a Check Constraint

```
ADDPFCST   FILE(PERSONNEL/SALARY)  TYPE(*CHKCST)
           CST(Upper_Salary_Limit)  CHKCST('EMPSAL <= 100000')
```

This command adds a check constraint to the SALARY file in the PERSONNEL library. The check constraint will ensure an employee's salary may be a maximum of 100,000.

Top

Error messages

***ESCAPE Messages**

CPF32B0
Constraint cannot be added to file &1.

CPF32B7
&3 constraint(s) added to file &1 but constraint(s) in error.

Top

Add Physical File Member (ADDPFM)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The Add Physical File Member (ADDPFM) command adds a named member to a physical file, which must already exist on the system. The maximum number of members that can be added to the file is specified for the **Maximum members (MAXMBRS)** parameter on the Create Physical File (CRTPF) command or the Change Physical File (CHGPF) command. To add other members to the file, use the ADDPFM command to specify each one.

The number of members that can be added to the physical file is limited to the number specified for the MAXMBRS parameter of the associated CRTPF command. Each member added has the same attributes as those defined in the physical file, its own set of data records, and its own access path, as specified in the data description specifications (DDS). The access path determines the order in which the records in that member are processed.

Restrictions:

- This command is conditionally threadsafe. In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA. This command is also not threadsafe and fails for Distributed Data Management (DDM) files of type *SNA, when SYSTEM(*RMT) or SYSTEM(*FILETYPE) is specified.

Note: An *EXCLRD lock is required on the file to add a member. Because this command adds a member to a file in a library, the library must not be locked (*SHRNUP or *EXCLRD in the Allocate Object command) by another job.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	Physical file	*Qualified object name*	Required, Positional 1
	Qualifier 1: Physical file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MBR	Member	*Name*	Required, Positional 2
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional
EXPDATE	Expiration date for member	*Date*, ***NONE**	Optional
SHARE	Share open data path	***NO**, *YES	Optional
SRCTYPE	Source type	*Simple name*, ***NONE**	Optional

Top

Physical file (FILE)

Specifies the physical file to which the member is to be added.

This is a required parameter.

Qualifier 1: Physical file

name Specify the name of the physical file.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the physical file. If no library is specified as the current library for the job, QGPL is used.

name Specify the library where the physical file is located.

Top

Member (MBR)

Specifies the file member to be added to the physical file. The file member name must be unique in the file.

If a DDM file is specified for the FILE parameter, and a member name is specified as part of the remote file name of the DDM file, the MBR name specified must match the member name in the remote file name in the DDM file.

This is a required parameter.

name Specify the name of the file member to be added.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the object.

***BLANK**
No text is specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Expiration date for member (EXPDATE)

Specifies the expiration date of the physical file member.

***NONE**
No expiration date is specified.

date Specify the date after which the file member cannot be used. The date must be enclosed in apostrophes if date separator characters are used in the value.

Top

Share open data path (SHARE)

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

***NO** The ODP is not shared with other programs in the routing step. A new ODP for the file is created and used every time a program opens the file.

***YES** The same ODP is shared with each program in the job that also specifies *YES when it opens the file.

Top

Source type (SRCTYPE)

Specifies the source type attribute to be assigned to the new member. A source type can only be specified if the physical file was created using the Create Source Physical File (CRTSRCPF) command, or using the Create Physical File (CRTPF) command with *SRC specified for the **File type (FILETYPE)** parameter.

***NONE**
No source type is specified.

character-value
Specify the source type for the file member. The source type is a character string of no more than 10 characters. The first character must be alphabetic (including the characters $, @, or #), and the remaining characters must be alphanumeric or an underline.

Top

Examples

```
ADDPFM   FILE(INVENTX)  MBR(MONDAYTX)
         TEXT('Monday''s Inventory Transactions')
```

This command adds a member named MONDAYTX to the physical file named INVENTX. The library list (*LIBL) is used to find the file because the FILE value is not qualified by a library name. The size of the member and the storage allocation values assigned to this member were specified in the CRTPF command that created the physical file. The text, *Monday's Inventory Transactions*, describes this member of the INVENTX file.

Top

Error messages

***ESCAPE Messages**

CPF3204
Cannot find object needed for file &1 in &2.

CPF7306
Member &1 not added to file &2 in &3.

Top

Add Physical File Trigger (ADDPFTRG)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The Add Physical File Trigger (ADDPFTRG) command adds a system trigger to a specified physical file. A **trigger** defines a program that is called with a delete, insert, update or read operation occurs for a file.

The trigger program can be specified to be called before or after a change operation occurs. The change operation can be an insert, update, delete or read operation through any interface. Change operations do not include clearing, initializing, moving, applying journal changes, removing journal changes, or changing end of data operations.

A maximum of 300 triggers can be added to one physical file. The trigger program to be called can be the same for each trigger or it can be a different program for each trigger.

An exclusive-no-read lock is held on the physical file when adding a trigger to that file. All logical files which are built over the physical file are also held with the exclusive-no-read lock.

Once a trigger is added to the physical file, all members of that specified file are affected by the trigger. When a change operation occurs on a member of the specified file, the trigger program is called. The trigger program is also called when a change operation occurs by way of either a dependent logical file or a Structured Query Language (SQL) view that is built over the physical file.

More information on the trigger program is in the Database category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Restrictions

- You must have read (*READ), object operational (*OBJOPR), and object alter (*OBJALTER) or object management (*OBJMGT) authorities to the physical file, execute (*EXECUTE) authority to the file library, and *EXECUTE authority to the trigger program and its library to use this command.
- You must have update (*UPD) and *OBJOPR authorities to the file if ALWREPCHG(*YES) has been specified.
- If the physical file or a dependent logical file or SQL view is opened in this or another job, a trigger cannot be added.
- While this command is running, neither the physical file nor any dependent logical files can be opened.
- The trigger program must be a program of object type *PGM. It cannot be an Integrated Language Environment (ILE) service program of object type *SRVPGM.
- This command is conditionally threadsafe. In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA.
- Read triggers may not be added to SQL materialized query tables.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	Physical file	*Qualified object name*	Required, Positional 1
	Qualifier 1: Physical file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TRGTIME	Trigger time	*BEFORE, *AFTER	Required, Positional 2
TRGEVENT	Trigger event	*INSERT, *DELETE, *UPDATE, *READ	Required, Positional 3
PGM	Program	*Qualified object name*	Required, Positional 4
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
RPLTRG	Replace trigger	***NO**, *YES	Optional
TRG	Trigger	*Character value*, ***GEN**	Optional
TRGLIB	Trigger library	*Name*, ***FILE**, *CURLIB	Optional
ALWREPCHG	Allow Repeated Change	***NO**, *YES	Optional
THDSAFE	Threadsafe	***UNKNOWN**, *NO, *YES	Optional
MLTTHDACN	Multithreaded job action	***SYSVAL**, *MSG, *NORUN, *RUN	Optional
TRGUPDCND	Trigger update condition	***ALWAYS**, *CHANGE	Optional

Top

Physical file (FILE)

Specifies the physical file to which this trigger program is added. The file must exist on the system.

This is a required parameter.

Qualifier 1: Physical file

name Specify the name of the file to which the trigger program is added.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Trigger time (TRGTIME)

Specifies the time when the trigger program is called.

This is a required parameter.

***BEFORE**
The trigger program is called before the change operation on the specified physical file.

***AFTER**
The trigger program is called after the change operation on the specified physical file.

Top

Trigger event (TRGEVENT)

Specifies the event (the change operation to the physical file) that calls the trigger program. Only one event can be specified for each command issued.

This is a required parameter.

***INSERT**
An insert operation calls the trigger program.

Note: If the physical file is not read and write capable, the *INSERT value cannot be specified.

***DELETE**
A delete operation calls the trigger program.

Notes:

- If the physical file is not read and delete capable, the *DELETE value cannot be specified.
- If the physical file has a referential constraint with a delete rule of CASCADE, the *DELETE value cannot be specified.

***UPDATE**
An update operation calls the trigger program.

Notes:

- If the physical file is not read and update capable, the *UPDATE value cannot be specified.
- If the physical file is a dependent file which has a foreign key with a delete rule of SET NULL or SET DEFAULT, the *UPDATE value cannot be specified.

***READ**
A read operation calls the trigger program.

Note: If the physical file is not read capable, the *READ value cannot be specified.

Top

Program (PGM)

Specifies the program that is called when the specified event occurs on the physical file. The program must exist on the system and be of object type *PGM.

This is a required parameter.

Qualifier 1: Program

name Specify the name of the program to be called when the specified event occurs on the specified physical file.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Note: The special values *LIBL and *CURLIB are the values of the job running when the trigger program is added.

Top

Replace trigger (RPLTRG)

Specifies whether an existing trigger is replaced by the trigger to be added when the triggers have the same trigger event and trigger time.

***NO** The existing trigger is not replaced.

***YES** The existing trigger is replaced. If *GEN was specified for the trigger name and the time and event match a single entry, the trigger will be replaced. If a trigger name was specified and it matches an existing entry, the trigger will be replaced. If a trigger with the specified trigger name does not exist, the new trigger is added to the physical file.

Top

Trigger (TRG)

Specifies the name of the trigger being added.

***GEN** The system generates a trigger name.

character-value
Specify the name of the trigger. The trigger name must be unique to the library. The trigger name is used to distinguish triggers with the same time and event values. You can specify a maximum of 128 characters without delimiters or 258 characters with quotation mark (″) delimiters.

Note: The case is preserved when lowercase characters are specified.

Top

Trigger library (TRGLIB)

Specifies the library for the trigger being added.

***FILE** The library for the file specified on the FILE parameter is used.

***CURLIB**
The current library for the job is used. If no library is specified as the current library for the job, the QGPL library is used.

Note: The special value *CURLIB is the value of the job running when the trigger program is added.

name Specify the name of the library to be used.

Top

Allow Repeated Change (ALWREPCHG)

Specifies whether repeated changes to a record within a trigger or any subsequent trigger(s) called while the trigger program is running are allowed.

***NO** Repeated changes to a record within a trigger or any subsequent trigger(s) called are not allowed.

***YES** Repeated changes to a record within a trigger or any subsequent trigger(s) called are allowed.

Top

Threadsafe (THDSAFE)

Specifies whether the trigger program is threadsafe. This is intended for documentation purposes only. It may be used in determining the MLTTHDACN value, but there is no direct relationship between the THDSAFE and MLTTHDACN keywords.

***UNKNOWN**
The threadsafe status of the trigger program is not known.

***NO** The trigger program is not threadsafe.

***YES** The trigger program is threadsafe.

Top

Multithreaded job action (MLTTHDACN)

Specifies the action to take when the trigger program is called in a multithreaded job. The THDSAFE attribute of the trigger program can be used in determining the action, however, there is no direct relationship between the THDSAFE and MLTTHDACN keywords.

***SYSVAL**
Use the QMLTTHDACN system value to determine the action to take.

***MSG** Run the trigger program in a multithreaded job, but send a diagnostic message.

***NORUN**
Do not run the trigger program in a multithreaded job. Send an escape message.

***RUN** Run the trigger program in a multithreaded job.

If you do use the THDSAFE value to determine the value for MLTTHDACN, please read the following recommendations:

- If the THDSAFE value is *NO, MLTTHDACN should be set to *NORUN.
- If the THDSAFE value is *UNKNOWN, MLTTHDACN should be set to *SYSVAL.
- If the THDSAFE value is *YES, MLTTHDACN should be set to *RUN.

Top

Trigger update condition (TRGUPDCND)

Specifies the condition under which an update event calls the trigger program.

Note: This parameter applies only when *UPDATE is specified for the **Trigger event (TRGEVENT)** parameter.

***ALWAYS**
The trigger program is called whenever a record is updated, whether or not a value changes.

***CHANGE**
The trigger program is called only when a record is updated and a value is changed.

Top

Examples

Example 1: Adding a Trigger for an Insert Event

```
ADDPFTRG   FILE(EMP)  TRGTIME(*AFTER)  TRGEVENT(*INSERT)
           PGM(LIB2/INSTRG)
```

This command adds a trigger with trigger program INSTRG in library LIB2 to the physical file named EMP. When an insert operation occurs on the EMP file, the program INSTRG is called after the insert operation. The library list (*LIBL) is used to find the file because the FILE value is not qualified by a library name.

Example 2: Setting Multiple Trigger Events to Call One Trigger Program

```
ADDPFTRG   FILE(EMP)  TRGTIME(*AFTER)  TRGEVENT(*INSERT)
           PGM(LIB2/INSTRG)
ADDPFTRG   FILE(EMP)  TRGTIME(*AFTER)  TRGEVENT(*UPDATE)
           PGM(LIB2/INSTRG)
```

These two commands add triggers to call the trigger program INSTRG in library LIB2 when an insert or update operation occurs on the EMP file.

Example 3: Adding a Trigger Only When an Update Event Changes Values

```
ADDPFTRG   FILE(EMP)  TRGTIME(*BEFORE)  TRGEVENT(*UPDATE)
           PGM(LIB2/UPDTRG)  TRGUPDCND(*CHANGE)
```

The trigger program UPDTRG in library LIB2 is called before a value for a field of a record in the EMP file changes during an update.

Example 4: Replacing an Existing Trigger

```
ADDPFTRG   FILE(EMP)  TRGTIME(*BEFORE)  TRGEVENT(*UPDATE)
           PGM(LIB2/NEWPGM)  RPLTRG(*YES)  TRGUPDCND(*CHANGE)
```

The trigger program NEWPGM being added to the file EMP has the same trigger time (*BEFORE) and trigger event (*UPDATE) as the trigger program UPDTRG that was added in Example 3. Therefore, the added trigger program NEWPGM replaces the existing trigger program UPDTRG.

Example 5: Replacing a Trigger with a Trigger for a Different Update Condition

```
ADDPFTRG   FILE(EMP)  TRGTIME(*BEFORE)  TRGEVENT(*UPDATE)
           PGM(LIB2/NEWPGM)  RPLTRG(*YES)  TRGUPDCND(*ALWAYS)
```

The trigger added in Example 4 that calls the trigger program NEWPGM only is the values are changed, is replace by a trigger that always calls the trigger program NEWPGM regardless of the values.

Top

Error messages

***ESCAPE Messages**

CPF32C6
Trigger operation not successful.

Top

Add Phy File Variable Len Mbr (ADDPFVLM)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Physical File Variable-Length Member (ADDPFVLM) command is used to add a physical file variable length member that can be used by Pascal programs.

Error messages for ADDPFVLM

None

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File	*Qualified object name*	Required, Positional 1
	Qualifier 1: File	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MBR	Member	*Name*	Required, Positional 2
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional
EXPDATE	Expiration date	*Date*, ***NONE**	Optional, Positional 3
SHARE	Share open data path	***NO**, *YES	Optional

Top

File (FILE)

Specifies the name of the physical file that will contain the variable-length members.

file-name
Enter the name of the physical file that will contain the variable-length members.

***LIBL** The system searches the library list to find the library where the physical file is located.

***CURLIB**
The name of the current library is used when ADDPFVLM is processed. If you have not specified the current library, QGPL is used.

library-name
Enter the name of the library in which the physical file is located.

Top

Member (MBR)

Specifies the name of the member that will contain the variable length records.

file-member-name
Enter the name of the member that will contain the variable-length records.

Top

Text 'description' (TEXT)

Specifies the text associated with the created file.

***BLANK**
No text to accompany file.

description
Enter the descriptive text no longer than 50 characters.

Top

Expiration date (EXPDATE)

Specifies the expiration date of the file.

***NONE**
No expiration date is specified.

expiration-date
Enter the expiration date for the file.

Top

Share open data path (SHARE)

Specifies the Share Open Data Path option.

***NO** File cannot be opened using the Share Open Data Path option.

***YES** When file is open, it will share the open data path if available.

Top

Examples

None

Top

Error messages

None

Top

Add Prefix to DLFM (ADDPFXDLFM)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Prefix to DLFM (ADDPFXDLFM) command registers a prefix with the DataLink File Manager. This prefix can then be used as a path for storing linked files.

Restrictions:

- To use this command, you must have input/output system configuration (*IOSYSCFG) special authority.
- Only prefixes in the root file system can be specified. The Display Mounted File System Information (DSPMFSINF) command can be used to identify the file system in which a prefix resides.

Top

Parameters

Keyword	Description	Choices	Notes
PREFIX	Prefix	Values (up to 300 repetitions): *Element list*	Optional, Positional 1
	Element 1: Name	*Character value*	
SRCFILE	Source file	*Qualified object name*	Optional, Positional 2
	Qualifier 1: Source file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*	Optional

Top

Prefix (PREFIX)

Specifies one or more prefixes to be registered with the DataLink File Manager. Up to 300 prefixes can be specified.

character-value
Specify a prefix to be registered.

Top

Source file (SRCFILE)

Specifies the source file that will be used to provide the prefixes to be registered.

Note: If a value is specified for this parameter, you must also specify a value for the **Source member (SRCMBR)** parameter.

Qualifier 1: Source file

name Specify the name of the source file.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the file. If no library is specified as the current library, QGPL is used.

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the source file member that contains the prefixes to be registered. Each row in the source member must contain a prefix.

Note: If a value is specified for this parameter, you must also specify a value for the **Source file (SRCFILE)** parameter.

name Specify the name of the source file member..

Top

Examples

Registering a Datalink File Manager Prefix

```
ADDPFXDLFM   PREFIX('/files/datalinks')
```

This command registers */files/datalinks* as a valid prefix for linked files.

Top

Error messages

***ESCAPE Messages**

CPF3168
DataLink File Manager (DLFM) command failed.

Top

Add Program (ADDPGM)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Program (ADDPGM) command adds one to twenty programs to the group of programs currently being debugged. When included in debug mode, the specified programs can have breakpoints and traces added to them for controlling and tracing their processing. The values of the programs' variables can also be displayed and changed.

When debugging one job from another job, debugging affects the running of the programs in the job being debugged, but not the job doing the debugging. You may run programs in a job doing the debugging, however, the programs will not be debugged.

Restrictions:

- You can use this command only in debug mode. To start debug mode, refer to the Start Debug (STRDBG) command.
- You cannot use this command if you are servicing another job, and that job is on a job queue, or is being held, suspended, or ended.
- No more than 20 programs can be debugged at the same time.
- Two or more programs with the same name cannot be debugged at the same time.
- This command cannot be used to add bound programs.
- You must have either *CHANGE authority to the program, or *USE authority to the program and *SERVICE special authority.

Top

Parameters

Keyword	Description	Choices	Notes
PGM	Program	Values (up to 20 repetitions): *Qualified object name*	Required, Positional 1
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DFTPGM	Default program	*Name*, ***SAME**, *NONE	Optional, Positional 2

Top

Program (PGM)

Specifies one or more programs to be debugged. The number of programs specified here depends on how many programs are already being debugged; 20 is the maximum number of programs that can be debugged at the same time. You cannot debug two programs that have the same name at the same time.

This is a required parameter.

Qualifier 1: Program

name Specify the name of the program to be debugged.

Qualifier 2: Library

***LIBL** The library list is used to locate the program to debug.

***CURLIB**
The current library for the job is used to locate the program to debug. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the program is located.

Top

Default program (DFTPGM)

Specifies the program to use as the default program during debug mode. The program specified here is used as the default program for any of the other debug commands for which *DFTPGM was supplied for the **Program (PGM)** parameter. That is, if a default program was previously specified, this parameter can change it.

***SAME**
The program currently specified as the default program, if any, does not change.

***NONE**
No program is specified as the default program; if a program was specified as a default program, it is no longer the default program. If the job has no default program, *DFTPGM cannot be specified for the **Program (PGM)** parameter of any other debug commands.

name Specify the name of the program to use as the default program during debug mode. The same name must also be specified for the PGM parameter of this command or have been specified on the Start Debug (STRDBG) command or on a previous Add Program (ADDPGM) command.

Top

Examples

```
ADDPGM   PGM(QGPL/MYPROG)
```

This command adds the program MYPROG, located in the QGPL library, to the current debug mode. Breakpoints and traces can be put in MYPROG, and its variables can be displayed and changed by other debug commands. Because DFTPGM was not specified, the default program is not changed.

Top

Error messages

***ESCAPE Messages**

CPF1999
Errors occurred on command.

Top

Add Prestart Job Entry (ADDPJE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Prestart Job Entry (ADDPJE) command adds a prestart job entry to the specified subsystem description. The entry identifies prestart jobs that may be started when the subsystem is started or when the Start Prestart Jobs (STRPJ) command is entered.

Restrictions:

1. To use this command, you must have:
   - object operational (*OBJOPR), object management (*OBJMGT), and read (*READ) authority to the specified subsystem description and execute (*EXECUTE) authority to the library containing that subsystem description.
   - object operational (*OBJOPR) and read (*READ) authority to the job description and execute (*EXECUTE) authority to the library containing that job description.
   - use (*USE) authority to the user profile.
2. Only a user with all object (*ALLOBJ) special authority is allowed to add an entry for which the job description does not exist.

Top

Parameters

Keyword	Description	Choices	Notes
SBSD	Subsystem description	*Qualified object name*	Required, Positional 1
	Qualifier 1: Subsystem description	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
PGM	Program	*Qualified object name*	Required, Positional 2
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
USER	User profile	*Name*, **QUSER**	Optional
STRJOBS	Start jobs	***YES**, *NO	Optional
INLJOBS	Initial number of jobs	1-9999, **3**	Optional
THRESHOLD	Threshold	1-9999, **2**	Optional
ADLJOBS	Additional number of jobs	0-999, **2**	Optional
MAXJOBS	Maximum number of jobs	1-9999, ***NOMAX**	Optional
JOB	Job name	*Name*, ***PGM**	Optional
JOBD	Job description	Single values: ***USRPRF**, *SBSD Other values: *Qualified object name*	Optional
	Qualifier 1: Job description	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MAXUSE	Maximum number of uses	1-1000, **200**, *NOMAX	Optional
WAIT	Wait for job	***YES**, *NO	Optional
POOLID	Pool identifier	1-10, **1**	Optional

Keyword	Description	Choices	Notes
CLS	Class	*Element list*	Optional
	Element 1: Class	Single values: ***SBSD** Other values: *Qualified object name*	
	Qualifier 1: Class	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 2: Number of jobs to use class	0-32766, ***CALC**, *MAXJOBS	
	Element 3: Class	Single values: ***NONE**, *SBSD Other values: *Qualified object name*	
	Qualifier 1: Class	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 4: Number of jobs to use class	0-32766, ***CALC**, *MAXJOBS	
THDRSCAFN	Thread resources affinity	Single values: ***SYSVAL** Other values: *Element list*	Optional
	Element 1: Group	*NOGROUP, *GROUP	
	Element 2: Level	*NORMAL, *HIGH	
RSCAFNGRP	Resources affinity group	***NO**, *YES	Optional

Top

Subsystem description (SBSD)

Specifies the name and library of the subsystem description to which the prestart job entry is being added. If no library qualifier is given, *LIBL is used to find the subsystem description.

This is a required parameter.

Qualifier 1: Subsystem description

name Specify the name of the subsystem description to which the prestart job entry is being added.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the subsystem description's library to which the prestart job entry is being added.

Top

Program (PGM)

Specifies the name and library of the program run by the prestart job. This program name is used to match an incoming request with an available prestart job. If the program does not exist when the entry is added, a library qualifier must be specified because the qualified name is kept in the subsystem description.

Note: Two entries with the same program name can exist in a single subsystem description, but they must have different library names.

This is a required parameter.

Qualifier 1: Program

name Specify the name of the program run by the prestart job.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the library of the program to be run by the prestart job.

Top

User profile (USER)

Specifies the name of the user profile under which the prestart job is initiated. In addition, the current user profile of the prestart job is set to this user whenever the job waits for a request to handle.

Note: When a prestart job is given a request to handle, the current user profile of the job is updated. Refer to the Work Management guide for information on how this profile is determined. This change in current user profile is for authority checking only. None of the other attributes of the user profile, such as the current library (CURLIB) or the initial program to call (INLPGM), are given to the prestart job.

QUSER
The IBM-supplied QUSER user profile is used.

name Specify the name of the user profile used for the prestart job.

Top

Start jobs (STRJOBS)

Specifies whether the prestart jobs should be started at the time the subsystem is started.

Note: Changing this value when the subsystem is active will produce no effect until the subsystem is ended and started again.

***YES** The prestart jobs are started at the time the subsystem is started.

***NO** The prestart jobs are not started at the time the subsystem is started. The Start Prestart Jobs (STRPJ) command must be used to start these prestart jobs.

Top

Initial number of jobs (INLJOBS)

Specifies the initial number of prestart jobs that are started when the subsystem specified on the **Subsystem description (SBSD)** parameter is started.

Notes:

1. The value of this parameter must be less than or equal to the value of the **Maximum number of jobs (MAXJOBS)** parameter.

2. The value of this parameter must be greater than or equal to the value of the **Threshold (THRESHOLD)** parameter.

3 Three prestart jobs are started when the subsystem is started.

1-9999 Specify the number of prestart jobs that are started when the subsystem is started. Valid values range from 1 through 9999.

Top

Threshold (THRESHOLD)

Specifies when additional prestart jobs are started. When the pool of available jobs (jobs available to service requests) is reduced below this number, more jobs (specified on the **Additional number of jobs (ADLJOBS)** parameter) are started and added to the available pool.

Note: The value of this parameter must be less than or equal to the value specified on the **Initial number of jobs (INLJOBS)** parameter.

2 When one prestart job is available, the number of jobs specified on the **Additional number of jobs (ADLJOBS)** parameter are started.

1-9999 Specify the minimum number of prestart jobs that must be available before additional prestart jobs are started. Valid values range from 1 through 9999.

Top

Additional number of jobs (ADLJOBS)

Specifies the additional number of prestart jobs that are started when the number of prestart jobs drops below the value specified on the **Threshold (THRESHOLD)** parameter.

Note: The value specified on this parameter must be less than the value specified on the **Maximum number of jobs (MAXJOBS)** parameter.

2 Two additional prestart jobs are started.

0-999 Specify the number of additional prestart jobs to start. Valid values range from 0 through 999.

Top

Maximum number of jobs (MAXJOBS)

Specifies the maximum number of prestart jobs that can be active at the same time for this prestart job entry. This value includes prestart jobs that are servicing a procedure start request, prestart jobs that are waiting to service a procedure start request, and prestart jobs that are being started as a result of reaching the value specified on the **Threshold (THRESHOLD)** parameter.

Notes:

1. The value of this parameter must be greater than or equal to the value specified on the **Initial number of jobs (INLJOBS)** parameter.
2. The value of this parameter must be greater than the value specified on the **Additional number of jobs (ADLJOBS)** parameter.

***NOMAX**

There is no maximum number of prestart jobs that can be active at the same time.

1-9999 Specify the maximum number of prestart jobs that can be active at the same time. Valid values range from 1 through 9999.

Top

Job name (JOB)

Specifies the name of the prestart job that is started. Prestart jobs are automatically started when the subsystem, specified on the **Subsystem description (SBSD)** parameter, is started if STRJOBS(*YES) is specified.

***PGM** The job name is the same as the program name specified on the **Program (PGM)** parameter.

job-name
Specify the name of the prestart job.

Top

Job description (JOBD)

Specifies the qualified name of the job description used for the prestart job. If the job description does not exist when the entry is added, a library qualifier must be specified because the qualified job description name is kept in the subsystem description.

Note: Only a user with all object (*ALLOBJ) special authority is allowed to add or change an entry for which the job description does not exist.

Single values

***USRPRF**
The job description name entered in the user profile specified on the **User profile (USER)** parameter is used.

***SBSD**
The job description having the same name as the subsystem description named on the **Subsystem description (SBSD)** parameter is used.

Qualifier 1: Job description

name Specify the name of the job description being used for this prestart job.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the job description's library name.

Top

Maximum number of uses (MAXUSE)

Specifies the maximum number of requests that can be handled by each prestart job in the pool before the job is ended.

200 A prestart job for this entry can service a maximum of 200 requests before it is ended and another prestart job is started to take its place.

***NOMAX**
There is no maximum number of program start requests that a prestart job can handle. The job is not ended by the subsystem.

Note: Avoid having jobs exist for long periods of time since this can cause the job log to exceed the maximum size. Also avoid situations in which jobs can create more than the maximum number of spooled files, or can exceed the maximum processing unit time or the maximum temporary storage allocation.

1-1000 Specify the maximum number of requests that a prestart job can handle before it is ended. Valid values range from 1 through 1000.

Top

Wait for job (WAIT)

Specifies whether program start requests wait for a prestart job to become available or are rejected if a prestart job is not immediately available when the procedure start request is received.

Note: Refer to the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ to determine the time-out considerations for the communications type being used.

***YES** Program start requests wait until there is an available prestart job or a prestart job is started to service the request.

***NO** Program start requests are rejected if there is no prestart job immediately available when the procedure start request is received.

Top

Pool identifier (POOLID)

Specifies the subsystem pool identifier that the prestart jobs will run in.

1 The prestart jobs run in pool 1.

1-10 Specify the subsystem pool identifier in which the prestart jobs run. Valid values range from 1 through 10.

Top

Class (CLS)

Specifies the name and library of the classes that the prestart jobs run under and how many prestart jobs run under each class. Jobs start by using the first class specified until the number of jobs specified for the first class is reached. After the number of jobs specified for the first class is reached, jobs are started under the second class. If the class does not exist when the prestart job entry is added, a library qualifier must be specified because the qualified class name is kept in the subsystem description.

Note: Two classes can be specified on this parameter.

Element 1: Class

Single values

***SBSD**
The class having the same name as the subsystem description, specified on the **Subsystem description (SBSD)** parameter, is used for prestart jobs.

Qualifier 1: Class

name Specify the name of the class being used for prestart jobs.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the library name of the class being used for prestart jobs.

Element 2: Number of jobs to use class

***CALC**
The system calculates how many prestart jobs use this class. If only one class is specified and *CALC is specified, all of the jobs use that class. If two classes are specified and *CALC is specified for both, the first class is the value specified on the **Maximum number of jobs (MAXJOBS)** parameter divided by two, and the second class is the value of the MAXJOBS parameter minus the value calculated for the first class. If a specific number of jobs is specified for either class and *CALC is specified for the other class, the system calculates the difference between MAXJOBS and the specific number of jobs for the *CALC designation.

***MAXJOBS**
All of the prestart jobs use the specified class.

0-32766
Specify the number of jobs that use this class. The sum of the values specified for both classes must total the value specified on the MAXJOBS parameter.

Element 3: Class

Single values

***NONE**
Specify this value if only one class is used for this prestart job entry.

***SBSD**
The class having the same name as the subsystem description, specified on the **Subsystem description (SBSD)** parameter, is used for prestart jobs.

Qualifier 1: Class

name Specify the name of the class being used for prestart jobs.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the library name of the class being used for prestart jobs.

Element 4: Number of jobs to use class

***CALC**
The system calculates how many prestart jobs use this class. If only one class is specified and *CALC is specified, all of the jobs use that class. If two classes are specified and *CALC is specified for both, the first class is the value specified on the **Maximum number of jobs (MAXJOBS)** parameter divided by two, and the second class is the value of the MAXJOBS parameter minus the value calculated for the first class. If a specific number of jobs is specified for either class and *CALC is specified for the other class, the system calculates the difference between MAXJOBS and the specific number of jobs for the *CALC designation.

***MAXJOBS**
All of the prestart jobs use the specified class.

0-32766
Specify the number of jobs that use this class. The sum of the values specified for both classes must total the value specified on the MAXJOBS parameter.

Top

Thread resources affinity (THDRSCAFN)

Specifies the affinity of threads to system resources.

Single values

***SYSVAL**
When the prestart job is started, the thread resources affinity value from the QTHDRSCAFN system value will be used.

Element 1: Group

***NOGROUP**
Prestart jobs will have affinity to a group of processors and memory. Secondary threads running under the job will not necessarily have affinity to the same group of processors and memory.

***GROUP**
Prestart jobs will have affinity to a group of processors and memory. Secondary threads running under the job will all have affinity to the same group of processors and memory as the initial thread.

Element 2: Level

***NORMAL**
A thread will use any processor or memory in the system if the resources it has affinity to are not readily available.

***HIGH**
A thread will only use the resources it has affinity to, and will wait until they become available if necessary.

Top

Resources affinity group (RSCAFNGRP)

Specifies whether or not prestart jobs using this entry will be grouped together having affinity to the same system resources (processors and memory). A value of *YES for this parameter will take precedence over the QTHDRSCAFN system value when set to *NOGROUP.

***NO** Prestart jobs that use this entry will not be grouped together.

***YES** Prestart jobs that use this entry will be grouped together such that they will have affinity to the same system resources. Jobs that share data in memory may perform better if they have affinity to the same resources.

Top

Examples

Example 1: Specifying Additional Prestart Jobs

```
ADDPJE   SBSD(QGPL/PJSBS)  PGM(QGPL/PGM1)  INLJOBS(15)
         THRESHOLD(5)  ADLJOBS(10)  WAIT(*NO)
```

This command adds a prestart job entry for the PGM1 program in the QGPL library to the PJSBS subsystem description contained in the QGPL library. The entry specifies that 15 prestart jobs (program PGM1 in the QGPL library) are started when subsystem PJSBS in the QGPL library is started. When the pool of available prestart jobs is reduced to four (because the prestart jobs are servicing requests specified for program PGM1 in the QGPL library), ten additional jobs are started. If no prestart jobs are available for this entry when a request is received, the request is rejected.

Example 2: Specifying Maximum Number of Prestart Jobs

```
ADDPJE   SBSD(QGPL/PJSBS)  PGM(QGPL/PGM2)  USER(PJUSER)
         MAXJOBS(100) CLS(QGPL/CLS1 75 QGPL/CLS2 *CALC)
         MAXUSE(50)
```

This command adds a prestart job entry for the PGM2 program in the QGPL library to the PJSBS subsystem description contained in the QGPL library. The entry specifies that the prestart job for this entry runs under the PJUSER user profile. The maximum number of prestart jobs that can be active at the same time for this entry is 100. Each prestart job in the pool can handle 50 requests before the job is ended. If 100 prestart jobs are active at the same time for this entry, 75 of them would use CLS1 in the QGPL library, and 25 of them would use CLS2 in the QGPL library. If 50 prestart jobs are active at the same time for this entry, all 50 of them would use class CLS1 in the QGPL library.

Top

Error messages

***ESCAPE Messages**

CPF1691
Active subsystem description may or may not have changed.

CPF1697
Subsystem description &1 not changed.

Top

Add Problem Action Entry (ADDPRBACNE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Problem Action Entry (ADDPRBACNE) command adds an entry to the specified problem filter. This entry describes the actions to take for a problem entry. A problem entry is assigned to the specified group by a selection entry in the specified problem filter.

Top

Parameters

Keyword	Description	Choices	Notes
FILTER	Filter	*Qualified object name*	Required, Positional 1
	Qualifier 1: Filter	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
GROUP	Group	*Name*	Required, Positional 2
ASNUSER	User assigned	*Simple name*, ***NOCHG**, *NONE	Optional
SNDDTAQ	Send to data queue	Single values: ***NONE** Other values (up to 5 repetitions): *Element list*	Optional
	Element 1: Data queue	*Qualified object name*	
	Qualifier 1: Data queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 2: Data queue key	*Character value*, ***NONE**, X''	

Top

Filter (FILTER)

Specifies the name of the filter.

This is a required parameter.

Qualifier 1: Filter

name Specify the name of the filter.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the filter. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the filter is located.

Top

Group (GROUP)

Specifies the group for which the actions are applied. The group name is assigned from selection criteria from a selection entry in the filter. Selection entries are added to the filter with the ADDPRBSLTE command.

This is a required parameter.

Top

User assigned (ASNUSER)

Specifies the user assigned to the problem log entry.

***NOCHG**
No new value is assigned to the problem log entry.

***NONE**
No user is assigned to the problem log entry.

assigned-user
Specify a user name.

Top

Send to data queue (SNDDTAQ)

Specifies the data queue for the problem notification record. Keyed data queues are supported.

Single values

***NONE**
No data queue is used.

Other values (up to 5 repetitions)

Element 1: Data queue

Qualifier 1: Data queue

name Specify the name of the data queue.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the data queue. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the data queue is located.

Element 2: Data queue key

***NONE**
No key is used on the data queue.

character-value
Specify the data queue key.

NOTES:

1. If an exact match is not found for the group name when a filter is applied, the default action entry is used to assign actions. The default action entry is automatically added to the filter when it is created. The default values are ASNUSER(*NONE) and SNDDTAQ(*NONE).
2. A keyed data queue is a queue with a key assigned to each entry on the queue. When retrieving entries, a key can be specified and the entries with that key are retrieved on a FIFO order. The key that is specified on the *SNDDTAQ parameter is assigned to the problem notification record when placed on a keyed data queue.

An 80-byte record is enqueued on the data queue specified by the user. This record is received when the QRCVDTAQ program is called. The data queue does not have to be used solely for problems; alerts and problems can share the same data queue.

If a key is specified, it is used when enqueueing the record on the queue. If the data queue is non-keyed, the record is enqueued without a key.

Note: The time stamp used is the system standard time stamp. This time is already stored in the problem record.

Record Format

Position	Type	Value	Description
1-10	CHAR	*PRBFTR	Problem filtering notification
11-11	CHAR	Function	Function performed 1 - Problem created 2 - Problem changed 3 - Problem deleted
12-19	CHAR	Function TOD	TOD time stamp for function
20-29	CHAR	Group	Group problem was filtered into
30-39	CHAR	Problem ID	Problem ID number
40-59	CHAR	Origin System	System where problem originated
60-60	CHAR	Last Event	Last event committed into the history log (see note)
61-68	CHAR	Event TOD	TOD time stamp for Last Event
69-80	CHAR	Reserved	Reserved for future use

Note: Valid Last Event values are the following:

```
'01'X   Problem entry opened
'02'X   Request received
'03'X   Opened by Alert
'04'X   Reported by HMC
'05'X   Reported by Service Partition
'06'X   Reported by current i5/OS Partition
'10'X   Problem analyzed
'11'X   Verification test ran
'12'X   Recovery procedure ran
'20'X   Prepared to report
'21'X   Service request sent
'22'X   Problem answered
'23'X   Response sent
'24'X   Reported by voice
'25'X   Fixes transmitted
```

```
'30'X   Fix verified
'41'X   Analyzed remotely
'42'X   Remote verification ran
'43'X   Remote recovery ran
'50'X   Alert created
'51'X   APAR created
'52'X   APAR data saved
'54'X   APAR data restored
'55'X   APAR data deleted
'60'X   Problem changed by Change Problem (CHGPRB) command
'61'X   Problem deleted by Delete Problem (DLTPRB) command
'99'X   Problem entry closed
```

Top

Examples

```
ADDPRBACNE   FILTER(MYLIB/MYFILTER)  GROUP(IOWA)
             ASNUSER(SYSOPR))  SNDDTAQ(*LIBL/PROBDTAQ)
```

The actions defined for group IOWA are: enqueue the problem on data queue PROBDTAQ; and assign the problem to user SYSOPR.

Top

Error messages

***ESCAPE Messages**

CPF2150
Object information function failed.

CPF2151
Operation failed for &2 in &1 type *&3.

CPF7A82
Error occurred while applying the problem filter.

CPF812F
Filter damaged.

CPF91DB
Group &4 already exists.

CPF91DE
Filter &1/&2 at maximum size.

CPF91EB
Filter type &3 not correct for this operation.

CPF91EC
Internal processing error occurred.

CPF91E8
Internal processing error occurred.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

Top

Add Problem Selection Entry (ADDPRBSLTE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Problem Selection Entry (ADDPRBSLTE) command allows you to define selection criteria that categorize a group of problem log entries. You can add a problem log selection entry to a problem log filter that was created using the Create Filter (CRTFTR) command.

Top

Parameters

Keyword	Description	Choices	Notes
FILTER	Filter	*Qualified object name*	Required, Positional 1
	Qualifier 1: Filter	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SELECT	Selection data	Single values: *ANY Other values (up to 10 repetitions): *Element list*	Required, Positional 2
	Element 1: Relationship	*IF, *AND, *OR	
	Element 2: Attribute	*EVENT, *ORIGIN, *ORGNETID, *ORGCPNAM, *RCVNETID, *RCVCPNAM, *PROBTYPE, *SEV, *MSGID, *ORGHDW, *RSCHDW, *RSCSFW	
	Element 3: Relational operator	*EQ, *GT, *LT, *NE, *GE, *LE, *CT	
	Element 4: Value	*Character value*	
SEQNBR	Sequence number	1-9999, ***GEN**	Optional
GROUP	Group	*Name*, ***DEFAULT**	Optional

Top

Filter (FILTER)

Specifies the name of the filter.

This is a required parameter.

Qualifier 1: Filter

name Specify the name of the filter.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the filter. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the filter is located.

Selection data (SELECT)

Specifies that a problem log entry is selected or not selected based on whether information in the problem log entry satisfies a specified relationship.

You can specify a single value (*ANY) or all four elements that define a relationship. When you specify the four elements, the attribute and attribute value are compared for the relationship specified by the relational operator.

This is a required parameter.

Single values

***ANY** Any problem log entry is selected.

Other values (up to 10 repetitions)

Element 1: Relationship

***IF** The specified relationship must be satisfied for a problem log entry to be selected.

***AND** The specified relationship must be satisfied in addition to the *IF relationship for a problem log entry to be selected.

***OR** The specified relationship must be satisfied in addition to or instead of the *IF relationship for a problem log entry to be selected.

Element 2: Attribute

***EVENT**
The filter is applied when the problem log entry is created (a value of 1), changed (a value of 2), or deleted (a value of 3). If the entry has been created and is changed before being committed, use the value of 1.

***ORIGIN**
The problem log entry was locally generated (a value of L) or was received from another system (a value of R).

***ORGNETID**
The network identifier (ID) of the system in which the problem log entry originated is specified. This information is displayed using the Work with Problems (WRKPRB) command which shows the details for a specific problem. Specify the value in the following form:

```
'nnnnnnnnnn'
```

***ORGCPNAM**
The control point name of the system in which the problem log entry originated is specified. This information is displayed using the Work with Problems (WRKPRB) command which shows the details for a specific problem. Specify the value in the following form:

```
'cccccccccc'
```

***RCVNETID**
The network identifier of the remote system from which the problem log entry was received is specified. This information is displayed using the Work with Problems (WRKPRB) command which shows the details for a specific problem. Specify the value on the following form:

```
'nnnnnnnnnn'
```

***RCVCPNAM**
This attribute specifies the Remote System Control Point name in which the problem log entry received from. This information is displayed using the Work with Problems (WRKPRB) command and shows the details for a specific problem. The value specified for this attribute should be of the following form:

```
'ccccccccc'
```

***PROBTYPE**
The type of problem entry created. Possible problems are machine-detected (a value of 1), user-detected (a value of 2), PTF order (a value of 3), application-detected (a value of 4), Client machine-detected (a value of 5), or Client user-detected (a value of 6).

Note: User-Detected Remote Hardware problems are grouped with number 2 User-Detected problems.

***SEV** The severity of the problem log entry created. Possible choices are high (a value of 1), medium (a value of 2), low (a value of 3), none (a value of 4), or not assigned (a value of 5).

Note: Problems do not have a severity level when locally created.

***MSGID**
The message ID found in the problem log entry. This is usually an i5/OS message ID.

***ORGHDW**
The origin hardware resource information in the problem log entry. This information is displayed using the Work with Problems (WRKPRB) command and shows the details for a specific problem. Specify the value in the following form:

```
'tttt mmm ss-ssssssss'
'tttt mmm ss-sssss'
'tttt mmm sssssss'
'tttt mmm sssss'
```

where tttt is the machine type, mmm is the model number and sssssssss is the serial number. Use this exact format to match a particular hardware resource exactly, or use a part of the hardware value with the Contains (*CT) relation to provide a partial match.

***RSCHDW**
The failing hardware resource information in the problem log entry. This information is displayed using the Work with Problems (WRKPRB) command and shows the details for a specific problem. Specify the value in the following form:

```
'tttt mmm ss-ssssssss'
'tttt mmm ss-sssss'
'tttt mmm sssssss'
'tttt mmm sssss'
```

where tttt is the machine type, mmm is the model number and sssssssss is the serial number. Use this exact format to match a particular hardware resource exactly, or use a part of the hardware value with the Contains (*CT) relation to provide a partial match.

***RSCSFW**
The failing software resource information in the problem log entry. This information is displayed using the Work with Problems (WRKPRB) command and shows the details for a specific problem. Specify the value in the following form:

```
'ppppppp vv rr mm'
```

where ppppppp is the licensed program ID, vv is the version number, rr is the release number, and mm is the modification level. Use this exact format to match a particular software resource exactly, or use a part of the software value with the Contains (*CT) relation to provide a partial match.

Element 3: Relational operator

The value specified for element 2 (**Attribute**) must have the following relationship to value specfied for element 4 (**Value**) of this parameter.

***EQ** Equal to

***GT** Greater than

***LT** Less than

***NE** Not equal to

***GE** Greater than or equal to

***LE** Less than or equal to

***CT** Contains

Element 4: Value

attribute-value
Specify a value of up to 30 characters to compare with the contents of the attribute specified for element 2 of this parameter. The value must be specified in character format and must be enclosed in apostrophes if it contains blanks or special characters. If a CL variable is specified for the value, it must be a character variable.

generic-attribute-value
Specify the generic attribute value. A generic value is a character string of one or more characters followed by an asterisk (*); for example, ABC*. If a generic name is specified, all values that begin with the generic value are selected. If an asterisk is not included with the generic (prefix) value, the system assumes it to be the complete value.

Top

Sequence number (SEQNBR)

Specifies the sequence number of the problem log selection entry. Selection entries in a filter are numbered by sequence number. When a filter is applied, the selection entries are tested in order of ascending sequence number.

***GEN** The system generates the sequence number.

1-9999 Specify a sequence number.

Top

Group (GROUP)

Specifies the group to which a problem log entry is assigned if it matches the criteria specified on the SELECT parameter.

***DEFAULT**
The problem log entry is assigned to the default group.

name Specify the name of the group.

Examples

Example 1: Adding a Selection Entry

```
ADDPRBSLTE   FILTER(PROBLIB/PROBFILTER)
             SELECT((*IF *EVENT *EQ 1) (*AND *SEV *EQ 1))
             SEQNBR(*GEN)  GROUP(HIGHPROB)
```

This command adds an entry to the filter PROBFILTER in library PROBLIB. Any problems that have been created and are of severity 1 are assigned to group HIGHPROB.

Example 2: Assigning Entries by Origin System Network ID

```
ADDPRBSLTE   FILTER(PROBLIB/PROBFILTER)
             SELECT((*IF *ORGNETID *EQ 'IOWA'))  SEQNBR(*GEN)
             GROUP(IOWA)
```

This command assigns any problems with a origin system network ID of IOWA to group IOWA.

Example 3: Assigning Entries by Problems for Messages

```
ADDPRBSLTE   FILTER(PROBLIB/PROBFILTER)
             SELECT((*IF *MSGID *EQ 'CPF89*'))  SEQNBR(*GEN)
             GROUP(MSGCPF89)
```

This command assigns any problems for message CPF8901, CPF8902, and so on, to group MSGCPF89.

Example 4: Assigning Entries by Hardware Problems

```
ADDPRBSLTE   FILTER(PROBLIB/PROBFILTER)
             SELECT((*IF *RSCHDW *CT 9404)
                    (*OR *RSCHDW *CT 9406)
                    (*OR *RSCHDW *CT 9402))
             SEQNBR(*GEN) GROUP(AS400USER)
```

All problems for System i5 hardware (the hardware resource information containing machine type 9402, 9404 or 9406) are assigned to group AS400USER.

Caution must be taken when using the contains operation. In this example if the sending machine had a serial number containing 9402, 9404, or 9406 it would also match this selection entry even if the machine type was not 9402, 9404, or 9406. A better example follows.

Example 5: Assigning Entries by Hardware Problems

```
ADDPRBSLTE   FILTER(PROBLIB/PROBFILTER)
             SELECT((*IF *RSCHDW *EQ 9404*)
                    (*OR *RSCHDW *EQ 9406*)
                    (*OR *RSCHDW *EQ 9402*))
             SEQNBR(*GEN) GROUP(AS400USER)
```

This command assigns all problems for System i5 hardware (the hardware resource information equals machine type 9402, 9404 or 9406) to group AS400USER.

This is a better way to select on the sending hardware machine type. Only those machines with types of 9402, 9404, or 9406 will result in a match.

Example 6: Assigning Entries by Machine-detected Problems

```
ADDPRBSLTE   FILTER(PROBLIB/PROBFILTER)
             SELECT((*IF  *PROBTYPE *EQ 1))
             SEQNBR(*GEN)  GROUP(MACHDETECT)
```

This command assigns any problems that are machine-detected to group MACHDETECT.

Example 7: Assigning Entries by Product-specific Problems

```
ADDPRBSLTE   FILTER(PROBLIB/PROBFILTER)
             SELECT((*IF *RSCSFW *EQ '5716SS1 03 06 00'))
             SEQNBR(15)  GROUP(OS400V3R6)
```

This command assigns any problems that are specifically for Operating System Version 3 Release 6 Modification 0 to group OS400V3R6. Notice that this entry is placed after entry number 10 in the filter, since 15 is specified as the sequence number.

Example 8: Assigning Entries by Matching Products

```
ADDPRBSLTE   FILTER(PROBLIB/PROBFILTER)
             SELECT((*IF *RSCSFW *EQ '5716SS1*'))
             SEQNBR(25)  GROUP(OS400)
```

This selection entry matches Version 3 Release 6 of the Operating System licensed program.

Notes:

1. The order of selection entries within a filter is important. When the filter is applied to the problem log entry, the selection entries are examined from the first entry to the last entry in ascending order. The first selection entry that matches a problem is used. To ensure correct operation the most specific selection entries should be first, and the least specific selection entries last.
2. If the selection entries are not order specific (i.e. each selection entry matches one and only one problem) then the most likely or the most common should be placed first. This will ensure the best performance as fewer selection entries will need to be checked.
3. If no selection entries result in a match when a filter is applied, then the *LAST selection entry is used to assign a group. The *LAST selection entry is automatically added to the filter when it is created. The SELECT parameter for the *LAST selection entry is *ANY, which will always result in a match.
4. The *AND logical operator takes precedence over the *OR logical operator within a selection entry. Therefore, the following SELECT specification:

   ```
   ((*IF *PROBTYPE *EQ 1) (*AND *SEV *EQ 1)
   (*OR *PROBTYPE *EQ 2) (*AND *SEV *EQ 1))
   ```

   is equivalent to the following Boolean expression:

   ```
   if ((*PROBTYPE = 1) and (*SEV = 1)) or
   ((*PROBTYPE = 2) and (*SEV = 1))
   ```

5. All attribute values are interpreted as character data, including numbers. When the problem filter is applied to a problem, the system converts all of the data in the filter to the type given in the problem template and compared. Message IDs are considered character data and are ordered as such.

Top

Error messages

***ESCAPE Messages**

CPF2150
Object information function failed.

CPF2151
Operation failed for &2 in &1 type *&3.

CPF7A82
Error occurred while applying the problem filter.

CPF812F
Filter damaged.

CPF91DA
Sequence number &4 already exists.

CPF91DE
Filter &1/&2 at maximum size.

CPF91D9
Sequence number cannot be automatically created.

CPF91EA
*IF relationship not in correct position.

CPF91EB
Filter type &3 not correct for this operation.

CPF91EC
Internal processing error occurred.

CPF91E6
Generic values only allowed with *EQ or *NE.

CPF91E7
Character in position &4 not valid in value specified.

CPF91E8
Internal processing error occurred.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

Top

Add RDB Directory Entry (ADDRDBDIRE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Relational Database Directory Entry (ADDRDBDIRE) command allows you to add an entry to the relational database directory. Relational database (RDB) entries can represent local databases or remote databases. The RDB associated with an entry can also be classified as a system database or a user database.

There is only one system database per system. It is defined as the system auxiliary storage pool (ASP number 1) and configured basic user ASPs (ASP numbers 2-32). A system can be configured to have one or more user databases. A user database is defined to be an ASP group that is configured and available. Such a database is joined to the system database in such a way that all of the objects on the system database are also accessible through it.

Note: As used in this context, 'system' can refer a logical partition of a System i machine configured with multiple partitions.

Local databases include the system database and any available user databases on this system. Remote databases normally reside on another system, but an unavailable ASP group configured on this system is also considered to be temporarily remote, because it might have been switched to another node within a cluster of systems.

Restrictions:

- You must have execute (*EXECUTE) authority to the program specified for the **Application requester driver (ARDPGM)** parameter.

Top

Parameters

Keyword	Description	Choices	Notes
RDB	Entry	*Element list*	Required, Key, Positional 1
	Element 1: Relational database	*Character value*	
	Element 2: Relational database alias	*Character value*, ***NONE**	
RMTLOCNAME	Remote location	Single values: *ARDPGM, *LOOPBACK Other values: *Element list*	Required, Positional 2
	Element 1: Name or address	*Character value*, *LOCAL	
	Element 2: Type	***SNA**, *IP	
TEXT	Text	*Character value*, ***BLANK**	Optional
PORT	Port number or service program	*Character value*, ***DRDA**	Optional

Keyword	Description	Choices	Notes
RMTAUTMTH	Remote authentication method	*Element list*	Optional
	Element 1: Preferred method	***USRENCPWD**, *USRID, *USRIDPWD, *ENCUSRPWD, *KERBEROS, *ENCRYPTED	
	Element 2: Allow lower authentication	***ALWLOWER**, *NOALWLOWER	
DEV	Device	*Element list*	Optional
	Element 1: APPC device description	*Name*, ***LOC**	
LCLLOCNAME	Local location	*Communications name*, ***LOC**, *NETATR	Optional
RMTNETID	Remote network identifier	*Communications name*, ***LOC**, *NETATR, *NONE	Optional
MODE	Mode	*Communications name*, ***NETATR**	Optional
TNSPGM	Transaction program	*Character value*, ***DRDA**	Optional
ARDPGM	Application requester driver	Single values: *DRDA Other values: *Element list*	Optional
	Element 1: Program	*Qualified object name*	
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

Entry (RDB)

Specifies the relational database name information.

This is a required parameter.

Note: Valid relational database names and aliases can contain any of the following: A-Z, 0-9, @, #, $ and _.

Element 1: Relational database

character-value
Specify the relational database name as identified at the remote location. You can specify a maximum of 18 characters for the name; however, DB2 UDB for z/OS relational database names are limited to 16 characters.

Element 2: Relational database alias

***NONE**
There is no local alias for the relational database.

character-value
Specify the relational database alias. The alias is used for locally identifying the relational database specified above. You can specify a maximum of 18 characters for the alias. A relational database alias name is not valid when specified with a *LOCAL remote location name.

Top

Remote location (RMTLOCNAME)

Specifies the remote location name of the system on which the relational database (RDB) is located.

This is a required parameter.

Single values

***ARDPGM**
The RDB is accessed by using the application requester driver program specified on the ARDPGM parameter. A remote location name is not used to locate the RDB.

Note: If *ARDPGM is specified, the PORT, DEV, LCLLOCNAME, RMTNETID, MODE, and TNSPGM parameters are ignored.

***LOOPBACK**
This value is an alias for the IP address of the host system.

Note: If *LOOPBACK is specified, the DEV, LCLLOCNAME, RMTNETID, MODE, TNSPGM and ARDPGM parameters are ignored, and the value of the second element is forced to *IP.

Element 1: Name or address

***LOCAL**
This entry is the system database (system ASP and any basic ASPs) on this system. You can specify *LOCAL for only one entry in the RDB directory.

Note: If *LOCAL is specified, the DEV, LCLLOCNAME, RMTNETID, MODE, TNSPGM and ARDPGM parameters are ignored, and the value of the second element is forced to *IP. A relational database alias name is not valid when specified with a *LOCAL remote location name.

character-value
The first element of this parameter can be specified in several forms:

- SNA remote location name (LU name). Specify a maximum of 8 characters for the remote location name. If this form is used, the second element of this parameter must be *SNA (the default).
- SNA remote network identifier and remote location name separated by a period. Specify a maximum of 8 characters for the remote location name, and a maximum of 8 characters for the remote network identifier. If this form of the parameter is used, the second element of this parameter must be *SNA (the default), and any value specified for the RMTNETID parameter must agree. If the RMTNETID parameter is not specified, the RMTNETID value will be set to agree with the RMTLOCNAME parameter.
- IP version 4 address in dotted decimal form. Specify an internet protocol version 4 address in the form nnn.nnn.nnn.nnn where each nnn is a number in the range 0 through 255. If this form is used, the second element of this parameter must be specified as *IP.
- IP version 6 address in colon hexadecimal form. Specify an internet protocol version 6 address in the form xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx where each xxxx is a hex number in the range 0 through FFFF. If this form is used, the second element of this parameter must be specified as *IP. IP version 6 includes the IPv4-mapped IPv6 address form (for example, ::FFFF:1.2.3.4). For IP version 6, the compressed form of the address is allowed.
- IP host domain name. Specify an internet host domain name of up to 254 characters in length. If this form is used, the second element of this parameter must be specified as *IP.

If *IP is specified for the second element, the DRDA server at the remote location must support the use of TCP/IP, and the DEV, LCLLOCNAME, RMTNETID, MODE, and TNSPGM parameters will be ignored.

If *SNA is specified for the second element, the DRDA server must support SNA connectivity. More information about SNA remote location names can be found in the APPC Programming

book, SC41-5443 and the APPN information in the Networking category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Element 2: Type

***SNA** The RDB system is accessed using a Systems Network Architecture (SNA) address and protocol.

***IP** The RDB system is found using a host name or an internet address over a TCP/IP connection.

Top

Text (TEXT)

Specifies the text that briefly describes the object.

***BLANK**
No text is specified.

character-value
Specify no more than 50 characters of text enclosed in single quotation marks.

Top

Port number or service program (PORT)

Specifies the TCP/IP port that is used at the remote location to communicate with the system on which the RDB is located. This parameter will be ignored if *IP is not specified in the RMTLOCNAME parameter.

***DRDA**
The DRDA well-known port of 446 will be used.

port-number
Specify a number ranging from 1 through 65535.

service-name
Specify a maximum of 14 characters for the service name. This name must be registered in the service database file.

Top

Remote authentication method (RMTAUTMTH)

Specifies the preferred remote authentication method on a DDM/DRDA TCP/IP connection request. The actual method used depends on the outcome of the negotiation process between client and server, which depends on the cryptographic support available and the server security configuration. The CHGDDMTCPA (Change DDM TCP/IP Attributes) command can be used to configure DDM/DRDA TCP/IP security on i5/OS systems. This parameter will be ignored if *IP is not specified in the **Remote location** (RMTLOCNAME parameter).

Element 1: Preferred method

Specifies the initial authentication method proposed to the server. Based on the authentication methods supported by the server and the value specified for the **Allow lower authentication** element of this parameter, an authentication method is negotiated that is acceptable to both the client and server.

***USRENCPWD**
User ID and associated encrypted password is sent on a DDM connection request. Cryptographic support must be available on both systems for this authentication method to be used.

***USRID**
User ID only is sent on a DDM connection request. This is the lowest authentication method.

***USRIDPWD**
User ID and associated password is sent on a DDM connection request. Passwords are not encrypted if this authentication method is used.

***ENCUSRPWD**
Encrypted user ID and associated encrypted password is sent on a DDM connection request. Cryptographic support must be available on both systems for this authentication method to be used.

***KERBEROS**
Authentication occurs using Kerberos. The RDB name must map to a target principal name in the Enterprise Identity Mapping (EIM) environment. Kerberos needs to be configured on both systems for this authentication method to be used.

Note: The following value is only supported for compatibility with the releases earlier than Version 5 Release 5 Modification 0 of the operating system.

***ENCRYPTED**
User ID and associated encrypted password is sent on a DDM connection request. Cryptographic support must be available on both systems for this authentication method to be used. It is recommended to use value *USRENCPWD in place of value *ENCRYPTED.

Element 2: Allow lower authentication

Specifies whether an authentication method lower than what was specified for the **Preferred method** element of this parameter will be accepted during negotiation with the server. If the server is configured to require a higher authentication method than the value specified for the **Preferred method** element of this parameter and the Application Requester system can support a higher authentication method, the negotiated authentication method can always be higher than the **Preferred method**. From highest to lowest, the authentication methods are:

- *KERBEROS
- *ENCUSRPWD
- *USRENCPWD or *ENCRYPTED
- *USRIDPWD
- *USRID

***ALWLOWER**
Allow negotiation of a lower authentication method than what was specified for the **Preferred method** element of this parameter.

***NOALWLOWER**
Do not allow negotiation of a lower authentication method than what was specified for the **Preferred method** element of this parameter.

Top

Device (DEV)

Specifies the advanced program-to-program communications (APPC) device description on this system that is used with this relational database (RDB) entry.

More information on device names is in the APPC Programming book, SC41-5443.

***LOC** If APPC is being used, the system determines which device description is used. If advanced peer-to-peer networking (APPN) is being used, the system ignores this parameter.

name Specify a maximum of 10 characters for the name of a device description.

Top

Local location (LCLLOCNAME)

Specifies the local location name by which this system is identified to the system on which the RDB is located. The local location name cannot be the same as the remote location name.

***LOC** If advanced program-to-program communications (APPC) is being used, the system determines which local location name is used. If advanced peer-to-peer networking (APPN) is being used, the system uses the default local location defined in the network attributes.

***NETATR**
The LCLLOCNAME value specified in the system network attributes is used.

communications-name
Specify a maximum of 8 characters for the local location name.

Top

Remote network identifier (RMTNETID)

Specifies the remote network identifier of the system on which the RDB is located. If this parameter is specified, the RMTLOCNAME parameter must be consistent with this RMTNETID parameter. If the RMTLOCNAME parameter specified a network ID, this parameter must agree (otherwise, an error message will be issued). If the RMTLOCNAME parameter does not specify any network ID, there is no possibility of conflict with this parameter.

More information on remote network identifiers is in the APPC Programming book, SC41-5443.

***LOC** If advanced program-to-program communications (APPC) is being used, the system determines which remote network identifier is used. If advanced peer-to-peer networking (APPN) is used, the system uses the local network identifier defined in this system's network attributes for the remote network identifier.

***NETATR**
The remote network identifier specified in the network attributes is used.

***NONE**
No remote network identifier (ID) is used.

remote-network-identifier
Specify a maximum of 8 characters for the remote network identifier.

Top

Mode (MODE)

Specifies the mode name to use with the remote location name to communicate with the system on which the RDB is located.

***NETATR**
The mode in the network attributes is used.

BLANK
A mode name of all blanks is used.

communications-name
Specify a maximum of 8 characters for the mode name.

Top

Transaction program (TNSPGM)

Specifies the name of the transaction program to use with the RDB entry.

***DRDA**
The distributed relational database architecture (DRDA) transaction program name, X'07F6C4C2', is used. DRDA is a means by which RDBs communicate with each other over a network.

name Specify the name of the transaction program in one of the following formats:

- A 4-byte hexadecimal name, which is entered by enclosing the 8 hexadecimal digits in single quotation marks with a prefix of X. For example, X'07F6C4C2' is a 4-byte hexadecimal name.
- An 8-byte character name.

Top

Application requester driver (ARDPGM)

Specifies the application requester driver that is the program to be called to process SQL requests directed to the RDB. The program must exist in a library that is located in the system database (system ASP or a configured basic user ASP) on this system, and must be of the object type *PGM.

Single values

***DRDA**
The Distributed Relational Database Architecture (DRDA) application requester is used.

Qualifier 1: Program

name Specify the name of the application requester driver program to be called to process the SQL requests.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched.

name Specify the name of the library where the program is located.

Top

Examples

Example 1: Adding an Entry

```
ADDRDBDIRE   RDB(MYRDB)
             RMTLOCNAME(*LOCAL)
```

This command adds an entry to the relational database directory. The entry identifies the local relational database. In an SQL program, this relational database name is used when referring to the local relational database.

Example 2: Adding an Entry

```
ADDRDBDIRE   RDB(YOURRDB)
             RMTLOCNAME(NEWYORK)
```

This command adds an entry to the relational database directory. The entry identifies a remote location, NEW YORK.

Example 3: Adding an Entry for an Application Requester Driver Program

```
ADDRDBDIRE   RDB(YOURRDB)
             RMTLOCNAME(*ARDPGM)
             ARDPGM(MYLIB/MYPGM)
```

This command adds an entry to the relational database directory. The entry indicates that access to relational database YOURRDB will be done by an application requester driver program named MYPGM in the library MYLIB.

Example 4: Adding an Entry for TCP/IP usage

```
ADDRDBDIRE   RDB(TCPRDB)
             RMTLOCNAME(ROCHESTER.XYZ.COM *IP)
             PORT(*DRDA)
```

This command adds an entry to the relational database directory. The entry specifies that the remote RDB associated with the RDB name of TCPRDB uses TCP/IP and is on the host with the domain name of ROCHESTER.XYZ.COM, and listens on the standard DRDA port of 446 (*DRDA is the default port so the PORT parameter is unnecessary in this case).

Example 5: Adding an Entry for TCP/IP using Dotted Decimal IP Version 4 Address and a Numeric Port Number

```
ADDRDBDIRE   RDB(DB2DSYS)
             RMTLOCNAME('9.5.36.17' *IP)
             PORT(5021)
```

This command adds an entry to the relational database directory. The entry specifies that the remote RDB associated with the RDB name of DB2DSYS uses TCP/IP and is on the host with an IP address of

9.5.36.17, and listens on port 5021. A System/390 MVS installation, for example, can have multiple DB2 subsystems, and TCP/IP can support only one server at each port number, so port numbers other than 446 are sometimes required.

Example 6: Adding an Entry for TCP/IP using Colon Hexadecimal IP Version 6 Address and a Numeric Port Number

```
ADDRDBDIRE   RDB(DB2DSYS)
             RMTLOCNAME('2001:DB8:0:B33D:8785:0:1734:F51C' *IP)
             PORT(32)
```

This command adds an entry to the relational database directory. The entry specifies that the remote RDB associated with the RDB name of DB2DSYS uses TCP/IP and is on the host with an IP address of 2001:DB8:0:B33D:8785:0:1734:F51C, and listens on port 32. A System/390 MVS installation, for example, can have multiple DB2 subsystems, and TCP/IP can support only one server at each port number, so port numbers other than 446 are sometimes required.

Example 7: Adding an Entry for TCP/IP using a Service Name for the Port Identification

```
ADDRDBDIRE   RDB(DB2ESYS)
             RMTLOCNAME(ROCHESTER.XYZ.COM *IP)
             PORT(DB2ESYS_PORT)
```

This command uses a service name to specify the port number when adding a new entry. The operating system will attempt to resolve the name DB2ESYS_PORT to a port number by use of the TCP/IP Service Table. In order for the name to be properly resolved, an entry for DB2ESYS_PORT must exist in the TCP/IP Service Table. The WRKSRVTBLE or CFGTCP command can be used to update the service table.

Top

Error messages

***ESCAPE Messages**

CPF3EC0

Add relational database directory entry failed.

Top

Add REXX Buffer (ADDREXBUF)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

Parameters
Examples
Error messages

The Add REXX Buffer (ADDREXBUF) command creates a buffer in the REXX external data queue.

Top

Parameters

Keyword	Description	Choices	Notes
BUFFER	Buffer	*Decimal number*	Optional, Positional 1

Top

Buffer (BUFFER)

Specifies the name of the variable that receives the number of the new buffer. In a control language (CL) program, a decimal variable with a minimum length of 11 digits and no decimal positions must be specified.

Top

Examples

```
ADDREXBUF
```

This command creates a logical buffer within the REXX external data queue.

Top

Error messages

***ESCAPE Messages**

CPF7CF7
REXX external data queue is damaged.

CPF7CF8
REXX external data queue is full.

Top

Add Remote Definition (ADDRMTDFN)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Remote Definition (ADDRMTDFN) command is used to define the attributes of a remote system and add them to the remote system definition table.

Restriction: The user must have *ALLOBJ authority.

Top

Parameters

Keyword	Description	Choices	Notes
SYSTEM	System name	*Element list*	Required, Positional 1
	Element 1: System name	*Character value*, *ANY	
	Element 2: System group	*Character value*	
TEXT	Text	*Character value*, ***BLANK**	Optional, Positional 2
MTGNTCDOC	Meeting notice document type	***FFTDCA**, *EMN	Optional
CALDTASTM	Calendar data stream	Single values: *NONE Other values (up to 5 repetitions): *Communications name*, ***OV400**	Optional
RMTCALPWD	Calendar password	*Simple name*, ***NONE**	Optional
RMTUSRAUT	Remote user authority	***PRIVATE**, *PUBLIC, *MINIMUM, *EXCLUDE	Optional
RMTLOCNAME	Remote location	*Communications name*, ***SYSTEM**	Optional
LCLLOCNAME	Local location	*Communications name*, ***LOC**, *NETATR	Optional
RMTNETID	Remote network identifier	*Communications name*, ***LOC**, *NETATR, *NONE	Optional
MODE	Mode	*Communications name*, ***NETATR**	Optional

Top

System name (SYSTEM)

Specifies the system name and system group of the remote system being defined.

This is a required parameter.

The possible values are:

***ANY** Adds a default definition for all remote systems not covered by the other entries.

The possible **System Name** value is:

system-name
Specify the name of the remote system being defined.

The possible **System Group** value is:

system-group
Specify the group name of the remote system being defined. The system group name is blank if this value is not specified.

Top

Text (TEXT)

Specifies text that describes the remote system definition.

The possible values are:

***BLANK**
Text is not specified.

'description'
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Meeting notice document type (MTGNTCDOC)

Specifies the type of meeting notice documents accepted by the remote system. If the system can accept Enterprise Meeting Notice Architecture documents (post-V2R1M1 AS/400 systems), you should specify *EMN for this parameter. If you are unsure, specify *FFTDCA for this parameter.

The possible values are:

***FFTDCA**
The remote system does not accept enterprise meeting notice documents, but does accept final-form text documents.

***EMN** The remote system accepts enterprise meeting notice documents.

Top

Calendar data stream (CALDTASTM)

Specifies the type of calendar data stream that the local system uses when sending a request for calendar information to this remote system. Each type of calendar data stream represents a format in which remote calendar requests are made from the local system to this remote system.

The possible values are:

***OV400**
The OfficeVision calendar data stream is used.

calendar-data-stream
Specify the name of the calendar data stream that is used. The name of the data stream can be a maximum of 10 characters.

Single Value:

***NONE**
No calendar data stream is used.

Top

Calendar password (RMTCALPWD)

Specifies the password that is associated with user profile QRMTCAL on the remote system. This user profile is used to sign on to the remote system when processing a request for calendar information.

The possible values are:

***NONE**
No password is used for user profile QRMTCAL.

calendar-password
Specify the password that is defined for QRMTCAL. If the password is numeric, it must begin with a Q (for example, specify Q1234 when 1234 is the password).

Top

Remote user authority (RMTUSRAUT)

Specifies the object authority for calendar objects on the local system to be used for incoming requests for calendar information from remote system users. This parameter is used by OfficeVision calendar processing to determine authority to calendars.

The possible values are:

***PRIVATE**
Private authority is used for requests from the remote system. If private authority does not exist, public authority is used.

***PUBLIC**
Public authority is used for requests from the remote system.

***MINIMUM**
The lesser of the private or the public authority is used for requests from the remote system.

***EXCLUDE**
Local system objects cannot be accessed by users on the remote system.

Top

Remote location (RMTLOCNAME)

Specifies the remote location name of the remote system being added.

The possible values are:

***SYSTEM**
The name specified on the SYSTEM parameter is used for the remote location name.

remote-location-name
Specify the full name of a remote location.

Top

Local location (LCLLOCNAME)

Specifies the location name that identifies the local system to the remote system being added.

The possible values are:

***LOC** The local location name associated with the remote location is used.

***NETATR**
The LCLLOCNAME value specified in the system network attributes is used.

local-location-name
Specify the name of the local location.

Top

Remote network identifier (RMTNETID)

Specifies the remote network identifier (ID) for the remote system being added.

The possible values are:

***LOC** The remote network ID associated with the remote location is used. If several remote network IDs are associated with the remote location, the system determines which remote network ID is used.

***NETATR**
The RMTNETID value specified in the system network attributes is used.

***NONE**
No remote network ID is used.

remote-network-ID
Specify the remote network ID.

Top

Mode (MODE)

Specifies the name of the mode that defines the device sessions used to request data from the remote system.

The possible values are:

***NETATR**
The mode name specified in the network attributes is used.

mode-name
Specify the name of the mode.

Top

Examples

Example 1: Adding a Specific Remote Definition

```
ADDRMTDFN   SYSTEM(ABCXYZ)  TEXT('System XYZ')  MTGNTCDOC(*EMN)
```

This command adds a definition for a remote system ABCXYZ and allows the system to accept enterprise meeting notice documents.

Example 2: Allowing Final Form Text Documents

```
ADDRMTDFN   SYSTEM(*ANY)  MTGNTCDOC(*FFTDCA)
```

This command allows all remote systems that do not have specific remote definitions to accept final form text meeting notices.

Example 3: Adding a Remote Definition with Password

```
ADDRMTDFN   SYSTEM(DALLAS1)  TEXT('SYSTEM1')  MTGNTCDOC(*EMN)
            RMTCALPWD(CALPWD)  LCLLOCNAME(*NETATR)
```

This command adds a definition for the remote system DALLAS1, which accepts enterprise meeting notice documents. The password to sign on the system is CALPWD. The remote system will identify the local system by the name specified in the system network attributes.

Top

Error messages

***ESCAPE Messages**

CPF6DCA
SYSTEM parameter cannot be local system.

CPF6DCB
Remote definition for system &1 &2 already exists.

CPF9899
Error occurred during processing of command.

Top

Add Remote Journal (ADDRMTJRN)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Remote Journal (ADDRMTJRN) command associates a remote journal on the target system, as identified by the relational database directory entry, with the specified journal on the source system. The journal on the source system may be either a local journal or another remote journal. A maximum of 255 remote journals may be associated with a single journal on a source system.

When adding a remote journal to a source journal, the remote journal is created on the target system using a combination of the attributes from the source journal and the input parameters provided on this command. The library that the remote journal will be created in must already exist on the target system prior to this command being used on the source system. When created by this command, the remote journal will be created with a journal type of *REMOTE and the remote journal will not have an attached journal receiver.

Note: A receiver will be attached when the remote journal is activated using either the Change Remote Journal (CHGRMTJRN) command, or Change Journal State (QjoChangeJournalState) API.

When adding the remote journal, the remote journal can either be created into the same named library as that of the source journal or into a redirected library on the target system. A redirected library provides a means for remote journals and any of their associated journal receivers to reside in different named libraries on the target system from the corresponding local journal and journal receivers on the local system. When specified, all validation for the journal library on the target system will be performed using the redirected library name. Similarly, the journal receivers that will later be created and associated with this remote journal can either reside in the same library as the source journal receivers on the source system, or into a distinct redirected library name on the target system. The journal receiver library redirection, if desired, must be specified when the remote journal is added using this command.

When adding a remote journal on a target system, two remote journal types can be specified, *TYPE1 and *TYPE2. The remote journal type influences the redirection capabilities, journal receiver restore operations, and remote journal association characteristics.

- Every *TYPE1 remote journal that is associated with the local journal must use the same redirection for both the remote journal and remote journal receiver. This allows a journal receiver to be saved from any system in the remote journal network and be restored to any system in the network and associated with either a local or remote journal.
- Every *TYPE2 remote journal that is associated with the local journal can use different redirection for the remote journal or remote journal receiver. This restricts the save and restore options. The journal receiver can only be associated with the remote journal if it was saved from the system on which the remote journal resides. If the journal receiver was saved from the system with the local journal, then it cannot be associated with the remote journal on any system.

If the specified journal already exists on the target system, the journal can be associated with the source journal if all of the following are true:

- the journal is of type *REMOTE
- the remote journal type matches the specified remote journal type
- the remote journal was previously associated with this same source journal

Also, the journal may or may not have an attached journal receiver.

After the remote journal has been successfully added on the target system, the remote journal will have a journal state of *INACTIVE. A journal state of *INACTIVE for a remote journal means that the remote journal is currently not ready to receive journal entries from its source journal on the source system. The Change Remote Journal (CHGRMTJRN) command or Change Journal State (QjoChangeJournalState) API is used to activate a remote journal and start the replication of journal entries from the source journal to the remote journal.

Once a remote journal has been added to a journal, the receiver that was attached to the source journal at the time of running this command or any subsequently attached receivers, will be protected from deletion if all journal entries for a given journal receiver have not yet been replicated to the remote journal. This protection ends when the remote journal is removed using the Remove Remote Journal (RMVRMTJRN) command or Remove Remote Journal (QjoRemoveRemoteJournal) API.

Restrictions:

- The Add Remote Journal (ADDRMTJRN) command may only be called from the source system.
- A user profile must exist on the target system by the same name as the user profile that is running the Add Remote Journal (ADDRMTJRN) command on the source system. This restriction is irrespective of the selected communications protocol.
- When adding a *TYPE1 remote journal to a source journal, the same journal and journal receiver library redirection must be specified that exists for any *TYPE1 remote journals which have already been added to the source journal. A remote journal will always use the redirected library, if any, that is specified for the local journal. The only way to change the value specified in the remote journal library field and the remote journal receiver library field is to do all of the following:
  1. Remove all of the associated *TYPE1 remote journals from the local journal.
  2. Delete the remote journal.
  3. Change the local journal to attach a new receiver.
  4. Add the remote journal specifying the new redirection.
- QTEMP cannot be specified for the remote journal library, remote journal receiver library, or remote message queue library.
- A remote journal whose name starts with a Q cannot specify a remote journal library that starts with a Q, unless the remote journal library is QGPL. This is required to prevent collisions between local and remote journals that are used for system functions.
- A *TYPE1 remote journal cannot be added to a *TYPE2 remote journal.
- The specified relational database (RDB) directory entry must meet the following rules:
  - The communications protocol used for the RDB must be supported for remote journal functions.
  - The remote location name in the RDB directory entry cannot refer to the *LOCAL database.
  - The RDB directory entry cannot use an application requester driver program (*ARDPGM) to locate the target system.
- The remote journal message queue on the remote system must be either in the same ASP group as the remote journal, or in the system ASP, or a basic user ASP.
- The remote receiver library and remote journal library on the remote system must both exist in either the system and basic user ASP's or in the same ASP group. They cannot be in two different ASP groups.

Top

Parameters

Keyword	Description	Choices	Notes
RDB	Relational database	*Name*	Required, Positional 1

Keyword	Description	Choices	Notes
SRCJRN	Source journal	*Qualified object name*	Required, Positional 2
	Qualifier 1: Source journal	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TGTJRN	Target journal	Single values: ***SRCJRN** Other values: *Qualified object name*	Optional
	Qualifier 1: Target journal	*Name*	
	Qualifier 2: Library	*Name*	
RMTRCVLIB	Remote receiver library	*Name*, ***SRCRCVLIB**	Optional
RMTJRNTYPE	Remote journal type	***TYPE1**, *TYPE2	Optional
MSGQ	Journal message queue	*Qualified object name*	Optional
	Qualifier 1: Journal message queue	*Name*, **QSYSOPR**	
	Qualifier 2: Library	*Name*, **QSYS**	
DLTRCV	Delete receivers	***NO**, *YES	Optional
DLTRCVDLY	Delete receiver delay time	1-1440, **10**	Optional
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Top

Relational database (RDB)

Specifies the name of the relational database directory entry that contains the remote location name of the target system. This name should match the name of the *LOCAL relational database directory entry on the target system.

This is a required parameter.

relational-database-entry
Specify a maximum of 18 characters for the name of the relational database directory entry.

Top

Source journal (SRCJRN)

Specifies the name of the journal on the source system to which the remote journal is being added, and the library where it resides. The journal on the source system may be either a local journal or another remote journal.

This is a required parameter.

source-journal-name
Specify the source journal to which the target journal is being added.

The name of the source journal can be qualified by one of the following library values:

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched.

library-name
Specify the name of the library to be searched.

Top

Target journal (TGTJRN)

Specifies the name of the remote journal on the target system.

***SRCJRN**
The target journal name is exactly the same as the source journal name.

library-name/target-journal-name
Specify the target journal that is being added as a remote journal to the source journal.

Top

Remote receiver library (RMTRCVLIB)

Specifies the name of the library for the remote journal receivers on the target system that will be associated with this remote journal.

***SRCRCVLIB**
The journal receivers are created on the target system in the same library as they exist on the source system.

remote-journal-receiver-library-name
Specify the name of the library for the remote journal receivers on the target system that will be associated with this remote journal.

Top

Remote journal type (RMTJRNTYPE)

Specifies the type of remote journal on the target system. The remote journal type influences the redirection capabilities, journal receiver restore operations, and remote journal association characteristics. See the Journal management topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for detailed descriptions of the differences.

***TYPE1**
A *TYPE1 remote journal is added. Every *TYPE1 remote journal that is associated with the local journal must use the same redirection for both the remote journal and remote journal receiver. This allows a journal receiver to be saved from any system in the remote journal network and be restored to any system in the network and associated with either a local or remote journal.

***TYPE2**
A *TYPE2 remote journal is added. Every *TYPE2 remote journal that is associated with the local journal can use different redirection for the remote journal or remote journal receiver. This restricts the save and restore options. The journal receiver can only be associated with the remote journal if it was saved from the system on which the remote journal resides. If the journal receiver was saved from the system with the local journal, then it cannot be associated with the remote journal on any system.

Top

Journal message queue (MSGQ)

Specifies the name of the message queue associated with the remote journal. This value is only set for a journal that is created on the target system.

QSYS/QSYSOPR
The message is sent to the QSYSOPR message queue.

library-name/journal-message-queue
Specify the name of the journal message queue to which the journal messages are sent. If this message queue is not available when a message is to be sent, the message is sent to the QSYSOPR message queue.

Note: Some messages that are sent to the journal message queue will also be sent to the QSYSOPR message queue and QHST.

Top

Delete receivers (DLTRCV)

Specifies whether the system deletes the target journal receivers when they are no longer needed or keeps them on the target system for the user to delete after they have been detached by the target system. This value is only set for a journal that is created on the target system.

***NO** The journal receivers are not deleted by the system.

***YES** The journal receivers are deleted by the system.

Top

Delete receiver delay time (DLTRCVDLY)

If the system cannot allocate an object needed to delete a journal receiver associated with the remote journal on the target system, this parameter specifies the time (in minutes) to be used to delay the next attempt to delete the target journal receiver. This value is only set for a journal that is created on the target system.

10 System waits 10 minutes before trying again.

delete-receiver-delay-time
System waits the specified number of minutes before trying again. Valid values range from 1 to 1440.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the remote journal on the target system. This value is only set for a journal that is created on the target system.

***BLANK**
Text is not specified.

'description'
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Examples

Example 1: Adding a *TYPE1 remote journal, specifying values for the DLTRCV and TEXT parameters.

```
ADDRMTJRN   SRCJRN(LOCLIB/J)  RDB(DETROIT)  TGTJRN(RMTLIB/J)
            DLTRCV(*NO)
            TEXT('Remote Journal Created for Application Z')
```

This command adds remote journal J in library RMTLIB to journal J in library LOCAL, and the DLTRCV parameter on the remote journal will be *NO, irrespective of the attribute of journal J in library LOCLIB. If journal J in RMTLIB does not already exist, then it is created, otherwise, it is reassociated with journal J in LOCLIB, if it meets the appropriate criteria.

Top

Error messages

***ESCAPE Messages**

CPF69A4
Remote journal &1 in &2 not added.

CPF695A
Remote journal &1 in &2 not added.

CPF695B
Remote journal &1 in &2 not added.

CPF695C
Remote journal &1 in &2 not added.

CPF695D
Remote journal &1 in &2 not added.

CPF695E
Remote journal &1 in &2 not added.

CPF695F
Remote journal &1 in &2 not added.

CPF6973
Source journal not compatible with target system.

CPF6982
Relational database directory entry &1 not valid.

CPF6983
Remote journal &1 in &2 not added.

CPF6984
Remote journal &1 in &2 not added.

CPF6985
Remote journal &1 in &2 not added.

CPF6988
Remote journal &1 in &2 not added.

CPF6989
Remote journal &1 in &2 not added.

CPF699B
User profile &8 not found.

CPF6991
Remote journal &1 in &2 not added.

CPF70DB
Remote journal environment ended for journal &1 in library &2.

CPF70D6
Remote journal environment ended for journal &1 in library &2.

CPF701B
Journal recovery of an interrupted operation failed.

CPF7010
Object &1 in &2 type *&3 already exists.

CPF7011
Not enough storage or resources.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9830
Cannot assign library &1.

Top

Add Reply List Entry (ADDRPYLE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Reply List Entry (ADDRPYLE) command is used to add an entry to the system-wide automatic inquiry message reply list. The automatic message reply list is the source for default responses to inquiry messages. Each entry in the inquiry message list specifies both a message identifier and the reply that is sent when that message is sent as an inquiry message. The entry may also include comparison data, which further qualifies the message identifier. The message identifier may be specific or generic in scope. One of the following actions may be taken when one of the specific inquiry messages is issued:

- The default reply specified in the inquiry message file is sent to the message reply queue specified when the inquiry message was sent.
- A specific reply to the inquiry message is sent to the message reply queue specified when the inquiry message was sent.
- A manual reply to the inquiry message may be required from the operator.

The entry may also specify the dumping of information associated with the job that is sending the inquiry message.

The reply list is used only when an inquiry message is sent by a job that has the inquiry message reply attribute specified as INQMSGRPY(*SYSRPYL). The INQMSGRPY attribute can be changed by using the Change Job Description (CHGJOBD) command.

Specific attributes of a reply list entry can be changed by using the Change Reply List Entry (CHGRPYLE) command. Each reply list entry remains in the list until it is removed by the Remove Reply List Entry (RMVRPYLE) command. The list can be shown by using the Work with Reply List Entry (WRKRPYLE) command.

Restrictions:

1. This command is shipped with public *EXCLUDE authority and the QPGMR user profile has private authority to use the command.
2. To use this command, you must be signed on as QPGMR, or have *USE special authority.

Top

Parameters

Keyword	Description	Choices	Notes
SEQNBR	Sequence number	1-9999	Required, Positional 1
MSGID	Message identifier	*Character value*, ***ANY**	Optional
CMPDTA	Compare data	*Element list*	Optional
	Element 1: Comparison data	*Character value*, ***NONE**	
	Element 2: Message data start position	1-999, ***NONE**	
RPY	Message reply	*Character value*, ***DFT**, *RQD	Optional
DUMP	Dump the sending job	***NO**, *YES	Optional

Keyword	Description	Choices	Notes
CCSID	Coded character set ID	1-65535, ***HEX**, *JOB	Optional

Top

Sequence number (SEQNBR)

Specifies the sequence number of the reply list entry being added to the reply list. The message identifier and message data of an inquiry message are matched against the reply list entry message identifiers and comparison data in ascending sequence number order. The search ends when a match occurs or the last reply list entry is passed.

Sequence numbers can range from 0001 to 9999. Duplicate sequence numbers are not allowed.

This is a required parameter.

Top

Message identifier (MSGID)

Specifies the inquiry message identifiers for which automatic system action is taken. The message identifier can be specific or generic in scope. Only predefined messages (messages known to the system by a message identifier) can be matched by reply list entries. Immediate messages cannot be used for comparison.

***ANY** This reply list entry matches any message identifier.

message-identifier
Specify a message identifier that is compared with the message identifier of an inquiry message. The message identifier must be 7 characters in length and in the following format: *pppnnnn*

The first 3 characters (ppp) must be a code consisting of one alphabetic character followed by two alphanumeric (alphabetic or decimal) characters; the last 4 characters (nnnn) must consist of the decimal numbers 0 through 9 and the characters A through F.

Top

Compare data (CMPDTA)

Specifies the comparison data that is used to determine whether this entry matches an inquiry message. If the identifier of the inquiry message matches the message identifier of this reply list entry, then the message data specified for the inquiry message is compared to this data.

Element 1: Comparison data

***NONE**
No comparison data is specified. If the inquiry message has the specified identifier, the action specified by this reply list entry is taken.

'comparison-data'
Specify a character string of no more than 28 characters (enclosed in apostrophes if blanks or other special characters are included). This string is compared with a string of the same length in the message data portion of the inquiry message.

Element 2: Message data start position

***NONE**
No starting position value is specified. If comparison data is specified for element 1 of this parameter, the default start value is 1.

message-data-start
Specify the character position in the message's replacement text (maximum value is 999) where the comparison data will start being compared to the replacement text. A start value is not valid without a specification of comparison data.

Coded Character Set Identifier (CCSID) Considerations

The text supplied for the CMPDTA parameter that corresponds to the *CCHAR type field is assumed to be in the CCSID of the job running this command unless, the CCSID parameter is coded. If the CCSID parameter is coded, the text is assumed to be in the CCSID specified. For more information about the *CCHAR type field, see the Add Message Description (ADDMSGD) command.

Top

Message reply (RPY)

Specifies how to reply to an inquiry message that matches this reply list entry. The reply specified is automatically sent by the system without requiring user intervention. The inquiry message does not cause the job to be interrupted or notified when the message has arrived at the message queue.

***DFT** The default reply to the inquiry message is sent.

***RQD** The inquiry message requires an explicit reply. No reply is automatically sent.

'message-reply'
Specify a character string of no more than 32 characters, enclosed in apostrophes (if blanks or other special characters are included) that is sent as a reply to the inquiry message.

Top

Dump the sending job (DUMP)

Specifies whether the job that sent the inquiry message is to be dumped.

***NO** The job is not dumped.

***YES** The job is dumped before the control returns to the program that is sending the message.

Top

Coded character set ID (CCSID)

Specifies the coded character set identification (CCSID) that the text specified for the CMPDTA parameter that corresponds to the *CCHAR type field is to be considered in.

***HEX** The CMPDTA that corresponds to the *CCHAR data type field is assumed to be 65535. No conversion occurs before the replacement data is compared with the CMPDTA.

When an inquiry message is sent in a job that is using the system reply list, the *CCHAR replacement data is counted to the CCSID of the CMPDTA that is stored in the system reply list before the comparison is made.

All other compare data is not converted before a comparison is made. For more information about the message handler and its use of CCSIDs, see the i5/OS globalization topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Note: When specifying a CCSID other than *HEX, all the CMPDTA specified is converted from that CCSID to the job CCSID when displayed on the Work with Reply List Entries panel. This occurs even when all CMPDTA does not correspond with *CCHAR data; therefore, when using a CCSID other than *HEX, specifying the length of the *CCHAR data or any other data field is not recommended.

***JOB** The CMPDTA that corresponds to the *CCHAR data type field is assumed to be in the CCSID of the job calling this command.

coded-character-set-identifier
Specify the CCSID you want the CMPDTA that corresponds to *CCHAR data type field to be considered in.

Top

Examples

Example 1: Reply Automatically Sent

```
ADDRPYLE   SEQNBR(10)  MSGID(RPG1241)  RPY(G)
```

This command adds a reply list entry to the reply list for message identifier RPG1241 (database record not found). Whenever a RPG1241 inquiry message is sent by a job that is using the reply list, a reply of 'G' is automatically sent. The inquiry does not cause a job that has allocated the message queue to be interrupted or notified when the inquiry arrives, and no opportunity is given to reply to the message. The sending job does not have a job dump processed.

Example 2: Default Reply is sent; Job Dump Processed

```
ADDRPYLE   SEQNBR(25)  MSGID(RPG1200)  RPY(*DFT)  DUMP(*YES)
```

This command adds a generic reply list entry to the reply list for all RPG12xx messages. Whenever an RPG12xx inquiry message is sent by a job that is using the reply list, the equivalent to DSPJOB OUTPUT (*PRINT) is automatically generated. The default reply will automatically be sent. This is either the default reply specified in the message description or (if none is specified in the message description) the system default reply. The inquiry does not cause a job that has allocated the message queue to be interrupted or notified when the inquiry arrives, and no opportunity is given to reply to the message. The sending job is dumped before control returns to the sending program. Note that because of the sequence numbers, the entry added by the previous example overrides this entry for message identifier RPG1241.

Example 3: Adding a Generic Reply List Entry

```
ADDRPYLE   SEQNBR(30)  MSGID(RPG0000)  RPY(D)  DUMP(*YES)
```

This command adds a generic reply list entry to the reply list for all RPG messages. Whenever an RPG inquiry message is sent by a job that is using the reply list, a reply of 'D' is sent automatically. The inquiry does not cause a job that has allocated the message queue to be interrupted or notified when the inquiry arrives, and no opportunity is given to reply to the message. (If a value of D is not valid for a particular RPGxxxx message, the user must reply as if *RQD were specified for the RPY parameter.) The sending job is dumped before control returns to the sending program. Note that the entries added by the previous two examples will override this entry for all RPG12xx messages.

Example 4: System Reply List for Spooled Output

```
ADDRPYLE   SEQNBR(40)  MSGID(CPA5316)
           CMPDTA('QPSPLPRT  QSYS      QSYSPRT')  RPY(*RQD)
```

This command illustrates how to use the system reply list for spooled output for device QSYSPRT. The file and library name for spooled output is QSYS/QPSPLPRT.

When compare value is specified, it is compared to the message data beginning with replacement variable &1. If the significant field appears in replacement variable &3, the compare value must include a value for replacement variables &1 and &2, or a message data start value may be entered to begin the comparison with replacement variable &3.

The message CPA5316 has a replacement data as follows:

```
&1        ODP file name        &CHAR  10
&2        ODP library name     &CHAR  10
&3        ODP device name      &CHAR  10
```

A compare for device name 'QSYSPRT' in replacement variable &3 must be preceded by values for &1 and &2 if a message data start value is not entered. Blanks are significant.

The message data of QSYSPRT is the DEVICE name as defined in the CPA5316 message. Whenever a CPA5316 inquiry message with comparison data of QSYSPRT is sent by a job that is using the reply list, the operator must make a manual reply to the inquiry. If the message queue to which the inquiry is sent is in break mode, the inquiry message interrupts. A reply is not sent (unless the queue is in the default mode or the message is sent to an external message queue in a batch job), and no job dump is taken.

Another reply list entry identical to the one listed above could be added, but with a different sequence number and with CMPDTA(WSPR01) specified. This would allow a unique response to a message based on the type of printer.

Example 5: Adding Reply List Entry For Any Message Identifier

```
ADDRPYLE   SEQNBR(9999)  MSGID(*ANY)  RPY(*RQD)  DUMP(*YES)
```

This command adds a reply list entry to the reply list for any message identifier. This entry applies to any predefined inquiry message that is not matched by an entry with a lower sequence number. A manual reply to the inquiry message is required for any predefined inquiry message not matched by a previous entry. If the message queue to which the inquiry message is sent is in break mode, the message interrupts. The job that sent the inquiry message is dumped (equivalent to DSPJOB OUTPUT(*PRINT)).

Example 6: Using Comparison Data

```
ADDRPYLE   SEQNBR(5) MSGID(CPA5316) CMPDTA(QSYSPRT 21)
           RPY(I) DUMP(*NO)
```

Assume that the message CPA5316 is sent to QSYSOPR with the message replacement text of TESTEDFILETSTLIBRARYQSYSPRT; because there is a match for MSGID, the message replacement text starting in position 21 (message data start) is tested by comparing it with the comparison data (for the length of the comparison data). This is a match because QSYSPRT = QSYSPRT, and therefore the reply of 'I' is sent.

Top

Error messages

***ESCAPE Messages**

CPF2435
System reply list not found.

CPF2436
System Reply List entry not added or changed.

CPF247E
CCSID &1 is not valid.

CPF2499
Message identifier &1 not valid.

CPF2555
Sequence number &1 already defined in system reply list.

CPF2557
System reply list damaged.

CPF2558
System reply list currently in use.

Top

Add Routing Entry (ADDRTGE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Routing Entry (ADDRTGE) command adds a routing entry to the specified subsystem description. Each routing entry specifies the parameters used to start a routing step for a job. For example, the routing entry specifies the name of the program to run when the routing data that matches the compare value in this routing entry is received.

Restrictions:

1. To use this command, you must have object operational (*OBJOPR), object management (*OBJMGT), and read (*READ) authority to the specified subsystem description and execute (*EXECUTE) authority to the library containing that subsystem description.

Top

Parameters

Keyword	Description	Choices	Notes
SBSD	Subsystem description	*Qualified object name*	Required, Positional 1
	Qualifier 1: Subsystem description	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SEQNBR	Routing entry sequence number	1-9999	Required, Positional 2
CMPVAL	Comparison data	Single values: *ANY Other values: *Element list*	Required, Positional 3
	Element 1: Compare value	*Character value*	
	Element 2: Starting position	1-80, **1**	
PGM	Program to call	Single values: *RTGDTA Other values: *Qualified object name*	Required, Positional 4
	Qualifier 1: Program to call	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
CLS	Class	Single values: ***SBSD** Other values: *Qualified object name*	Optional
	Qualifier 1: Class	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MAXACT	Maximum active routing steps	0-1000, ***NOMAX**	Optional
POOLID	Storage pool identifier	1-10, **1**	Optional
THDRSCAFN	Thread resources affinity	Single values: ***SYSVAL** Other values: *Element list*	Optional
	Element 1: Group	*NOGROUP, *GROUP	
	Element 2: Level	*NORMAL, *HIGH	
RSCAFNGRP	Resources affinity group	***NO**, *YES	Optional

Top

Subsystem description (SBSD)

Specifies the name and library of the subsystem description to which the routing entry is added.

This is a required parameter.

Qualifier 1: Subsystem description

name Specify the name of the subsystem description to which the routing entry is added.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the name of the subsystem description's library to which the routing entry is being added.

Top

Routing entry sequence number (SEQNBR)

Specifies the sequence number of the routing entry that is added or changed. Routing data is matched against the routing entry compare values in ascending sequence number order. Searching ends when a match occurs or the last routing entry is reached. Therefore, if more than one match possibility exists, only the first match is processed.

This is a required parameter.

1-9999 Specify a sequence number between 1 and 9999.

Top

Comparison data (CMPVAL)

Specifies a value that is compared with the routing data to determine whether this routing entry is used for starting a routing step for the job. If the routing data matches the routing entry compare value, that routing entry is used. A starting position in the starting data character string can be used to specify the starting position in the routing data for comparison against the routing entry compare value.

This is a required parameter.

Single values

***ANY** Any routing data is considered a match. To specify *ANY, the routing entry must have the highest sequence number value of any routing entry in the subsystem description.

Element 1: Compare value

character-value
Specify a value (any character string not exceeding 80 characters) that is compared with routing data for a match. When a match occurs, this routing entry is used to start a routing step.

Element 2: Starting position

1 The comparison between the compare value and the routing data begins with the first character in the routing data character string.

1-80 Specify a value, 1 through 80, that indicates which position in the routing data character string is the starting position for the comparison. The last character position compared must be less than or equal to the length of the routing data used in the comparison.

Top

Program to call (PGM)

Specifies the name and library of the program called as the first program run in the routing step. No parameters can be passed to the specified program. The program name can be either explicitly specified in the routing entry, or extracted from the routing data. If a program name is specified in a routing entry, selection of that routing entry results in the routing entry program being called (regardless of the program name passed in an EVOKE function). If the program specified in the EVOKE function is called, *RTGDTA must be specified. If the program does not exist when the routing entry is added or changed, a library qualifier must be specified because the qualified program name is kept in the subsystem description.

This is a required parameter.

Single values

***RTGDTA**
The program name is taken from the routing data that was supplied and matched against this entry. A qualified program name is taken from the routing data in the following manner: the program name is taken from positions 37 through 46, and the library name is taken from positions 47 through 56. Care should be used to ensure that routing entries that specify *RTGDTA are selected only for EVOKE functions on jobs that have specified the program name in the correct position in the routing data.

Qualifier 1: Program to call

name Specify the name of the program that is run from this routing entry.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the library where the named program is located.

Top

Class (CLS)

Specifies the name and library of the class used for the routing steps started through this routing entry. The class defines the attributes of the routing step's running environment. If the class does not exist when the routing entry is added, a library qualifier must be specified because the qualified class name is kept in the subsystem description.

Single values

***SBSD**
The class having the same qualified name as the subsystem description, specified on the **Subsystem description (SBSD)** parameter is used for routing steps started through this entry.

Qualifier 1: Class

name Specify the name of the class used for routing steps started through this entry.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the library name of the class used for routing steps started through this entry.

Top

Maximum active routing steps (MAXACT)

Specifies the maximum number of routing steps (jobs) that can be active at the same time through this routing entry. In a job, only one routing step is active at a time. When a subsystem is active and the maximum number of routing steps is reached, any subsequent attempt to start a routing step through this routing entry fails. The job that attempted to start the routing step is ended, and a message is sent by the subsystem to the job's log.

***NOMAX**
There is no maximum number of routing steps that can be active at the same time and processed through this routing entry. This value is normally used when there is no reason to control the number of routing steps.

0-1000 Specify the maximum number of routing steps that can be active at the same time through this routing entry. If a routing step being started would exceed this number, the job is ended.

Top

Storage pool identifier (POOLID)

Specifies the pool identifier of the storage pool in which the program runs. The pool identifier specified here relates to the storage pools in the subsystem description.

1 Storage pool 1 of this subsystem is the pool in which the program runs.

1-10 Specify the identifier of the storage pool defined for this subsystem in which the program runs.

Top

Thread resources affinity (THDRSCAFN)

Specifies the affinity of threads to system resources.

Single values

***SYSVAL**
When a job is started using this routing entry, the thread resources affinity value from the QTHDRSCAFN system value will be used.

Element 1: Group

***NOGROUP**
Jobs using this routing entry will have affinity to a group of processors and memory. Secondary threads running under the job will not necessarily have affinity to the same group of processors and memory.

***GROUP**
Jobs using this routing entry will have affinity to a group of processors and memory. Secondary threads running under the job will all have affinity to the same group of processors and memory as the initial thread.

Element 2: Level

***NORMAL**
A thread will use any processor or memory if the resources it has affinity to are not readily available.

***HIGH**
A thread will only use the resources it has affinity to, and will wait until they become available if necessary.

Top

Resources affinity group (RSCAFNGRP)

Specifies whether or not jobs using this routing entry will be grouped together having affinity to the same system resources (processors and memory). A value of *YES for this parameter will take precedence over the QTHDRSCAFN system value when set to *NOGROUP.

***NO** Jobs that use this routing entry will not be grouped together.

***YES** Jobs that use this routing entry will be grouped together such that they will have affinity to the same system resources. Jobs that share data in memory may perform better if they have affinity to the same resources.

Top

Examples

Example 1: Adding to the Routing Portion of a Subsystem Description

```
ADDRTGE   SBSD(ORDLIB/PERT)  SEQNBR(46)  CMPVAL(WRKSTN2)
          PGM(ORDLIB/GRAPHIT)  CLS(MYLIB/AZERO)  MAXACT(*NOMAX)
          POOLID(2)
```

This command adds routing entry 46 to the subsystem description PERT in the ORDLIB library. To use routing entry 46, the routing data must start with the character string WRKSTN2 starting in position 1. Any number of routing steps can be active through this entry at any one time. The program GRAPHIT in the library ORDLIB is to run in storage pool 2 by using class AZERO in library MYLIB.

Example 2: Adding to the Subsystem Description

```
ADDRTGE   SBSD(QGPL/ABLE)  SEQNBR(5)  CMPVAL(XYZ)
          PGM(QGPL/REORD)  CLS(LIBX/MYCLASS)  MAXACT(*NOMAX)
```

This command adds routing entry 5 to the subsystem description ABLE in the QGPL library. The program REORD in library QGPL is started and uses the class MYCLASS in LIBX when a compare value of XYZ (starting in position 1) is matched in the routing data. The program runs in storage pool 1, and there is no maximum on the number of active routing steps allowed.

Top

Error messages

***ESCAPE Messages**

CPF1619
Subsystem description &1 in library &2 damaged.

CPF1691
Active subsystem description may or may not have changed.

CPF1697
Subsystem description &1 not changed.

Top

Add Search Index Entry (ADDSCHIDXE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Search Index Entry (ADDSCHIDXE) command is used to load panel group online help information into a search index.

A search index is a reference to online help information from one or more panel groups. A panel group contains online help information, which the user can access from display panels, by pressing the HELP key, or through the index search function using the Start Search Index (STRSCHIDX) command.

The sequence in which panel groups are loaded into a search index controls the sequence in which topic entries are presented when an index search is requested. The topics (ISCH tag entries) from the first-loaded panel group are presented first.

Restrictions:

- You must have change (*CHANGE) authority for the search index that is to be changed, use (*USE) authority for the panel group, and *USE authority for the library where the search index is located.
- Only user-created panel groups can be added to user-created search indexes and only IBM-supplied panel groups can be added to IBM-supplied search indexes.
- Panel group names must be unique within a search index.

Top

Parameters

Keyword	Description	Choices	Notes
SCHIDX	Search index	*Qualified object name*	Required, Positional 1
	Qualifier 1: Search index	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
PNLGRP	Panel group	*Qualified object name*	Required, Positional 2
	Qualifier 1: Panel group	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

Search index (SCHIDX)

Specifies the search index into which entries are to be loaded.

This is a required parameter.

Qualifier 1: Search index

name Specify the name of the search index.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the search index. If no library is specified as the current library for the job, QGPL is used.

name Specify the name of the library where the search index is located.

Top

Panel group (PNLGRP)

Specifies the panel group that contains the help source for which the index entries are to be loaded into the search index.

The search index object contains the name and library of the panel group. When help information is displayed for a search index, the name and library of the panel group that is contained in the search index object is used to find the panel group.

When *LIBL is used to qualify the panel group name, *LIBL is saved in the search index object. When the panel group name is qualified with a library name or with *CURLIB, the actual name of the library containing the panel group is saved in the search index object.

The names of panel groups added to a search index must be unique.

This is a required parameter.

Qualifier 1: Panel group

name Specify the name of the panel group.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is used to locate the panel group. If no library is specified as the current library for the job, QGPL is used.

name Specify the name of the library where the panel group is located.

Top

Examples

```
ADDSCHIDXE   SCHIDX(ACCOUNTING)  PNLGRP(PAYROLL)
```

This command adds panel group PAYROLL to search index ACCOUNTING. Both the panel group and the search index must exist in the library list.

Top

Error messages

***ESCAPE Messages**

CPF6E07
Panel group cannot be added to search index.

CPF6E08
Panel group cannot be added to search index.

CPF6E09
Panel group does not contain any synonyms or root words.

CPF6E12
Panel group not added to search index.

CPF6E47
Panel group &1 cannot be added to search index &3

CPF6E48
Panel group &1 cannot be added to search index &3

CPF6E49
Panel group &1 cannot be added to search index &3

CPF6E61
Panel group &1 cannot be added to search index &3.

CPF6E62
Panel group &1 already exists in search index &3.

CPF6E63
Error occurred while trying to recover from another error.

Top

Add Sphere of Control Entry (ADDSOCE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Sphere of Control Entry (ADDSOCE) command allows a CL user or program to add advanced peer-to-peer (APPN) network node control points to the Alert Sphere of Control.

Top

Parameters

Keyword	Description	Choices	Notes
ENTRY	Entry	Values (up to 50 repetitions): *Element list*	Optional
	Element 1: Network identifier	*Communications name*, *NETATR	
	Element 2: Control point	*Communications name*	

Top

Entry (ENTRY)

Specifies the systems to add to the sphere of control. The systems are specified by network ID and control point name.

You can specify 50 values for this parameter.

Element 1: Network identifier

***NETATR**
Use the NETID network attribute as the value of the Network ID. The NETID network attribute is the value that gets stored. *NETATR is not stored.

communications-name
Specify the network ID of the system you want to add to the sphere of control.

Element 2: Control point

communications-name
Specify the control point name of the system you want to add to the sphere of control.

Top

Examples

```
ADDSOCE   ENTRY((*NETATR RCHSTR1) (*NETATR RCHSTR2))
```

This command adds two systems (RCHSTR1 and RCHSTR2) to the alert sphere of control.

Top

Error messages

Unknown

Top

Add Service Table Entry (ADDSRVTBLE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Service Table Entry (ADDSRVTBLE) command is used to add a service entry to the service table. You can use the service table to manage the mapping of network services to ports and to record the protocols that the services use.

The service table is shipped with some standard port assignments. Values for common functions supported by Transmission Control Protocol/Internet Protocol (TCP/IP) are available to the Internet community in the assigned numbers **RFC** (Request for Comments) document, a formal specification of proposals and standards for a portion of TCP/IP.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
SERVICE	Service	*Character value*	Required, Positional 1
PORT	Port	1-65535	Required, Positional 2
PROTOCOL	Protocol	*Character value*	Required, Positional 3
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional
ALIAS	Aliases	Single values: ***NONE** Other values (up to 4 repetitions): *Character value*, ***NONE**	Optional

Top

Service (SERVICE)

Specifies the network service to be added to the table. A service can be added to the table more than once. Each service must be uniquely identified by a combination of the port number and the protocol name parameters.

This is a required parameter.

character-value
Specify the name of the network service to be added.

Top

Port (PORT)

Specifies the port number to be assigned to the service.

This is a required parameter.

1-65535
Specify the port number for the network service.

Top

Protocol (PROTOCOL)

Specifies the protocol that the service uses. You can specify a maximum of 32 characters for the protocol name. No checking is done to ensure that the protocol exists.

This is a required parameter.

character-value
Specify the name of the protocol that the network service uses.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the network service entry.

***BLANK**
Text is not specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Aliases (ALIAS)

Specifies the alternate name for the network service. You can specify a maximum of four aliases. No checking is done to ensure that an alias is unique.

Single values

***NONE**
The service has no alternate name.

Other values (up to 4 repetitions)

character-value
Specify an alternate service name.

Top

Examples

```
ADDSRVTBLE   SERVICE(FTP)  PORT(21)  PROTOCOL(TCP)
```

This command adds a service entry to the service table for the FTP network service. The service uses port 21 and the TCP protocol.

Top

Error messages

***ESCAPE Messages**

TCP290A
Service entry already exists in table. Entry was not added.

TCP2914
Service entry contains characters that are not valid. Entry was not added.

TCP8050
*IOSYSCFG authority required to use &1.

Top

Add Server Auth Entry (ADDSVRAUTE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Server Authentication Entry (ADDSVRAUTE) command adds authentication information for use by application requesters in connecting to application servers.

When using the ADDSVRAUTE command to add a server authorization entry for a Distributed Relational Database Architecture (DRDA) application that uses TCP/IP, make sure that the server name is entered in upper case.

Restrictions: You must have security administrator (*SECADM) special authority, and object management (*OBJMGT) and use (*USE) authorities to the user profile to which the server authentication entry is to be added, or else be signed on under that user profile, to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
USRPRF	User profile	*Simple name*, *CURRENT	Required, Positional 1
SERVER	Server	*Character value*	Required, Positional 2
USRID	User ID	*Character value*, ***USRPRF**	Optional
PASSWORD	User password	*Character value*, ***NONE**	Optional

Top

User profile (USRPRF)

Specifies the user profile for which the server authentication entry is to be added.

This is a required parameter.

***CURRENT**
The server authentication entry will be added for the current user.

name Specify the name of the user for which to add the server authentication entry.

Top

Server (SERVER)

Specifies the name of the application server for which the entry is to be added.

This is a required parameter.

name Specify the name for the particular application server for which the entry is to be added. Specify no more than 200 characters.

Note: Refer to the documentation for the server that you are using to determine if there are any values that have special meaning. For example, the server name QDDMSERVER has special meaning if you are using the Distributed Data Management (DDM) server.

Top

User ID (USRID)

Specifies the user name for which requests will be made to the application server.

***USRPRF**
The name specified in the user profile parameter will be the user ID specified on connection requests to the server.

'name' Specify the name of the user to be used on connection requests. Specify no more than 1000 characters.

Top

User password (PASSWORD)

Specifies the password to be used to authenticate the user when the client attempts to connect to the server.

Note: If the retain server security data (QRETSVRSEC) system value is set to 0 (do not retain data), then the password is not saved in the entry.

***NONE**
No password is supplied on the connection request.

'password'
Specify the password associated with the user ID. Specify no more than 696 characters.

Top

Examples

Example 1: Adding a default remote user ID and password for the current user

```
ADDSVRAUTE   USRPRF(*CURRENT)  SERVER(*ANY)  USRID('JOHN')
             PASSWORD('XU53W4')
```

This command adds a server authentication entry for the currently signed on user specifying that for connection requests to any server for which there is no specific authentication entry, a remote user ID of JOHN and a password of XU53W4 is to be used.

Example 2: Adding an entry for another user for a specific server

```
ADDSVRAUTE   USRPRF(SUSAN)  SERVER('MPLS_RDB')  USRID('SUSIE')
             PASSWORD('S23084')
```

This command adds an entry such that when a user is signed on to the local system under the user profile of SUSAN and attempts to connect to the server named MPLS_RDB, the user ID and password accompanying the connection request will be SUSIE and S23084.

Top

Error messages

***ESCAPE Messages**

CPF2204
User profile &1 not found.

CPF2213
Not able to allocate user profile &1.

CPF2222
Storage limit is greater than specified for user profile &1.

CPF224F
Server authentication entry already exists.

CPF225F
Not all information stored.

CPF226C
Not authorized to perform function.

Top

Add Tape Cartridge (ADDTAPCTG)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Tape Cartridge (ADDTAPCTG) command adds the specified cartridge identifiers to a usable category. Cartridges are placed in the insert category when they are placed in the library device and must be added to a usable category before they can be used by a tape device.

The cartridge identifier must be unique within a library device. If a duplicate cartridge identifier does exist in a library device, both cartridges are unusable until one is physically removed from the library device.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Library device	*Name*	Required, Positional 1
CTG	Cartridge ID	Values (up to 40 repetitions): *Character value*	Required, Positional 2
CGY	Category	Single values: *SHARE400 Other values: *Element list*	Optional
	Element 1: Category name	*Character value*, ***NOSHARE**, *IPL, *NL, *CNV	
	Element 2: Category system	*Character value*, ***CURRENT**	
CHKVOL	Check volume identifier	***YES**, *NO	Optional

Top

Library device (DEV)

Specifies the library device to be used. The device name must have been created previously on the system using the Create Device Media Library (CRTDEVMLB) command.

This is a required parameter.

name Specify the name of the library device.

Top

Cartridge ID (CTG)

Specifies a maximum of 40 cartridge identifiers that are currently in the insert category that are added to the category specified.

This is a required parameter.

character-value
Specify the cartridge identifier to be added. Each cartridge identifier can be a maximum of 6 characters.

Note: The cartridge identifier should represent the external identifier if the library device has a bar code scanner to read external identifiers.

Top

Category (CGY)

Specifies the category to which the tape cartridge is added. The cartridge is moved to a slot in the library device, unless the cartridge is added to the convenience (CNV) category.

Single values

***SHARE400**
The cartridge identifier specified can be shared with other systems that are attached to the same device.

Element 1: Category name

***NOSHARE**
The cartridge identifier specified cannot be shared with other systems that are attached to the same device.

***IPL** The cartridge identifier specified can be used for an alternate initial program load (IPL) of a system. The management of the cartridges in the category must be done by the user.

```
**** Attention ********************************
When using the *IPL category for an alternate
IPL of a system, you must ensure that the
cartridges are the proper ones to be used for the
alternate IPL.  Conflicts may arise if high-end
and low-end systems are attached to the same
library device.
************************************************
```

***NL** The cartridge is used as a non-labeled tape.

***CNV** The cartridge identifier specified is added to the special convenience category. It is not moved to a slot in the library device. When the cartridge is unloaded from a device, it is removed (exported) to the convenience station.

character-value
Specify the name of a user-defined category. This category name must have been created previously with the Create Tape Category (CRTTAPCGY) command.

Element 2: Category system

This element identifies the system to which the category belongs. The system name is obtained from the pending system name field of a Display Network Attributes (DSPNETA) command. If there is no pending system name, the current system name attribute is used.

```
***** Attention *********************************
If the system name is changed, the category
information associated with all tape cartridges
in library devices is not valid.
************************************************
```

***CURRENT**
The category belongs to the system currently running the command.

character-value
Specify the name of the system to which the category belongs.

Top

Check volume identifier (CHKVOL)

Specifies whether the logical volume identifier is forced to be identical to the external identifier if the library device has a bar code scanner to read the bar code identifier.

***YES** The tape cartridge is verified for the correct logical volume identifier by reading the volume label existing on the tape cartridge.

***NO** The tape cartridge is not verified for the correct logical volume identifier. If a cartridge is added in which the external identifier does not match the logical volume identifier, the cartridge is valid for read-only operations. Output operations to the tape cartridge are not allowed unless the logical volume identifier is initialized to match the external identifier. If the tape is a non-labeled tape, this match is not enforced because there is no logical volume identifier.

Top

Examples

Example 1: Adding a Single Cartridge to the *SHARE400 Category

```
ADDTAPCTG   DEV(LIB01)  CTG(VOL4)  CGY(*SHARE400)  CHKVOL(*NO)
```

This command adds the cartridge identifier VOL4 to the usable category *SHARE400. The logical volume identifier in the volume labels of VOL4 are not verified.

Example 2: Adding Multiple Cartridges to the *NOSHARE Category

```
ADDTAPCTG   DEV(LIB01)  CTG(VOL1 VOL2 VOL3)
            CGY(*NOSHARE)  CHKVOL(*YES)
```

This command adds the cartridge identifiers VOL1, VOL2, and VOL3 to the usable category *NOSHARE. The logical volume identifiers on the tape cartridges are verified when the command is run.

Top

Error messages

***ESCAPE Messages**

CPF67AB
&6 cartridges not added

CPF67A6
Category does not exist

CPF67D2
Cartridge command was not successful.

CPF67E4
Library device function not successful

CPF67EA
Function not successful

CPD67EB
Cartridge &2 not in *INSERT category

CPF67EC
Library device description &1 does not exist

CPF67ED
Library device &1 not available

CPF67F5
Duplicate cartridge or virtual volume name found

CPF6708
Command ended due to error.

CPF6718
Cannot allocate device &1.

CPF6745
Device &1 not a media library device.

CPF9814
Device &1 not found.

CPF9825
Not authorized to device &1.

Top

Add TCP/IP Host Table Entry (ADDTCPHTE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add TCP/IP Host Table Entry (ADDTCPHTE) command adds an internet address and its associated host names along with an optional text description field to the local host table. A host table entry consists of one internet address, up to 65 host names, and a text description field.

See also the following host table commands:

- Change TCP/IP Host Table Entry (CHGTCPHTE) command changes one or more host names or the text description field.
- Copy TCP/IP Host Table (CPYTCPHT) command copies the contents of the local host table to a physical file member.
- Merge TCP/IP Host Table (MRGTCPHT) command merges host names, internet addresses, and text comment entries from a physical file member into the local host table. A replace option is also provided that allows the entire local host table to be replaced by the host table entries in a user specified physical file member.
- Rename TCP/IP Host Table Entry (RNMTCPHTE) command renames the internet address of a host table entry to another internet address.
- Remove TCP/IP Host Table Entry (RMVTCPHTE) command removes an entire entry from the local host table.

If an internet address already exists in the host table that matches the internet address specified in the command, an escape message is sent to the user and the duplicate internet address is not added.

If a remote name server is being used by your system for resolving a host name or an internet address, the choice to first search the remote name server or the local host table depends on the setting of the host name search priority (HOSTSCHPTY) parameter on the Change TCP/IP Domain (CHGTCPDMN) command. To change the host name search priority use the CHGTCPDMN command or use the Configure TCP/IP (CFGTCP) command and select option 12.

The TCP/IP host table is shipped with two loopback entries — one for IPv4 and one for IPv6. The IPv4 entry has an internet address of 127.0.0.1 and two host names: LOOPBACK and LOCALHOST. The LOOPBACK host name can only be associated with an internet address that has a first-byte value equal to 127. The IPv6 entry has an internet address of ::1 and two host names: IPV6-LOOPBACK and IPV6-LOCALHOST. The IPV6-LOOPBACK host name can only be associated with the IPv6 internet address ::1. The IPV6-LOCALHOST host name can only be associated with an IPv6 internet address.

Warning: Temporary Level 2 Header

Warning: Temporary Level 3 Header

Related APPC over TCP/IP Information

APPC over TCP/IP (AnyNet) uses the host name to map location names to internet addresses. The host name must be in the form:

```
location.netid.SNA.IBM.COM
```

Where *location* is the remote location the program is opening to, and *netid* is the network identifier for this connection. *SNA.IBM.COM* is the qualifier that designates this as the APPC over TCP/IP domain.

Location names support characters that **cannot** be present in host names (for example: $ (dollar), @ (at sign), and # (number sign)). Therefore, the APPC application can open only to locations that fulfill the TCP/IP host name syntax. This limits location names used for APPC over TCP/IP to the characters A-Z (uppercase and lowercase) and 0-9.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
INTNETADR	Internet address	*Character value*	Required, Positional 1
HOSTNAME	Host names	Values (up to 65 repetitions): *Element list*	Required, Positional 2
	Element 1: Name	*Character value*	
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Top

Internet address (INTNETADR)

Specifies the internet address that the host names and text descriptions are associated with.

An IPv4 internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An IPv4 internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address.

An IPv6 internet address is specified in the form *x:x:x:x:x:x:x:x*, where *x* is a hexadecimal number ranging from 0 through X'FFFF'. "::" may be used once in the IPv6 address to indicate one or more groups of 16 bits of zeros. The "::" may be used to compress leading, imbedded, or trailing zeros in the address.

An IPv4-mapped IPv6 address may be specified. An IPv4-mapped IPv6 address is specified in the form *::FFFF:nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255.

If the internet address is entered from a command line, the address must be enclosed in apostrophes.

This is a required parameter.

character-value
Specify the internet address.

Top

Host names (HOSTNAME)

Specifies the host names corresponding to the internet address. The host name can be either the short form or the full domain version of the name. A common practice is to define one short name that is unique within your local network and to also define the full domain version of the host name that is unique within the Internet.

A host name is a text string that has 1 to 255 characters. Host names consist of one or more labels separated by periods. Each label can contain up to 63 characters. The first character of each label must be an alphabetical character or a digit. The last character of each label must be an alphabetical character, a digit, or a period. The following characters are allowed in host names:

- Alphabetical characters A through Z
- Digits 0 through 9
- Underscore (_)
- Minus sign (-)
- Period (.). Periods are allowed only when they separate host name labels or as the last character in the host name (refer to RFC 1034). A host name cannot have two consecutive periods.

Note: These characters are part of the Syntactic/Invariant Character Set (character set number 640).

Other domain name and host name conventions include the following:

- Uppercase and lowercase characters are allowed, but no significance is attached to the case. The host name (HOSTNAME) may be converted to uppercase depending on the combination of characters and digits. If the HOSTNAME is enclosed in apostrophes (’), the case is maintained as entered.
- The host name returned when searching the host table for an internet address is the first host name associated with the internet address. For example, if the address 9.130.38.187 is defined in the host table with names ROCHESTER, JOHN, and RCHAS100, the name ROCHESTER would be returned. The other two host names would not be used in this type of search. However, these host names would be used when searching the host table to find the internet address associated with the names JOHN and RCHAS100.
- Try to limit your domain name labels to 12 characters. Shorter labels are easier to remember.
- It is a common practice to use hierarchical names that allow predictable extensions for change and growth. Domain names normally reflect the delegation of authority or hierarchy used to assign them. For example, the name SYS1.MFG.ABC.COM can be broken down into the following:

  COM All commercial networks.

  ABC.COM
  All systems in the ABC company’s commercial network.

  MFG.ABC.COM
  All manufacturing systems in the ABC company’s commercial network.

  SYS1.MFG.ABC.COM
  A host named SYS1 in the manufacturing area of the company’s commercial network.

  The COM designation is one of several domain names used by convention when connecting to the Internet. Some of the other domain names that follow this convention are:

  AERO Air-transport industry

  BIZ Businesses

  CAT Catalan linguistic and cultural community

  COM Commercial organizations

  COOP Cooperative associations

EDU Educational institutions

GOV United States of America government institutions

INFO Global information

INT Organizations established by international treaties between governments

JOBS Human resource managers

MIL United States of America military groups

MOBI Consumers and providers of mobile products and services

MUSEUM
Museums

NAME
Individuals

NET Major networks support centers

PRO Credentialed professionals and related entities

TRAVEL
Travel industry

ORG Organizations other than those above

ARPA Temporary ARPANET domain

Country or Region Code
Countries or regions other than the USA

This is a required parameter.

You can specify 65 values for this parameter.

character-value
Specify a host name to be associated with the specified internet address. When running APPC over TCP/IP, the host name is in the form:

```
location.netid.SNA.IBM.COM
```

The default if a host name is not specified is blanks. An IP address cannot be a host name.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the host table entry.

***BLANK**
No text is specified.

character-value
Specify no more than 64 characters of text, enclosed in apostrophes.

Top

Examples

Example 1: Adding a Short Host Name

```
ADDTCPHTE   INTNETADR('132.28.71.5')  HOSTNAME(ETHERNETHOST)
            TEXT('ETHERNETHOST on Ethernet subnet')
```

This command associates the host name ETHERNETHOST with the internet address of 132.28.71.5. The text 'ETHTERNETHOST on Ethernet subnet' is saved as the text description for this host table entry.

Example 2: Adding Five Host Names

```
ADDTCPHTE   INTNETADR('1234::5678')
            HOSTNAME((SYSTEM1.SALES.ABC.COM)
                     ('system1.sales.abc')
                     ('System1.Distribution.Abc.Com')
                     ('System1.Marketing.Abc.Com')
                     ('System1.Marketing.Abc'))
            TEXT('Entry verified by J. Jones')
```

This command associates the host names SYSTEM1.SALES.ABC.COM, SYSTEM1.SALES.ABC, SYSTEM1.DISTRIBUTION.ABC.COM, SYSTEM1.MARKETING.ABC.COM, and SYSTEM1.MARKETING.ABC with the internet address of 1234::5678. Because host names are case insensitive, a match is found on host name SYSTEM1.SALES.ABC.COM, system1.sales.abc, System1.Distribution.Abc.Com, System1.Marketing.Abc.Com, or System1.Marketing.Abc. The text 'Entry verified by J. Jones' is saved as the text description for this host table entry.

Top

Error messages

***ESCAPE Messages**

TCP1901
Internet address &2 not valid.

TCP1902
Internet address &1 not valid.

TCP1903
Specified host name not valid.

TCP1904
Duplicate internet address &2 found in host table.

TCP1908
Internet address &1 not valid.

TCP1910
LOOPBACK internet address &2 not valid.

TCP1929
Host table not available.

TCP8050
*IOSYSCFG authority required to use &1.

TCP9999
Internal system error in program &1.

Top

Add TCP/IP Interface (ADDTCPIFC)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add TCP/IP Interface (ADDTCPIFC) command is used to add a new interface to the Transmission Control Protocol/Internet Protocol (TCP/IP) configuration. The interfaces added by the ADDTCPIFC command are logical interfaces. They are not physical interfaces. Each interface is associated with a line description. The line description is the physical connection from the system to the TCP/IP network.

The i5/OS TCP/IP implementation supports *multihoming*. This allows either a single interface or multiple interfaces to be specified per line description. The system can appear as any one or combination of the following:

- A single host on a network over a communications line
- Multiple hosts on the same network over the same communications line
- Multiple hosts on different networks over the same communications line
- Multiple hosts on the same network over multiple communications lines
- Multiple hosts on different networks over multiple communications lines

Notes:

1. In SNMP, an interface is a physical interface. The physical interface relates directly to an input/output processor (IOP).
2. The interface table is shipped with a default IPv4 interface of 127.0.0.1. The line description value associated with the 127.0.0.1 interface is *LOOPBACK. The host table is also shipped with an entry that has an internet address of 127.0.0.1 and host names of LOOPBACK and LOCALHOST.
3. The interface table is shipped with a default IPv6 interface of ::1. The line description value associated with the ::1 interface is *LOOPBACK. The host table is also shipped with an entry of ::1 and host names of IPV6-LOOPBACK and IPV6-LOCALHOST.
4. A maximum of 16K interfaces can be added and up to 2K interfaces can be active on any one line with the Start TCP/IP Interface (STRTCPIFC) command at any given time. This is true for all line types (for example; Ethernet, token-ring, frame relay, and so forth).

Attention: Before attempting to start an X.25 interface, ensure that the remote system information (RSI) for non-DDN X.25 interfaces that use a permanent virtual circuit (PVC) is configured. Use the Add TCP/IP Remote System (ADDTCPRSI) command to do this. Incoming data from a remote system on the X.25 network is not processed unless an RSI entry for the PVC is configured on the X.25 interface before the interface is started.

Restrictions:

- Input/output system configuration (*IOSYSCFG) special authority is necessary to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
INTNETADR	Internet address	*Character value*, *IP6SAC	Required, Positional 1

Keyword	Description	Choices	Notes
LIND	Line description	*Name*, *LOOPBACK, *VIRTUALIP, *OPC	Required, Positional 2
SUBNETMASK	Subnet mask	*Character value*, *HOST	Optional, Positional 3
ADRPFXLEN	Address prefix length	1-128, **64**	Optional
ALIASNAME	Alias name	*Simple name*, ***NONE**	Optional
LCLIFC	Associated local interface	*Character value*, ***NONE**	Optional
TOS	Type of service	*MINDELAY, *MAXTHRPUT, *MAXRLB, *MINCOST, ***NORMAL**	Optional
MTU	Maximum transmission unit	576-16388, ***LIND**	Optional
AUTOSTART	Autostart	***YES**, *NO	Optional
PVCLGLCHLI	PVC logical channel identifier	Values (up to 64 repetitions): *Character value*, ***NONE**	Optional
IDLVCTTIMO	X.25 idle circuit timeout	1-600, **60**	Optional
MAXSVC	X.25 maximum virtual circuits	0-64, **64**	Optional
DDN	X.25 DDN interface	*YES, ***NO**	Optional
BITSEQ	TRLAN bit sequencing	***MSB**, *LSB	Optional
IFCID	Interface ID	0000000000000001-FFFFFFFFFFFFFFEF, ***LIND**, *GEN	Optional
DADMAXTRN	DAD maximum transmits	0-10, **1**	Optional
PVYEXN	Privacy extensions	***NO**, *YES	Optional
PREFIFC	Preferred interfaces	Single values: ***NONE**, *AUTO Other values (up to 10 repetitions): *Character value*	Optional
PREFLIND	Preferred line descriptions	Single values: ***NONE** Other values (up to 10 repetitions): *Name*	Optional
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Top

Internet address (INTNETADR)

Specifies an internet address that the local system responds to on this interface. The internet address may be an IPv4 or IPv6 address. An interface is associated with a line description.

This is a required parameter.

***IP4DHCP**
Enable Dynamic Host Configuration Protocol for the line description specified (LIND parameter).

***IP6SAC**
Enable IPv6 stateless address auto-configuration for the line description specified (LIND parameter).

character-value
Specify a particular internet address.

An IPv4 internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An IPv4 internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address.

An IPv6 internet address is specified in the form *x:x:x:x:x:x:x:x*, where *x* is a hexadecimal number ranging from 0 through X'FFFF'. "::" may be used once in the IPv6 address to indicate one or more groups of 16 bits of zeros. The "::" may be used to compress leading, imbedded, or trailing zeros in the address.

An IPv6 internet address must be a unicast address and must not contain an imbedded IPv4 address (compatibility or mapped). If *VIRTUALIP is specified for LIND, the IPv6 address must be a global unicast address.

If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Top

Line description (LIND)

Specifies the name of the line description associated with the new interface. The line description must exist before the TCP/IP interface can be added.

The following conditions are based on the type of line description:

Token-ring
The line description must be previously created with the Create Line Description (Token-Ring Network) (CRTLINTRN) command.

X.25 The line description must be previously created with the Create Line Description (X.25) (CRTLINX25) command.

Ethernet
The line description must be previously created with the Create Line Description (Ethernet) (CRTLINETH) command.

DDI The line description must be previously created with the Create Line Description (DDI Network) (CRTLINDDI) command.

Frame relay
The line description must be previously created with the Create Line Description (Frame Relay Network) (CRTLINFR) command.

Wireless
The line description must be previously created with the Create Line Description (Wireless Network) (CRTLINWLS) command.

Twinax (TDLC)
The line description must be previously created with the Create Line Description (TDLC) (CRTLINTDLC) command.

TCP/IP can also be used on certain line descriptions attached to these network interfaces (NWI):

- A frame relay NWI using a frame relay, token ring, Ethernet, or DDI line description.
  - The frame relay NWI is created using the Create Network Interface Frame Relay Network (CRTNWIFR) command.
  - The line description is created using the appropriate Create Line command and attached to the frame relay NWI by specifying the NWI and NWIDLCI parameters.

This is a required parameter.

***LOOPBACK**
The interface being changed is the loopback or LOCALHOST interface. Because processing

associated with loopback does not extend to a physical line, there is no line description associated with a loopback address. This special value must be used for any INTNETADR that has a first octet value of 127.

***VIRTUALIP**
The virtual interface is a circuitless interface. It is used in conjunction with the associated local interface (LCLIFC) when adding standard interfaces. This special value is used to accommodate any of the following cases:

1. Load balancing. This is the means of having a fixed source IP address regardless of which interface the traffic is being distributed.
2. Frame-relay multi-access network to define the local network IP address. This allows for multiple virtual circuits to share the same IP network.
3. Alternate method of network access translation (NAT). This eliminates the need for a NAT box by assigning a globally unique single IP address directly to the box without the need to define an entire network.
4. Unnumbered networks. This provides a means of associating a local source IP address for an unnumbered point-to-point network.

***OPC** This special value is used if a TCP/IP over OptiConnect interface is being added. This interface is attached to the optical bus (OptiConnect). INTNETADR must specify an IPv4 address.

name Specify the name of the line description to be used for this interface. If INTNETADR specifies an IPv6 address, or *IP6SAC, the line description must be for an Ethernet line. A line description name is required if INTNETADR is *IP4DHCP.

Top

Subnet mask (SUBNETMASK)

Specifies the subnet mask, which is a bit mask that defines the part of the network where this IPv4 interface attaches. The mask is a 32-bit combination that is logically ANDed with the IPv4 internet address to determine a particular subnetwork. The bits of the mask set to the value one (1) determine the network and subnetwork portions of the address. The bits set to the value zero (0) determine the host portion of the address.

Note: The network portion must be equal to one bits in the subnetmask. The host portion of an address must be at least two bits wide. This parameter is ignored if INTNETADR specifies an IPv6 address, *IP4DHCP, or *IP6SAC.

***HOST**
The subnet mask value used will be 255.255.255.255. Specify this value for use with Proxy ARP (Address Resolution Protocol).

character-value
Specify the mask for the network subnet field and host address field of the internet address that defines a subnetwork. The subnet mask is in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. The subnet mask must mask off all bits of the network class's network ID portion of the internet address. For example, 255.255.255.0 could define a subnet mask for an interface with a Class B internet address. In this example, the first two octets must be 1 bits because these octets define the network ID portion of the Class B internet address. The third octet of this subnet mask defines the actual subnet mask ID portion of the interface's internet address. It is also all 1 bits. This leaves the fourth octet to define the host ID portion of the interface's internet address.

Note: The bits that identify the subnetwork are not required to be adjacent in the address. However, it is strongly advised that the subnet bits be contiguous and located in the most significant bits of the host address.

Top

Address prefix length (ADRPFXLEN)

Specifies the address prefix length in bits. The address prefix length specifies how much of the leftmost portion of an IPv6 address is the subnet prefix. This parameter applies only if INTNETADR is an IPv6 address.

64 The default is 64 since most IPv6 unicast addresses have a 64 bit interface ID (address suffix).

1-128 Specify the address prefix length.

Top

Alias name (ALIASNAME)

Specifies a name that can be used in place of the internet address. The name must be unique among all of the interfaces on the system.

***NONE**
No alias name is associated with the TCP/IP interface being added.

simple-name
Specify an alias for the interface. The alias name must be 25 characters or less.

Top

Associated local interface (LCLIFC)

Use this parameter to associate the IPv4 interface being added with an existing local IPv4 TCP/IP interface. This parameter is ignored if INTNETADR specifies an IPv6 address, *IP4DHCP, or *IP6SAC.

The associated local interface (LCLIFC) is used to allow 'transparent subnetting' (also known as 'Proxy Arp') between the associated interfaces, to define unnumbered networks, or for load balancing.

Conditions for using the LCLIFC for unnumbered networks:

- The line type of the interface being added MUST be Frame Relay, Opti-Connect, or Point-to-Point and the subnet mask (SUBNETMASK) parameter must be *HOST (255.255.255.255).

Conditions for using LCLIFC for transparent subnetting:

- The network of the associated local interface must be broadcast capable.
- The interface being added must be defined as a subnet of the network with which it is associated (using LCLIFC).

Condition for using the LCLIFC for load balancing:

- This is the means of having a fixed source IP address regardless of which interface the traffic is being distributed.
- The line type of the associated local interface must be *VIRTUALIP.

Note: LCLIFC can only be used to associate this interface with another interface that is already added. Once associated, the interface specified in LCLIFC and this interface must both be started in order for them to work together properly.

***NONE**
No TCP/IP interface is associated with the interface being added.

character-value
Specify the internet address of the interface you want to associate with the interface being added.

Top

Type of service (TOS)

Specifies the type of service to be used. The type of service specifies how the internet hosts and routers should make trade-offs between throughput, delay, reliability, and cost. This parameter is ignored if INTNETADR specifies an IPv6 address, *IP4DHCP, or *IP6SAC.

***NORMAL**
Normal service is used for delivery of data.

***MINDELAY**
Minimize delay means that prompt delivery is important for data on this connection.

***MAXTHRPUT**
Maximize throughput means that a high data rate is important for data on this connection.

***MAXRLB**
Maximize reliability means that a higher level of effort to ensure delivery is important for data on this connection.

***MINCOST**
Minimize monetary cost means that lower cost is important for data on this connection.

Top

Maximum transmission unit (MTU)

Specifies the maximum size (in bytes) of IP datagrams that can be transmitted through this interface. A datagram is a basic unit of information passed over an internet network. For an IPv4 interface, the minimum MTU value is 576 bytes. For an IPv6 interface, the minimum MTU value is 1280 bytes.

***LIND**
The MTU is determined by the information specified in the line description. If *LIND is specified, the MTU is equal to the largest amount of data that can be transmitted on the line.

576-16388
Specify a value for the maximum transmission unit in bytes. The maximum MTU that can be specified for this interface depends on the type of physical connection to the network. The following table lists the maximum MTU values that can be specified based on the line type:

X.25 4096

Token ring (4 meg)
4060

Token ring (16 meg)
16388

Ethernet 802.3
8992

Ethernet Version 2
9000

DDI 4352

Frame relay
8177

Wireless 802.3
1492

Wireless Version 2
1500

Twinax (TDLC)
4105

Notes:

1. The actual MTU value used for an interface is resolved during interface activation. This value is the minimum of either the specified MTU value for the interface or the largest amount of data that can be transmitted on the line.
2. It is suggested (not required) that the same MTU values be used for all interfaces on the same network. However, all interfaces must have an MTU that does not exceed the value used when *LIND is specified for the interface MTU.
3. To view the MTU value actually used for an interface, do the following:
   a. Use the ADDTCPIFC command to add the interface.
   b. Use the Start TCP/IP Interface (STRTCPIFC) command to activate the interface.
   c. Use the Work with TCP/IP Status (WRKTCPSTS or NETSTAT) command to view the actual MTU value of the interface in bytes.

Top

Autostart (AUTOSTART)

Specifies whether the interface is automatically started when the TCP/IP stack is activated by the Start TCP/IP (STRTCP) command.

***YES** The interface is automatically started when TCP/IP is started.

***NO** The interface is not started when TCP/IP is started.

Note: The Start TCP/IP Interface (STRTCPIFC) command can be used to start an interface after TCP/IP has been started.

Top

PVC logical channel identifier (PVCLGLCHLI)

Specifies the permanent virtual circuit (PVC) logical channel identifiers that can be established on an X.25 interface by the TCP/IP protocol stack. Up to 64 unique channel identifiers may be specified. These logical channel identifiers must be specified in the X.25 line description.

With this parameter the line can be shared with other communications software, such as Systems Network Architecture (SNA). It prevents the TCP/IP protocol stack from monopolizing the PVCs defined for the line.

Notes:

1. This parameter is valid only for an interface defined on an X.25 line description.
2. PVCs cannot be used in a DDN network.

3. When specifying PVCs for an X.25 interface, all interfaces on the same X.25 network should have the same set of PVC logical channel identifiers specified. This is especially important if one or more remote system information (RSI) entries will use a PVC to connect to the RSI entry's remote system on the X.25 network.
4. If the RSI entries are defined such that two or more remote internet addresses can be reached across the same PVC, that PVC is shared.
5. The sum of the maximum switched virtual circuits (MAXSVC) and the number of PVCs cannot exceed 64.

***NONE**
No PVC logical channel identifier values are specified.

character-value
Specify the PVC logical channel identifier value. The value may be from 001 to FFF. Up to 64 PVC logical channel identifiers can be specified.

Top

X.25 idle circuit timeout (IDLVCTTIMO)

Specifies the duration (in seconds) that TCP/IP waits before clearing an idle virtual circuit established on an X.25 link. Clearing an idle virtual circuit frees resources on the network. TCP/IP automatically reestablishes virtual circuits when required to send or receive data. Virtual circuits are transparent to a TCP/IP client and have no noticeable effect on TCP connections.

Note: This parameter is valid only for switched virtual circuits (SVCs) on an interface defined on an X.25 line description. It is not valid for permanent virtual circuits (PVCs).

60 The idle virtual circuit timeout is 60 seconds.

1-600 Specify the number of seconds to be used for the idle virtual circuit timeout.

Top

X.25 maximum virtual circuits (MAXSVC)

Specifies the maximum number of concurrent switched virtual circuits (SVC) that can be established on an X.25 interface by the TCP/IP protocol stack.

With this parameter the line can be shared with other communications software such as Systems Network Architecture (SNA). It prevents the TCP/IP protocol stack from monopolizing the SVCs defined for the line. This parameter is valid only for an interface defined on an X.25 line description.

Note: The sum of the maximum switched virtual circuits (MAXSVC) and the number of PVCs cannot exceed 64.

64 If 64 is specified, the number of SVCs that are configured is the sum of the number of *SVCIN, *SVCOUT and *SVCBOTH SVCs defined for the line description (LIND) being used by this interface. This is the maximum number of SVCs that can be authorized for processing by the TCP/IP protocol stack.

0-64 Specify the number of SVCs that TCP/IP protocol stack can use simultaneously.

Top

X.25 DDN interface (DDN)

Specifies whether the X.25 interface is connected to the Defense Data Network (DDN). The DDN network is a special type of X.25 network used by TCP/IP customers with special security needs.

Note: This parameter is valid only for switched virtual circuits (SVCs) on an interface defined on an X.25 line description. It is not valid for permanent virtual circuits (PVCs).

Warning:

If multiple interfaces are specified to the same X.25 network, the DDN value should be equal for all of those interfaces. This is not enforced by the ADDTCPIFC or CHGTCPIFC commands.

If the X.25 network is on the DDN network, do not define the remote system information for any of the remote systems on the network. The remote system information for the DDN X.25 network is determined from the destination IP address.

***NO** The X.25 interface is not connected to the Defense Data Network.

***YES** The X.25 interface is connected to the Defense Data Network.

Top

TRLAN bit sequencing (BITSEQ)

Specifies the order, most or least significant bit first, in which the Address Resolution Protocol (ARP) places the bits in the hardware address. This parameter is valid only for a token-ring local area network (TRLAN) line.

Note: All interfaces defined to a single token-ring line must have the same BITSEQ value. This is checked to ensure consistent values.

***MSB** The most significant bit is placed first.

***LSB** The least significant bit is placed first.

Top

Interface ID (IFCID)

Specifies the interface ID portion of the IPv6 address(es) generated when INTNETADR is *IP6SAC. The interface ID is the low-order 64 bits of an IPv6 address created as a result of IPv6 stateless address auto-configuration.

***LIND**
Use the local adapter address in the associated line description to generate the interface ID.

***GEN** Generate a random local scope interface ID.

0000000000000001-FFFFFFFFFFFFFFEF
Specify the interface ID to be used.

Top

DAD maximum transmits (DADMAXTRN)

Specifies the maximum number of neighbor solicitation messages to send for IPv6 duplicate address detection (DAD). This parameter applies only if INTNETADR is an IPv6 address or *IP6SAC.

1 The recommended default value.

0-10 Specify the number of neighbor solicitation messages to be sent. A value of zero indicates that duplicate address detection is not performed.

Top

Privacy extensions (PVYEXN)

Specifies whether privacy extensions are to be used for IPv6 stateless address auto-configuration. This parameter applies only when INTNETADR is *IP6SAC.

Note: Privacy extensions are not supported for communications resource types 2838, 2849, and 287F. If the line description associated with this interface specifies a resource name that is associated with one of these resource types, PVYEXN must be *NO.

***NO** Do not use privacy extensions.

***YES** Use privacy extensions.

Top

Preferred interfaces (PREFIFC)

Specifies a list of preferred IPv4 interfaces that are to be used with the IPv4 interface being added for proxy Address Resolution Protocol (ARP) agent selection. The IPv4 interface being added must specify a LIND value of *VIRTUALIP or specify a LIND for a virtual Ethernet line. Up to 10 associated interfaces may be specified in order of preference — the first being the most preferred, the second the next preferred, etc. Each associated interface must be a standard IPv4 interface, not a virtual interface.

Single values

***NONE**
No associated interfaces are specified.

***AUTO**
Interface selection is performed automatically by the system.

Other values (up to 10 repetitions)

character-value
Specify the IPv4 address of the preferred interface.

Top

Preferred line descriptions (PREFLIND)

Specifies a list of preferred line descriptions that are to be used with the IPv6 interface being added for virtual IP address (VIPA) proxy Neighbor Discovery agent selection. The IPv6 interface being added must have a LIND of *VIRTUALIP. Up to 10 line descriptions may be specified in order of preference — the first being the most preferred, the second the next preferred, etc. Each line description must be used by at least one IPv6 interface.

Single values

***NONE**
No line descriptions are specified.

Other values (up to 10 repetitions)

name Specify the name of the preferred line description.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the interface.

***BLANK**
No text is specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Examples

Example 1: Adding a Non-AUTOSTART Interface

```
ADDTCPIFC   INTNETADR('130.14.3.5')  LIND(COTTAGELAN)
            AUTOSTART(*NO)  SUBNETMASK('255.255.255.0')
```

Let's assume that an Ethernet line has been created named COTTAGELAN using the CRTLINETH command, this command adds the interface 130.14.3.5 to the TCP/IP configuration. This interface uses the line description named COTTAGELAN. It is not automatically started when the STRTCP command is run. This interface must be started using the Start TCP/IP Interface (STRTCPIFC) command. The STRTCPIFC can be issued either directly from a command line or by using option 9 from either of the following lists:

- The Work with TCP/IP Interface Status list. Use menu option 1 from the menu displayed when the Work With TCP/IP Status (WRKTCPSTS) command is issued to display this list.
- The Work with TCP/IP Interfaces list. Use menu option 1 from the menu displayed when the Configure TCP/IP (CFGTCP) command is issued to display this list.

Example 2: Adding an AUTOSTART Interface

```
ADDTCPIFC   INTNETADR('8.77.0.21')  LIND(COTTAGEX25)
            IDLVCTTIMO(45)  MAXSVC(15)  DDN(*YES)
            SUBNETMASK('255.255.255.0') AUTOSTART(*YES)
```

Let's assume that an X.25 line has been created named COTTAGEX25 using the CRTLINX25 command, this command adds interface 8.77.0.21 to the TCP/IP configuration. This interface uses the line description named COTTAGEX25. When TCP/IP is started using the Start TCP/IP (STRTCP) command, the interface is automatically started. The idle virtual circuit timeout is 45 seconds. The maximum number of concurrent SVCs allowed to be used by TCP/IP on this interface is 15. This interface is connected to the Defense Data Network. You do not need to define any remote system information (RSI) entries for this X.25 network because it is a DDN network.

Example 3: Adding an Interface for a Twinax Line that is using an Associated Local Interface

```
ADDTCPIFC   INTNETADR('199.1.1.99')  LIND(TDLCLINE)
            SUBNETMASK(255.255.255.0)  LCLIFC('199.1.1.1')
```

This command will add a TCP/IP interface for the twinax line named TDLCLINE. This interface will be associated with local interface 199.1.1.1. This means that the devices attached to twinax line 199.1.1.99 can take advantage of 'appearing' to be on the same network as the local 199.1.1.1 interface (transparent subnetting). No special routing is required to ensure packets from the twinax connnected hosts can travel to the local 199.1.1.0 network. Also, hosts on the 199.1.1.0 network can also reach the twinax hosts without any additional routing on the host systems.

Example 4: Adding an Interface with an Alias

```
ADDTCPIFC   INTNETADR('123.17.5.1')  LIND(ETHLINE)
            SUBNETMASK('255.255.255.0')
            ALIASNAME(TEST_NETWORK)
```

This command will add a TCP/IP interface for the Ethernet line named ETHLINE. This interface will have the alias name TEST_NETWORK. When starting or ending this interface, the alias name of TEST_NETWORK can be used in place of the IP address.

Example 5: Adding an IPv6 Interface

```
ADDTCPIFC   INTNETADR('1234::5678:9abc:def')  LIND(ETHLINE)
            ADRPFXLEN(32) DADMAXTRN(10)
            TEXT('Example 5 IPv6 interface')
```

This command will add an IPv6 TCP/IP interface for the Ethernet line named ETHLINE. The address prefix length is specified as 32, the duplicate address detection maximum transmits is 10, and a text description for the interface has also been specified.

Example 6: Adding an IPv6 Stateless Address Auto-Configuration Interface

```
ADDTCPIFC   INTNETADR(*IP6SAC)  LIND(ETHLINE) MTU(1280)
            IFCID(123456789abcdef0) DADMAXTRN(4) PVYEXT(*YES)
            TEXT('ETHLINE IPv6 SAC interface')
```

This command will add an IPv6 stateless address auto-configuration interface for line ETHLINE. The MTU size is 1280, the interface ID to be used in the generated addresses is x'123456789abcdef0', the duplicate address detection maximum transmits is 4, and IPv6 stateless address auto-configuration privacy extensions are to be used.

Example 7: Adding an IPv4 Virtual Interface

```
ADDTCPIFC   INTNETADR('192.168.2.3')  LIND(ETHLINE1)
            SUBNETMASK('255.255.255.0')
ADDTCPIFC   INTNETADR('192.168.2.2')  LIND(ETHLINE2)
            SUBNETMASK('255.255.255.0')
ADDTCPIFC   INTNETADR('192.168.2.1')  LIND(*VIRTUALIP)
            SUBNETMASK('255.255.255.255')
            PREFIFC('192.168.2.2' '192.168.2.3')
```

These commands add two IPv4 interfaces and an IPv4 virtual interface that has the first two interfaces as preferred IPv4 interfaces. Interface 192.168.2.2 is preferred over 192.168.2.3.

Example 8: Adding an IPv6 Virtual Interface

```
ADDTCPIFC   INTNETADR(*IP6SAC) LIND(ETHLINE1)
ADDTCPIFC   INTNETADR('1234::1111') LIND(ETHLINE2)
ADDTCPIFC   INTNETADR('FE80::1') LIND(ETHLINE3)
ADDTCPIFC   INTNETADR('1234::5678')  LIND(*VIRTUALIP)
            ADRPFXLEN(64) PREFLIND(ETHLINE3 ETHLINE1 ETHLINE2)
```

These commands add three IPv6 interfaces and an IPv6 virtual interface that has the line descriptions specified by the first three interfaces to be used for VIPA proxy Neighbor Discovery agent selection. Line description ETHLINE3 is preferred over ETHLINE1 and ETHLINE2, and ETHLINE1 is preferred over ETHLINE2.

Top

Error messages

***ESCAPE Messages**

TCP1D03
&1 member record length not correct.

TCP1D04
Error occurred processing member &1 of &2/&3.

TCP1901
Internet address &2 not valid.

TCP1902
Internet address &1 not valid.

TCP1908
Internet address &1 not valid.

TCP2665
&2 &1 not added successfully.

TCP2666
&2 &1 not added.

TCP8050
*IOSYSCFG authority required to use &1.

TCP9999
Internal system error in program &1.

Top

Add TCP/IP Port Restriction (ADDTCPPORT)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add TCP/IP Port Restriction (ADDTCPPORT) command is used to restrict a port or range of ports in the TCP/IP configuration to a particular user profile. A port can be restricted for use by multiple user profiles. The addition of the user profile takes effect immediately. Any user profiles currently using a port that will not have access to that port after the use of this command are allowed to finish processing.

The default authorization for TCP/IP ports is to allow any user profile access to any port. If it is unnecessary to restrict a port to a user profile or a group of user profiles, the system administrator does not need to use this command.

Once an application running under a user profile has obtained the use of a restricted port, TCP/IP does not prohibit that application from passing its rights to another job that may be running under another user profile. The new user profile for the port is not checked against the list of user profiles having exclusive rights to that port. That is because the allocation of the port occurred under the user profile that had exclusive rights to that port.

The check for restricted use of the port occurs only on the BIND operation to the port. If other user profiles are currently using a port and an administrator wants to restrict a port or range of ports, the administrator may need to end all current TCP connections or user datagram protocol (UDP) sockets using that port. To do this, enter NETSTAT, select option 3, then select all of the connections or listening sockets that are using the port that you want to restrict. Enter an option 4 (ENDTCPCNN) for each.

There are two independent sets of ports. One set is for TCP processing and the other is for UDP processing. They are completely independent sets of ports and have no relationship to one another.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
PORT	Range of port values	*Element list*	Required, Positional 1
	Element 1: Lower value	1-65535	
	Element 2: Upper value	1-65535, ***ONLY**	
PROTOCOL	Protocol	*UDP, *TCP	Required, Positional 2
USRPRF	User profile	*Character value*	Required, Positional 3

Top

Range of port values (PORT)

Specifies the port number or range of port numbers identifying the port or ports that are being restricted. Valid values range from 1 through 65535. However, some of the ports in the range 1 through 1023 are used by system-supplied TCP/IP applications. If the user specifies one of these ports, it can affect the operation of those applications. See the assigned numbers RFC for the definition of port numbers currently used by TCP/IP applications.

This is a required parameter.

Element 1: Lower value

1-65535
Specify the port value or the lower port value in a range that you want restricted.

Element 2: Upper value

***ONLY**
The port value specified in the lower port value is the only port value that is restricted.

1-65535
Specify the upper port value in a range that you want restricted.

Top

Protocol (PROTOCOL)

Specifies the transport protocol associated with the port or range of ports being restricted. Each transport protocol has its own distinct set of ports in the range of 1 to 65535.

This is a required parameter.

***UDP** The port is a User Datagram Protocol (UDP) transport protocol port.

***TCP** The port is a Transmission Control Protocol (TCP) transport protocol port.

Top

User profile (USRPRF)

Specifies the user profile to which the port or range of ports is being restricted. Only jobs running under this profile or group profile may use the port or range of ports specified.

A user profile that is used as a group profile may be specified in the user profile field of this command. If users have a group profile specified in their user profile and that group profile was specified for a particular port or range of ports, then these users are given access to the specified port or range of ports. However, adopted authorities are not used when deciding whether this port is restricted or not. Each user profile or group profile that wants to use a port or range of ports must be explicitly added.

When a socket application issues the bind() system call, the user profile that the job is running under is checked against the list of user profiles that are associated with the specified port. If there is not a match on that user profile, then a check is made to determine if this user profile is part of a group and that the group profile is in the list of user profiles that are associated with the specified port.

For example, there are two user profiles, USER_1 and USER_2. USER_2 is specified as a member of a group associated with USER_1. If the TCP port 1015 has a user profile list consisting of USER_1, then a bind() by USER_2 will work because USER_2 is a part of the group profile USER_1.

This is a required parameter.

name Specify the name of the user profile that the port or range of ports is restricted to.

Top

Examples

Example 1: Adding a Single User Profile

```
ADDTCPPORT   PORT(7059)  PROTOCOL(*UDP)  USRPRF(TCPUSER)
```

This command adds the user profile TCPUSER to the set of user profiles that are allowed to bind UDP port 7059. User profiles that have not been added to this set or are not in a group profile that has been added will not be allowed to use UDP port 7059.

Example 2: Adding Multiple User Profiles

```
ADDTCPPORT   PORT(1590)  PROTOCOL(*TCP)  USRPRF(USER1)
ADDTCPPORT   PORT(1590)  PROTOCOL(*TCP)  USRPRF(USER2)
```

These commands show that a port can be restricted for use by multiple user profiles. User profiles USER1 and USER2 are the only users that are allowed to bind to TCP port 1590.

Example 3: Adding a Single User Profile to a Range of Ports

```
ADDTCPPORT   PORT(1591 1600)  PROTOCOL(*TCP)  USRPRF(USER3)
```

This command adds the user profile USER3 to the set of user profiles that are allowed to bind TCP ports 1591 through 1600.

Top

Error messages

***ESCAPE Messages**

TCP1D03
&1 member record length not correct.

TCP1D04
Error occurred processing member &1 of &2/&3.

TCP26E2
User profile &1 damaged.

TCP26E4
Port restriction action successful, but TCP/IP errors occurred.

TCP26FC
Upper port value must be *ONLY.

TCP26F1
Range of ports not valid.

TCP2677
Port restriction not added.

TCP2679
port entry was added successfully but errors occurred.

TCP2680
Duplicate port restriction found.

TCP8050
*IOSYSCFG authority required to use &1.

TCP9503
File &3 in library &2 not available.

TCP9509
Line &1 not found.

TCP9517
Duplicate port entry found.

TCP9526
User profile &1 not found.

TCP9999
Internal system error in program &1.

Top

Add Point-to-Point Profile (ADDTCPPTP)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Point-to-Point Profile (ADDTCPPTP) command is used to create a simple PPP (point-to-point protocol) connection profile. Profiles can be created to answer incoming calls by specifying OPRMODE(*ANS), or to dial remote systems by specifying OPRMODE(*DIAL).

Note: The preferred method of creating point-to-point profiles is through System i Navigator, since it supports all features of PPP. In cases where it is not feasible to create a point-to-point profile using System i Navigator, the ADDTCPPTP command can be used to create a simple point-to-point profile.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
CFGPRF	Configuration profile	*Character value*	Required, Positional 1
OPRMODE	Operating mode	*ANS, *DIAL	Required, Positional 2
RSRCNAME	Resource name	*Name*, ***CALC**	Optional, Positional 3
MODEM	Modem	*Character value*, ***RSRCNAME**, *SELECT	Optional
DIALTONE	Wait for dial tone	***WAIT**, *NOWAIT	Optional
CALLNBR	Calling number	*Character value*, ***NONE**	Optional
LCLINTNETA	Local internet address	*Character value*, ***OPRMODE**, *CURRENT, *DYNAMIC	Optional
RMTINTNETA	Remote internet address	*Character value*, ***OPRMODE**, *DYNAMIC	Optional
ENBPPPAUT	Enable PPP authentication	***NO**, *YES	Optional
PPPAUT	PPP authentication	*Element list*	Optional
	Element 1: User name and password	Values (up to 10 repetitions): *Element list*	
	Element 1: User name	*Character value*	
	Element 2: User password	*Character value*	
	Element 2: Authentication protocol	***ENCRYPTED**, *UNENCRYPTED	
	Element 3: Validation list	*Name*, ***CFGPRF**	
INACTTMR	Inactivity timer	15-65535, ***NOMAX**	Optional
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional
FULLMASQ	Full masquerading	***NO**, *YES	Optional
IPDTGFWD	IP datagram forwarding	***NO**, *YES	Optional
ALWRMTOVR	Allow remote address override	***NO**, *YES	Optional

Keyword	Description	Choices	Notes
ADDDFTRTE	Add default route	*NO, *YES	Optional

Top

Configuration profile (CFGPRF)

Specifies the point-to-point configuration profile to be added.

This is a required parameter.

character-value
Specify the name of a point-to-point configuration profile to add.

Top

Operating mode (OPRMODE)

Specifies the mode of operation for this point-to-point connection profile.

This is a required parameter.

***ANS** The profile is defined to answer calls from a remote system.

***DIAL**
The profile is defined to dial a remote system.

Top

Resource name (RSRCNAME)

Specifies the communications hardware resource to be used by this profile.

***CALC**
The resource name will be determined as follows:

- Look for resources being used by the 2771 integrated modem. If only one 2771 is defined, use that resource for this PPP line. *CALC is not valid if more than one 2771 modem is defined.
- If a 2771 cannot be used, determine if any resources are defined for use by ECS (Electronic Customer Support). If an ECS resource is available, use that resource for this PPP line.
- If neither a single 2771 integrated modem or ECS resource is available, the resource cannot be calculated and it will have to be explicitly defined.

name Specify the name of the hardware resource.

Note: You can use the Work With Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name.

Top

Modem (MODEM)

Specifies the modem description to use for this point-to-point profile. The list of currently supported modem descriptions can be seen by using the *SELECT value in interactive mode or by using option 11 from the Configure Point-to-Point TCPIP (CFGTCPPTP) command.

***RSRCNAME**

The modem name will be determined based on the value defined for the RSRCNAME parameter.

- If the resource is defined to use the 2771 integrated modem, the '2771 Internal Modem' description will be used.
- If the resource is defined to use the 2772 integrated modem, the '2772 Internal Modem' description will be used.
- If the resource is defined to use the 2761 internal modem, the '2761 Internal Modem' description will be used.
- If the ECS resource was chosen, the 'IBM 7852-400' modem description will be used.
- Otherwise, if the resource does not have a pre-defined modem description, MODEM(*RSRCNAME) cannot be used and the modem description will have to be explicitly defined.

***SELECT**

A list of modems is shown from which you will select the modem to use. This option is only valid when running the ADDTCPPTP command in interactive mode, otherwise an error will occur. If you are running interactively, it is recommended that you use the *SELECT value to help ensure that you properly select the modem to use.

character-value

Specify the name of the modem to use. Note that modem names are case sensitive and must match exactly to the modems defined for the system.

You can also specify a generic name of the modem you wish to use. A generic modem name is a character string of one or more characters followed by an asterisk (*); for example, 'abc*'. If a generic name is specified, then the **first** modem name that matches the generic name will be used. It is recommended that you include as many characters in the modem name string as possible to avoid any ambiguity. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete modem name. The actual modem name chosen will be posted in a message in the job log.

Top

Wait for dial tone (DIALTONE)

Specifies whether or not the modem should wait for a dial tone before dialing out.

***WAIT**

The modem will wait for a dial tone before dialing out.

***NOWAIT**

The modem will dial out without waiting for a dial tone.

Top

Calling number (CALLNBR)

Specifies the telephone number of the remote system to call. This parameter is only used when OPRMODE(*DIAL) is defined.

***NONE**

No telephone number is defined. If OPRMODE(*DIAL) is specified, a value other then *NONE must be defined for this parameter.

character-value

Specify the telephone number to call to connect to the remote system. If additional numbers are

required to establish an outside call, they must also be specified. Special character ',' (comma) may be used to signify if a delay is required before dialing the next number. Typically this delay is one second for most modems.

Top

Local internet address (LCLINTNETA)

Specifies the local internet address of the system to be used for this PPP connection.

***OPRMODE**
The mode of operation will determine the local IP address. If OPRMODE(*DIAL) is specified, the local IP address will be defined as *DYNAMIC. If OPRMODE(*ANS) is specified, the local IP address will defined as *CURRENT.

***DYNAMIC**
The IP address will be defined by the remote system during PPP negotiations.

***CURRENT**
The current local IP address will be used. This address is determined as follows:

- If a local host name has been defined using the Change TCP/IP Domain (CHGTCPDMN) command, this host name will be resolved to an IP address (either defined in the local host table or by a domain name server). If this IP address is found to also exist on the local system, it will be used.
- If no local host name is defined, or could not be resolved to, the first valid local IP address found will be used.
- If no valid local IP addresses are found, *CURRENT will not be allowed.

character-value
Specify the internet address to use as the local IP address for this PPP profile. The address specified here can already exist on the system or a unique IP address can be defined.

Top

Remote internet address (RMTINTNETA)

Specifies the internet address of the remote system to use for this PPP connection.

***OPRMODE**
The mode of operation will determine the remote IP address. If OPRMODE(*DIAL) is specified, the remote IP address will be defined as *DYNAMIC. If OPRMODE(*ANS) is specified, the remote IP address will be defined as '169.254.x.x' where 169.254.x.x is a reserved IANA LINKLOCAL network address. The actual host portion (x.x) of this address will be determined at run-time.

***DYNAMIC**
The IP address will be defined by the remote system during PPP negotiations.

character-value
Specify the internet address to use as the remote IP address for this PPP profile. The address specified here **must** be an unique IP address for this system.

Top

Enable PPP authentication (ENBPPPAUT)

Specifies whether PPP authentication will be enabled for this profile.

***NO** No PPP authentication will be required to either connect to the remote system or to allow the remote system to connect to the local system.

***YES** PPP authentication will be required to either connect to the remote system or to allow the remote system to connect to the local system.

Top

PPP authentication (PPPAUT)

Specifies the PPP authentication values to use for PPP authentication. This parameter is only in effect if ENBPPPAUT(*YES) is defined.

Element 1: User name and password

Specifies the user names and passwords to use for PPP authentication. Only one user name and password can be specified for OPRMODE(*DIAL) profiles. Up to ten user names and passwords can be specified for OPRMODE(*ANS) profiles. Additional entries can be added with System i Navigator.

Element 1: User name

character-value
Specify the user name to be used for PPP authentication. Each user name can be up to 64 characters in length. User names are case sensitive and will be stored exactly as they are entered.

Element 2: User password

character-value
Specify the password associated with the user name. Each password can be up to 64 characters in length. Passwords are case sensitive and will be stored exactly as they are entered.

Element 2: Authentication protocol

Specifies which authentication protocol to use for PPP authentication. This value also specifies which authentication protocol to associate with each specified user name and password.

***ENCRYPTED**
Only PPP authentication methods using encrypted passwords will be used. This value will support both EAP (Extended Authentication Protocol) and CHAP (Challenge Handshake Authentication Protocol - using MD-5 hashing algorithm).

***UNENCRYPTED**
Only PPP authentication methods using unencrypted passwords will be used. Currently this is limited to PAP (Password Authentication Protocol).

Note: This method of authentication is not as secure as CHAP or EAP since it allows user name and password information to flow over the link unprotected.

Element 3: Validation list

Specifies the validation list used to store remote user name and password information for OPRMODE(*ANS) profiles. This element is ignored if OPRMODE(*DIAL) is specified. All validation lists defined for use by PPP must exist in library QUSRSYS.

***CFGPRF**
The validation list to store user name and password information is the same name as the point-to-point profile. If the validation list does not exist, it will be created in library QUSRSYS.

name Specify the name of the validation list in library QUSRSYS to store username and password information. If the validation list does not exist, it will be created.

Top

Inactivity timer (INACTTMR)

Specifies the time (in seconds) that the system waits for user data activity for this profile before disconnecting. This timer is started once LCP (Link Control Protocol) and NCP (Network Control Protocol) negotiations have completed successfully, and restarted when user data is sent or received. LCP and NCP packets do not cause this timer to be restarted.

***NOMAX**
The inactivity timer is disabled.

15-65535
Specify the number of seconds to use as the timer value.

Top

Text 'description' (TEXT)

Specifies the text description for the point-to-point profile.

***BLANK**
No text is specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Full masquerading (FULLMASQ)

Specifies whether full masquerading will be used. If enabled, all IP addresses will be hidden behind the remote IP address for the dial PPP connection. This parameter is only in effect if OPRMODE(*DIAL) is specified. This feature allows all outbound IP traffic to have it's source IP address translated to the remote IP address defined for the PPP link. The source port is also modified, so that return IP traffic can be properly associated with the correct conversation and have it's IP address and destination port changed back to the correct values.

This feature is particularly useful to allow other hosts on the same network as the local system to use the system as a gateway to the Internet. If the local system is connected to the Internet using an ISP (Internet Service Provider), the other hosts, such as PCs, could also gain access to the Internet and 'hide' behind the system's assigned PPP IP address.

***NO** No IP addresses will be hidden (masqueraded) behind the system's PPP IP address.

***YES** All IP addresses will be hidden (masqueraded) behind the system's PPP IP address when traffic flows out the PPP link.

Top

IP datagram forwarding (IPDTGFWD)

Specifies whether IP datagram forwarding is enabled for this PPP connection. This parameter is only in effect if OPRMODE(*ANS) is specified.

***NO** Internet Protocol (IP) will discard those datagrams from the remote system that are not destined for any addresses local to this system.

***YES** This allows Internet Protocol (IP) datagrams not destined for this system to pass through this system onto a connected network. Enabling IP datagram forwarding essentially enables the system to act as router for this connection. Careful security considerations should be reviewed prior to enabling IP forwarding for the PPP link. Note that this will only take effect if system wide IP datagram forwarding is enabled, otherwise it will be ignored even if marked. System-wide IP datagram forwarding is controlled by the IPDTGFWD parameter on the Change TCP/IP Attributes (CHGTCPA) command.

Top

Allow remote address override (ALWRMTOVR)

Specifies whether remote systems will be allowed to override the remote IP address defined in RMTINTNETA. This parameter is only in effect if OPRMODE(*ANS) is specified and RMTINTNETA(*DYNAMIC) is not specified.

***NO** If a specific remote IP address is defined for RMTINTNETA, remote systems will not be allowed to define their own address. The remote system must use the address defined by the local system or the PPP connection will be terminated.

***YES** If a specific remote IP address is defined for RMTINTNETA, remote systems will still be allowed to define their own address. This is useful if you want to allow more than one type of remote client to be able to dial into the system. Typically the remote system dialing in will request that it be told what its IP address is. By specifying ALWRMTOVR(*YES), you tell remote clients what their IP address should be, but also allow other remote clients to specify their own address without the need for an additional profile or resource.

Top

Add default route (ADDDFTRTE)

Specifies whether you want a default route added when this Point-to-Point Connection profile is started. This parameter is only in effect if OPRMODE(*DIAL) is specified.

***NO** A default route will NOT be automatically added.

***YES** A default route will be automatically added when this Point-to-Point profile is started. The next hop address will be the IP address of the remote system.

Top

Examples

Example 1: Create an Answer Profile

```
ADDTCPPTP   CFGPRF(ANSPROFILE)  OPRMODE(*ANS)
```

This command will create answer PPP profile with the following properties

- The resource will be calculated and the modem description will be determined by the resource. Assuming a 2771 integrated modem is found then the '2771 Internal modem' modem description will be used.
- Line description 'QPPPCMNxx' will be created, where CMNxx is the 2771 resource.
- An existing IP address on the local system will be defined as the local IP address. If there is an IP address associated with the local host name then this address will be used. If not, then the first local IP address found for the local system will be used.
- The remote IP address (address that is assigned to the remote system) will be defined as 169.254.x.x, where x.x is determined at runtime.
- Authentication is not enabled.

Example 2: Create a PPP Dial Profile

```
ADDTCPPTP   CFGPRF(DIALPROF)  OPRMODE(*DIAL)  RSRCNAME(CMN14)
            MODEM('2761 Internal Modem') DIALTONE(*WAIT)
            CALLNBR('1,,9876543')
            ENBPPPAUT(*YES)
            PPPAUT(((dialuser dialpw)) *ENCRYPTED *CFGPRF)
            FULLMASQ(*YES)
```

This command will create a dial PPP profile with the following properties:

- The profile will use a PPP line named 'QPPPCMN14', defined to use communication resource CMN14.
- The 2761 internal modem will be used (Modem name as seen in CFGTCPPTP, option 11).
- When calling the remote system, a '1' will be dialed first (possibly to reach an outside line), then there will a 2 second delay (approximately), then telephone number '9875432' will be called.
- Authentication is enabled and an authentication protocol using encryption will be used (EAP - extended authentication protocol or CHAP - Challenge authentication protocol (MD-5)). The user name and password defined will be used for authentication.
- The local and remote IP addresses will be defined as *DYNAMIC, which means the addresses will be defined by the remote system during the IPCP (Internet Protocol Control Protocol) negotiation phase of the PPP connection.
- All IP traffic going out the PPP link will appear as if it originated from the local system's PPP IP address.

Example 3: Create Profile Using Predefined IP Addresses

```
ADDTCPPTP   CFGPRF(ANSPROFILE)  OPRMODE(*ANS)  RSRCNAME(CMN10)
            MODEM('USRobotics 56K*')  LCLINTNETA('10.9.8.1')
            RMTINTNETA('10.9.8.2')  ENBPPPAUT(*YES)
            PPPAUT(((RmtID1 RmtPW1) (RmtID2 RmtPW2))
                    *ENCRYPTED PPPVLDL)
            TEXT('PPP *ANS profile')  IPDTGFWD(*YES)
```

This command will create an answer PPP profile with the following properties:

- The profile will use a PPP line named 'QPPPCMN10', defined to use communication resource CMN10.
- Modem name of 'USRobotics 56K V.90 Sportster' will be used assuming it is the first modem name found starting with the string 'USRobotics 56K'. The actual modem selected will be posted to the joblog in a message.
- Authentication is enabled and an authentication protocol using encryption will be used (EAP - extended authentication protocol or CHAP - Challenge authentication protocol (MD-5)). Both user RmtID1 and RmtID2 are authorized to connect using this profile.
- Validation list PPPVLDL in library QUSRSYS will be used to store the user names and passwords.

- The local IP address will be 10.9.8.1 and the remote IP address will be 10.9.8.2. The 10.9.8.1 local address is an existing IP address on the system and is attached to the 10.9.8.0 network.
- The remote system will be allowed to directly access the 10.9.8.0 network.

Top

Error messages

***ESCAPE Messages**

TCP83D0

Point-to-point profile &1 not added.

Top

Add TCP/IP Remote System (ADDTCPRSI)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add TCP/IP Remote System (ADDTCPRSI) command is used to associate an internet address with an X.25 network address or a local permanent virtual circuit (PVC) logical channel identifier in the TCP/IP configuration.

When the user works with an X.25 public or private data network, the internet address and the network address of each remote system or local (PVC) logical channel identifier needs to be specified.

ATTENTION:

1. Do not specify the X.25 network address for systems on the X.25 Defense Data Network (DDN). The X.25 DDN has a built-in conversion algorithm that converts an IP address to the remote DTE address. If you specify an X.25 network address for remote systems on an X.25 DDN, the DDN conversion algorithm is bypassed. In this case it is possible that you will not be able to connect to the requested host.
2. Before attempting to start an X.25 interface, ensure that the remote system information (RSI) for non-DDN X.25 interfaces that use a permanent virtual circuit (PVC) is configured. Use the ADDTCPRSI command to do this. Incoming data from a remote system on the X.25 network is not processed unless an RSI entry for the PVC is configured on the X.25 interface before the interface is started.
3. Attempts to change or remove a route or interface that is required to reach an existing RSI entry will fail.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.

Note: If specific values are entered for DFTPKTSIZE and DFTWDWSIZE and the interfaces or routes are changed, conflicts could result. If *LIND is used, these values are adjusted accordingly if changes occur at the interface and route level.

Top

Parameters

Keyword	Description	Choices	Notes
INTNETADR	Internet address	*Character value*	Required, Positional 1
NETADR	Network address	*Character value*	Optional, Positional 2
PVCLGLCHLI	PVC logical channel identifier	*Character value*	Optional, Positional 3
RVSCRG	X.25 reverse charge	***NONE**, *REQUEST, *ACCEPT, *BOTH	Optional

Keyword	Description	Choices	Notes
DFTPKTSIZE	Default packet size	*Element list*	Optional
	Element 1: Transmit packet size	***LIND**, 64, 128, 256, 512, 1024, 2048, 4096	
	Element 2: Receive packet size	***LIND**, *TRANSMIT, 64, 128, 256, 512, 1024, 2048, 4096	
DFTWDWSIZE	Default window size	*Element list*	Optional
	Element 1: Transmit window size	1-15, ***LIND**	
	Element 2: Receive window size	1-15, ***LIND**, *TRANSMIT	

Top

Internet address (INTNETADR)

Specifies the internet address of the remote system. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

This is a required parameter.

Top

Network address (NETADR)

Specifies the X.25 network address or the DTE that is to be associated with the X.25 internet address. The user can specify a decimal number that is 1 through 17 digits in length.

Note: If you specify a value for this parameter, the PVCLGLCHLI parameter value cannot be specified.

network-address
Specify the X.25 network address of a remote X.25 system.

Top

PVC logical channel identifier (PVCLGLCHLI)

Specifies the local permanent virtual circuit (PVC) logical channel identifier that is used to establish an X.25 PVC interface to the specified remote internet address. One unique channel identifier may be specified. This unique channel identifier must have been previously identified in the ADDTCPIFC or CHGTCPIFC command that defined a TCP/IP X.25 interface. The TCP/IP X.25 PVC logical channel identifier is used to establish the circuit between this system's TCP/IP X.25 interface and the host defined by the remote internet address. The logical channel identifier must also exist in the X.25 line description used for the TCP/IP X.25 interface.

Notes:

1. If this parameter's value is specified, the NETADR parameter value cannot be specified.
2. When specifying a PVC, consider which interface or set of interfaces this RSI entry could use to connect to the remote system. Each of the interfaces that could be used to reach this RSI entry's remote system must have the specified PVC logical channel ID configured as part of the interface.

logical-channel-identifier
Specify the PVC logical channel identifier value. The value may be from 001 to FFF. Only 1 PVC logical channel identifier can be specified.

Top

X.25 reverse charge (RVSCRG)

Specifies whether reverse charges are accepted or requested on an X.25 remote system basis.

***NONE**
Reverse charges are not accepted or requested.

***REQUEST**
Reverse charges are requested on outgoing call request packets. Reverse charges are not accepted on incoming call request packets.

***ACCEPT**
Reverse charges are accepted on incoming call request packets. Reverse charges are not requested on outgoing call request packets.

***BOTH**
Reverse charges are requested for outgoing call request packets and are accepted on incoming call request packets.

Top

Default packet size (DFTPKTSIZE)

Specifies the default packet size used by the X.25 network for transmission and reception. The values specified here should match the default values used by the X.25 network.

Element 1: Transmit packet size

***LIND**
The value specified in the line description associated with the X.25 interface used to reach the remote system is used as the default packet size.

transmit-packet-size
Specify a default packet size for transmission. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

Element 2: Receive packet size

***LIND**
The value specified in the line description associated with the X.25 interface used to reach the remote system is used as the default packet size.

***TRANSMIT**
The value specified as the packet size for transmission is used as the default for reception.

receive-packet-size
Specify a default packet size for reception. The valid values for the packet size are 64, 128, 256, 512, 1024, 2048, and 4096.

Top

Default window size (DFTWDWSIZE)

Specifies the default packet window size for transmission to and reception from remote systems attached to the X.25 line.

Element 1: Transmit window size

***LIND**
The value specified in the line description associated with the X.25 interface used to reach the remote system is used as the default window size.

1-15
Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use 128 packet numbering. The modulus value is specified on the X.25 line description.

Element 2: Receive window size

***LIND**
The value specified in the line description associated with the X.25 interface used to reach the remote system is used as the default window size.

***TRANSMIT**
The value specified as the default window size for transmission is used as the default for reception.

receive-window-size
Specify the appropriate default window size. Valid values range from 1 through 7 for networks that use modulus 8 packet numbering. Valid values range from 1 through 15 for networks that use 128 packet numbering. The modulus is specified on the X.25 line description.

Top

Examples

Example 1: Adding RSI with NETADR

```
ADDTCPRSI   INTNETADR('8.76.0.12')  NETADR(4005)
```

This command allows the TCP/IP protocol stack to associate the internet address of 8.76.0.12 with the X.25 network address of 4005. Defaults are used for the remaining parameters.

Example 2: Adding RSI with PVCLGLCHLI

```
ADDTCPRSI   INTNETADR('145.9.43.188')  PVCLGLCHLI(231)
```

This command allows the TCP/IP protocol stack to associate the internet address of 145.9.43.188 with the X.25 PVC local logical channel identifier 231. Defaults are used for the remaining parameters.

Example 3: Adding RSI with Additional Parameters

```
ADDTCPRSI   INTNETADR('135.63.45.23')  NETADR(6031546)
            RVSCRG(*BOTH)  DFTPKTSIZE(1024 *TRANSMIT)
            DFTWDWSIZE(*LIND *TRANSMIT)
```

This command allows the TCP/IP protocol stack to associate the internet address of 135.63.45.23 with the X.25 network address of 6031546. The reverse charges are used for both outgoing and incoming call request packets. The default packet size is set to 1024, and the default window size is set to the value specified in the line description associated with the X.25 interface used to reach the remote system.

Top

Error messages

***ESCAPE Messages**

TCP1D03
&1 member record length not correct.

TCP1D04
Error occurred processing member &1 of &2/&3.

TCP1901
Internet address &2 not valid.

TCP1902
Internet address &1 not valid.

TCP1908
Internet address &1 not valid.

TCP26D5
Error occurred processing file.

TCP8050
*IOSYSCFG authority required to use &1.

TCP9999
Internal system error in program &1.

Top

Add TCP/IP Route (ADDTCPRTE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add TCP/IP Route (ADDTCPRTE) command is used to identify a route to a remote network or a route to a remote destination system in the Transmission Control Protocol/Internet Protocol (TCP/IP) configuration.

Five parameter values uniquely define an IPv4 route. These values are the route destination (RTEDEST), the subnet mask (SUBNETMASK), the type of service (TOS), the IPv4 address of the next system on the route (NEXTHOP), and the preferred binding interface (BINDIFC).

Four parameter values uniquely define an IPv6 route. These values are the route destination (RTEDEST), the address prefix length (ADRPFXLEN), the IPv6 address of the next system on the route (NEXTHOP), and the binding line description (BINDLIND).

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.
- An route cannot be added unless the internet address specified by the NEXTHOP parameter can be reached directly through a network associated with a previously defined TCP/IP interface. An interface can be added using the ADDTCPIFC command.
- A route destination value of 127.nnn.nnn.nnn (where nnn is any value from 0 to 255) is not allowed. It is a reserved value for *LOOPBACK.

Top

Parameters

Keyword	Description	Choices	Notes
RTEDEST	Route destination	*Character value*, *DFTROUTE, *DFTMCAST, *DFT6ROUTE, *DFT6MCAST	Required, Positional 1
SUBNETMASK	Subnet mask	*Character value*, ***NONE**, *HOST	Optional, Positional 2
TOS	Type of service	*MINDELAY, *MAXTHRPUT, *MAXRLB, *MINCOST, ***NORMAL**	Optional, Positional 3
NEXTHOP	Next hop	*Character value*	Optional
ADRPFXLEN	Address prefix length	1-128, **64**, *HOST, *NONE	Optional
BINDIFC	Preferred binding interface	*Character value*, ***NONE**	Optional
BINDLIND	Binding line description	*Name*	Optional
MTU	Maximum transmission unit	576-16388, ***IFC**	Optional
METRIC	Route metric	1-16, **1**	Optional
REDST	Route redistribution	***NO**, *YES	Optional
DUPRTEPTY	Duplicate route priority	1-10, ***MEDIUM**, *HIGH, *LOW	Optional
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional

Top

Route destination (RTEDEST)

Specifies the route destination being added.

For an IPv4 route, you must specify all 4 bytes that make up an internet address though some of the bytes may be equal to 0. For example, a route to all the hosts on the 9.5.11 subnetwork is identified by entering 9.5.11.0 for the route destination. Used in combination with a subnetmask, type of service value, and next hop, the route destination uniquely identifies a route to a network or system.

For an IPv6 route, the entire 16 byte IPv6 address must be specified, and the suffix portion of the RTEDEST field must be zero. For example, a RTEDEST with value 1234:5678:: and address prefix length (ADRPFXLEN) of 32 is a valid combination, but a RTEDEST with value 1234:5678:8000:: and ADRPFXLEN of 32 is not.

This is a required parameter.

***DFTROUTE**
Specifies that a default IPv4 route is being added. A default route is used by the system to send data to a remote destination for which a specific route is not defined. The default routes are used based on the availability of the next hop gateway and the type of service (TOS). If the application requests a specific TOS, the TOS of the default route used must match the TOS requested. If no default route is found that matches the requested TOS, the first available default route with a TOS of *NORMAL is used.

***DFTMCAST**
Use the *DFTMCAST special value to indicate that the static IPv4 route you are adding is a default multicast route. A default multicast route is used by an application when sending data to a multicast destination address and a specific outgoing interface is not specified.

Note: When RTEDEST(*DFTMCAST) is specified, then SUBNETMASK(*NONE) must also be specified and the NEXTHOP parameter must be a local TCP/IP interface (on this system).

***DFT6ROUTE**
Specifies that a default IPv6 route is being added. A default IPv6 route is used by the system to send data to a remote IPv6 destination for which a specific route is not defined. If *DFT6ROUTE is specified, ADRPFXLEN must specify *NONE.

***DFT6MCAST**
Specifies that a default IPv6 multicast route is being added. A default IPv6 multicast route is used by an application when sending data to a IPv6 multicast destination address and a specific outgoing physical interface is not specified. If *DFT6MCAST is specified, ADRPFXLEN must specify *NONE.

character-value
Specify the route destination being added. For an IPv4 route, the route destination can be specified in the form *nnn.0.0.0*, for Class A, *nnn.nnn.0.0* for Class B, and *nnn.nnn.nnn.0* for Class C, or *nnn.nnn.nnn.nnn* for any combination thereof, where *nnn* is a decimal number ranging from 0 through 255.

Any combination thereof means that you may specify a route, such as 9.5.0.0 to the hosts on the 9.5 subnet, even though all 9.5.x.x addresses are class A network addresses.

Exceptions:

- The first byte (octet) must be greater than 0 and less than 255.
- The last byte (octet) may not equal 255.
- The last byte (octet) may not equal 0 if *HOST is specified for the SUBNETMASK value.
- Routes to a broadcast address are not allowed.

For an IPv6 route, the route destination is specified in the form *x:x:x:x:x:x:x:x*, where *x* is a hexadecimal number ranging from 0 through X'FFFF'. "::" may be used once in the route destination to indicate one or more groups of 16 bits of zeros. The "::" may be used to compress leading, imbedded, or trailing zeros in the route destination. The suffix portion of the route destination must contain zeros.

Top

Subnet mask (SUBNETMASK)

Specifies a bit mask that identifies to TCP/IP which bits of the value specified for the route destination (RTEDEST) compose the network and subnet portions of the internet address. By defining the network portion and subnetwork portion of the RTEDEST address, the subnet mask also defines which bits of the RTEDEST address make up the host portion.

The mask is a 32-bit combination that is logically ANDed with the internet address to determine a particular subnetwork. The bits of the mask set to the value one (1) determine the network and subnetwork portions of the address. The bits set to the value zero (0) determine the host portion of the address.

SUBNETMASK is ignored if RTEDEST contains an IPv6 address.

***NONE**
No subnet mask is used. A subnet mask is not used when specifying default routes. For example, when RTEDEST(*DFTMCAST) or RTEDEST(*DFTROUTE) is specified, SUBNETMASK(*NONE) must also be specified.

***HOST**
The internet address value specified in the route destination field is a host address. The subnetmask value is calculated to be 255.255.255.255.

character-value
Specify the mask of the subnet field. The internet address is in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. For example, a destination route's internet address value of 129.35.192.0 identifies a Class B subnetwork. The network ID part of its address is 129.35. The portion of the subnetmask that is associated with the network portion of a particular class of address must equal 255. Therefore, the upper 2 bytes must be equal to 255.255 in the subnetmask. The subnetmask in this example may be 255.255.192.0 if the third octet is used as the subnetwork ID portion of the internet address.

Top

Type of service (TOS)

Specifies the type of service to be used. The type of service defines how the internet hosts and routers should make trade-offs between throughput, delay, reliability, and cost.

TOS is ignored if RTEDEST contains an IPv6 address.

***NORMAL**
Normal service is used for delivery of data.

***MINDELAY**
Minimize delay means that prompt delivery is important for data on this connection.

***MAXTHRPUT**
Maximize throughput means that a high data rate is important for data on this connection.

***MAXRLB**
Maximize reliability means that a higher level of effort to ensure delivery is important for data on this connection.

***MINCOST**
Minimize monetary cost means that lower cost is important for data on this connection.

Top

Next hop (NEXTHOP)

Specifies the internet address of the next system (gateway) on the route.

A route cannot be added unless the internet address specified by the NEXTHOP parameter can be reached directly through a network associated with a previously defined TCP/IP interface. An interface can be added by using the Add TCP/IP Interface (ADDTCPIFC) command.

Note: If RTEDEST specifies an IPv4 address, NEXTHOP must contain an IPv4 address. If RTEDEST specifies an IPv6 address, NEXTHOP must contain an IPv6 address.

character-value
Specify the internet address.

An IPv4 internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An IPv4 internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address.

An IPv6 internet address is specified in the form *x:x:x:x:x:x:x:x*, where *x* is a hexadecimal number ranging from 0 through X'FFFF'. "::" may be used once in the IPv6 address to indicate one or more groups of 16 bits of zeros. The "::" may be used to compress leading, imbedded, or trailing zeros in the address.

:: (null IPv6 address), ::1 (IPv6 loopback address) and addresses beginning with FF (IPv6 multicast address) are not allowed.

If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Note: If you are adding a *DFTMCAST (default multicast route) the *NEXTHOP internet address must be local, on this system, and not one hop away.

Top

Address prefix length (ADRPFXLEN)

Specifies the IPv6 address prefix length (in bits) of the RTEDEST parameter. The address prefix length specifies how much of the leftmost portion of an IPv6 address is the subnet prefix. This parameter applies only if RTEDEST is an IPv6 address.

64 The default is 64 since most IPv6 networks have a 64 bit interface ID (address suffix).

***HOST**
The RTEDEST field specifies the IPv6 address of a specific host system. *HOST corresponds to a value of 128.

***NONE**
No address prefix length is specified.

1-128 Specify how much of the leftmost portion of the RTEDEST IPv6 address is the subnet prefix.

Top

Preferred binding interface (BINDIFC)

Specifies the IP interface this route will be bound to. The bind is absolute.

Note: This parameter is useful only when there are multiple interfaces defined on your system for the same network. BINDIFC allows you to define which interface should be used to reach the network for a particular route destination. In this way you can distribute traffic (load balancing) across multiple interfaces so all routes do not use the same interface to reach the network.

If the IP interface you specify is active, this route will bind to it, otherwise it will follow the normal route binding rules (which are also used when BINDIFC is defined as *NONE).

BINDIFC is ignored if RTEDEST contains an IPv6 address.

***NONE**
No particular IP interface will be bound to this route. The first active IP interface on the network defined by the NEXTHOP and SUBNETMASK parameters will be used. This is the default value.

character-value
Specify the internet address (IP address) of the interface you want this route to bind to. The binding is preferred and absolute.

Top

Binding line description (BINDLIND)

Specifies the line description (LIND) object to which this route will be bound. This value must be specified for an IPv6 route and is ignored for an IPv4 route.

name Specify the name of the line description. The line description must be for an Ethernet line, and must already have at least one IPv6 interface using it.

Top

Maximum transmission unit (MTU)

Specifies the maximum size (in bytes) of IP datagrams that can be transmitted through this route. A datagram is a basic unit of information passed over an internet network. For IPv4, the minimum size of any maximum transmission unit value is 576 bytes. For IPv6, the minimum value is 1280 bytes.

***IFC** For an IPv4 route, the maximum transmission unit (MTU) is the MTU of the interface that is associated with this route. For an IPv6 route, the MTU is the maximum frame size of the binding line description (BINDLIND) associated with the route.

576-16388
Specify a value for the maximum transmission unit in bytes. The maximum MTU that can be specified for this route depends on the type of physical connection to the network. The following table lists the maximum MTU values that can be specified based on the line type:

X.25 4096

Token ring (4 meg)
4060

Token ring (16 meg)
16388

Ethernet 802.3
8992

Ethernet Version 2
9000

DDI 4352

Frame relay
8177

Wireless 802.3
1492

Wireless Version 2
1500

Twinax (TDLC)
4105

Notes:

1. TCP/IP uses the route MTU value to calculate the size of the datagrams it sends. If you are using path MTU discovery, specify MTU(*IFC). This will allow the TCP/IP support to calculate the most efficient MTU for this route. If you are not using path MTU discovery, and you do not know the smallest MTU used by host systems along the entire path of this route, use 576 (for IPv4 routes) or 1280 (IPv6 routes).
2. The MTU of a route cannot exceed the MTU of the interface on which the NEXTHOP value is accessed. If the interface's MTU value was specified as *LIND, the interface's MTU value is derived from the line description. If the route's MTU value is specified as *IFC and the interface's MTU value is specified as *LIND, both values are derived from the line description.
3. The actual MTU value used for a route is resolved during interface activation. This value is the minimum of either the specified MTU value for the route or the MTU value determined from the associated interface used by the route.

Top

Route metric (METRIC)

Allows you to assign a routing metric ″cost″ value to this route. The metric cost of a route is a factor in determining the desirability of the route. The metric value range is from 1 to 16. A metric value of 1 is close (one router hop) and therefore desirable. Desirability decreases as the metric value (distance) increases. A metric value of 16 is considered unreachable (an infinite distance away).

You can discourage the routing table from choosing this route by specifying a metric value that is higher than the actual number of hops to the destination and therefore reduce traffic on this route.

METRIC is ignored if RTEDEST contains an IPv6 address.

1 A routing metric cost value of 1 is used.

1-16 Specify the metric value to be used.

Top

Route redistribution (REDST)

Specifies whether this static route information will be shared with other routers. You can reduce traffic on this route by specifying *NO.

REDST is ignored if RTEDEST contains an IPv6 address.

***NO** This route will not be shown or shared with other routers.

***YES** This route will be shown to any requesting router.

Note: REDST(*YES) is analogous to the RIPv1 specification of STATIC. REDST(*NO) is analogous to the RIPv1 specification of PASSIVE.

Top

Duplicate route priority (DUPRTEPTY)

Specifies the duplicate route priority of this static route. This value determines which route is selected when multiple routes with the same value for route destination, subnet mask, and type of service (for IPv4 routes) or route destination and address prefix length (for IPv6 routes) satisfy the primary route selection criteria. Routes with a higher duplicate route priority (DUPRTEPTY) are used before routes with a lower one. The values allowed for this parameter are 1 (lowest priority) to 10 (highest priority).

***MEDIUM**
Specify medium priority. *MEDIUM corresponds to a value of 5.

***HIGH**
Specify high priority. *HIGH corresponds to a value of 10.

***LOW** Specify low priority. *LOW corresponds to a value of 1.

1-10 Specify the duplicate route priority value to be used.

Note: IPv4 routes have 10 different priority values. IPv6 routes have only 3: *HIGH, *MEDIUM, and *LOW. For an IPv6 route, if 1, 2, or 3 are specified for DUPRTEPTY, the route will have *LOW priority. If 4, 5, or 6 are specified for DUPRTEPTY, the IPv6 route will have *MEDIUM priority. If values 7, 8, 9, or 10 are specified for DUPRTEPTY, the IPv6 route will have *HIGH priority.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the route.

***BLANK**
No text is specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Examples

Example 1: Adding an IPv4 Route

```
ADDTCPRTE   RTEDEST('132.65.8.0')  SUBNETMASK('255.255.255.0')
            TOS(*MINDELAY)  NEXTHOP('148.92.6.40')  MTU(*IFC)
```

This command specifies the following for this route:

- A route destination of a class B network.
- Subnetting through the third octet.
- A minimum delay type of service for the interface.
- This route is connected to or can be reached by going through a gateway identified as 148.92.6.40.
- The maximum transmission unit (MTU) is to be calculated based on the interface associated with the next hop for this route.

Example 2: Adding an IPv4 Route with a Specific MTU

```
ADDTCPRTE   RTEDEST('9.10.45.0')  SUBNETMASK('255.255.255.0')
            TOS(*MAXRLB)  NEXTHOP('9.5.11.128') MTU(1994)
            TEXT('Route with specific MTU')
```

This command specifies the following for this route:

- A route destination of a class A network.
- Subnetting through the third octet.
- A maximum reliability type of service for the interface.
- This route is connected to or can be reached by going through a gateway identified as 9.5.11.128.
- A maximum transmission unit (MTU) of 1994.
- A text description of the route.

Example 3: Adding IPv4 Default Routes

```
ADDTCPRTE   RTEDEST(*DFTROUTE)  SUBNETMASK(*NONE)  TOS(*MINCOST)
            NEXTHOP('186.49.126.108')  MTU(*IFC)
ADDTCPRTE   RTEDEST(*DFTROUTE)  SUBNETMASK(*NONE)  TOS(*NORMAL)
            NEXTHOP('129.65.34.98')  MTU(576)
```

These commands specify that:

- Two default routes are used for this host.
- Data may be routed over either default route.
- Processing will use the first *DFTROUTE specified that also has the same type of service requested by the application.
- Minimum cost (*MINCOST) type of service is used for the first route and normal (*NORMAL) type of service is used for the second route.
- A maximum transmission unit (MTU) of *IFC is used for the first route and 576 for the second route.

Note: You cannot specify a subnetmask on a default route entry. It must equal *NONE.

Example 4: Adding an IPv6 Network Route

```
ADDTCPRTE   RTEDEST('1234::') ADRPFXLEN(16)
            NEXTHOP('5678::1234') BINDLIND(ETHLINE) MTU(*IFC)
```

This command specifies the following:

- A route to IPv6 subnetwork 1234::/16 (route destination of 1234:: and ADRPFXLEN of 16).
- A next hop of 5678::1234.
- The route is to be bound to line description ETHLINE.
- The MTU value is determined from the maximum frame size defined in line description ETHLINE.

Example 5: Adding an IPv6 Default Route

```
ADDTCPRTE   RTEDEST(*DFT6ROUTE) ADRPFXLEN(*NONE)
            NEXTHOP('1212::3434') BINDLIND(ETHLINE2) MTU(1280)
```

This command specifies the following:

- A default IPv6 route.
- A next hop of 1212::3434.
- The route is to be bound to line description ETHLINE2.
- An MTU value of 1280.

Example 6: Adding an IPv6 Host Route

```
ADDTCPRTE   RTEDEST('AAAA::BBBB') ADRPFXLEN(*HOST)
            NEXTHOP('3434::1212') BINDLIND(ETHLINE3)
            TEXT('IPv6 host route')
```

This command specifies the following:

- A route to IPv6 host address AAAA::BBBB.
- A next hop of 3434::1212.
- The route is to be bound to line description ETHLINE3.
- A text description is specified for the route.

Top

Error messages

***ESCAPE Messages**

TCP1D03
&1 member record length not correct.

TCP1D04
Error occurred processing member &1 of &2/&3.

TCP1901
Internet address &2 not valid.

TCP1902
Internet address &1 not valid.

TCP1908
Internet address &1 not valid.

TCP261C
Process completed successfully.

TCP2665
&2 &1 not added successfully.

TCP2666
&2 &1 not added.

TCP8050
*IOSYSCFG authority required to use &1.

TCP9509
Line &1 not found.

TCP9999
Internal system error in program &1.

Top

Add TCP/IP Server (ADDTCPSVR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add TCP/IP Server (ADDTCPSVR) command is used to add servers to the list of servers supported by the Start TCP/IP Server (STRTCPSVR) and End TCP/IP Server (ENDTCPSVR) commands.

Servers added with the ADDTCPSVR command can be changed with the Change TCP/IP Server (CHGTCPSVR) command or removed with the Remove TCP/IP Server (RMVTCPSVR) command.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) and all object (*ALLOBJ) special authorities to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
SVRSPCVAL	Server special value	*Character value*	Required, Positional 1
PGM	Program to call	*Qualified object name*	Required, Positional 2
	Qualifier 1: Program to call	*Name*	
	Qualifier 2: Library	*Name*	
SVRNAME	Server name	*Character value*	Required, Positional 3
SVRTYPE	Server type	*Character value*	Required, Positional 4
AUTOSTART	Autostart	*YES, ***NO**	Optional, Positional 5
TEXT	Text 'description'	*Character value*, ***BLANK**	Optional, Positional 6

Top

Server special value (SVRSPCVAL)

Specifies the special value to be used to identify the server being added. This value will be used by the Start TCP/IP Server (STRTCPSVR) and End TCP/IP Server (ENDTCPSVR) commands.

This is a required parameter.

character-value

Specify a special value (up to 10 characters) that will identify the server to be added. The following rules and restrictions apply:

- The first character must be an asterisk ('*')
- The second character must be an uppercase letter ('A-Z'), $, # or @.

- The remaining characters must be uppercase letters ('A-Z'), numbers ('0-9'), $, #, @, underscores (_), and periods (.).

Top

Program to call (PGM)

Specifies the program to be called when this server is started or ended by the following commands:

- Start TCP/IP Server (STRTCPSVR)
- End TCP/IP Server (ENDTCPSVR)
- Start TCP/IP (STRTCP) - if the server is defined as AUTOSTART(*YES)
- End TCP/IP (ENDTCP)

The following data will be passed in a single argument to the defined program:

- A 10 character field indicating the action being performed:
  - "*START " - if STRTCPSVR or STRTCP is issued.
  - "*END " - if ENDTCPSVR or ENDTCP is issued.
- A 20 character reserved field containing blanks.
- A 2 byte reserved field containing zeros.
- A 32 character field containing the instance name to be started or ended. This field is padded with blanks.
- A 4 byte binary field containing the length of the instance startup values.
- Up to 300 characters containing the instance startup values.

Note: Servers that do not support multiple instances are only required to process the first 10 characters of data which contain the action being performed.

This is a required parameter.

Qualifier 1: Program to call

name Specify the name of the program to be called when this server is started or ended.

Qualifier 2: Library

name Specify the name of library where the program is located.

Top

Server name (SVRNAME)

Specifies the textual server name that will be used by System i Navigator to display an entry for this server.

This is a required parameter.

character-value
Specify the text name of this server.

Top

Server type (SVRTYPE)

Specifies the server type that will be used by Work Management functions from System i Navigator to find job information, such as joblogs and server status, for this server.

This is a required parameter.

character-value
Specify the server type name to be used by System i Navigator to find joblog information and server status. The following rules and restrictions apply:

- Imbedded blanks or null characters are not allowed.
- The server job running on the system must **also** have the server type defined for that job. This is done by adding the server type definition using the Change Job (QWTCHGJB) API **after** the server job is started. See the Change Job (QWTCHGJB) API for more detail on how to define the server type within the server job. If the server type is not set within the server job or if the server type does not match what is defined on the SVRTYPE parameter, joblog information and server status will not be available using System i Navigator.

Top

Autostart (AUTOSTART)

Specifies whether the server being added should be started when the Start TCP/IP (STRTCP) command is run.

***NO** The server being added should not start when the STRTCP command runs.

***YES** The server being added should start when the STRTCP command runs.

Top

Text 'description' (TEXT)

Specifies a text description for the server being added.

***BLANK**
No text is specified.

character-value
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Examples

```
ADDTCPSVR   SVRSPCVAL(*XYZ)
            PGM(MYLIB/UUSTARTXYZ)
            SVRNAME('XYZ protocol server')
            SVRTYPE('XYZ')  AUTOSTART(*YES)
```

This command adds a new server to the list of servers supported by the STRTCPSVR (Start TCP/IP Server) and ENDTCPSVR (End TCP/IP Server) CL commands. The new server type can be selected on the STRTCPSVR or ENDTCPSVR commands by specifying SERVER(*XYZ). User program UUSTARTXYZ

in library MYLIB will be called when the XYZ server is started or ended. The new server will be started automatically when the STRTCPSVR command is run specifying SERVER(*AUTOSTART), or when the STRTCP (Start TCP/IP) command is run.

Top

Error messages

***ESCAPE Messages**

TCP1621
TCP/IP server &1 not added.

Top

Add Trace (ADDTRC)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Trace (ADDTRC) command specifies which program statements in a program to trace in debug mode. Up to five ranges of high-level language (HLL) statements or machine instructions can be traced during the processing of a program through one or more ADDTRC commands, and up to 10 program variables can be recorded or monitored for change in each specified statement range. A separate ADDTRC command is required for each unique variable associated with a statement range. When the specified program being traced is run, the system records the sequence in which the traced statements are processed and optionally records the value of the variables associated with the trace each time a traced statement is processed. After a trace has been completed, you can display this information using the Display Trace Data (DSPTRCDTA) command.

All of the trace ranges specified in a program are active at the same time. If both an HLL statement identifier and a machine instruction number are used to specify a given trace range, the trace range is treated as an HLL trace range. That is, in addition to tracing the machine instruction number specified, the system traces the HLL statement identifiers between that machine instruction number and the specified HLL statement identifier. More information on testing and debugging at the machine interface level is in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Restrictions:

- You can use this command only in debug mode. To start debug mode, refer to the Start Debug (STRDBG) command.
- You cannot use this command if you are servicing another job, and that job is on a job queue, or is being held, suspended, or ended.
- You cannot use this command to trace bound programs.

Top

Parameters

Keyword	Description	Choices	Notes
STMT	Statements to trace	Single values: ***ALL**, *ALLINST Other values (up to 5 repetitions): *Element list*	Optional, Positional 1
	Element 1: Starting statement identifier	*Character value*	
	Element 2: Ending statement identifier	*Character value*	
PGMVAR	Program variables	Single values: ***NONE** Other values (up to 10 repetitions): *Element list*	Optional, Positional 2
	Element 1: Program variable	*Character value*, *CHAR	
	Element 2: Basing pointer variable	Values (up to 5 repetitions): *Character value*	
OUTFMT	Output format	***CHAR**, *HEX	Optional, Positional 5
PGM	Program	*Name*, ***DFTPGM**	Optional

Keyword	Description	Choices	Notes
START	Char output start position	*Integer*, **1**	Optional, Positional 3
LEN	Characters to display	*Integer*, ***DCL**	Optional, Positional 4
OUTVAR	When output	***CHG**, *ALWAYS	Optional
TRCPGM	Trace handling program	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Trace handling program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

Statements to trace (STMT)

Specifies which program statements (or machine instructions) to trace in the program.

***ALL** All statements in the specified high-level language program are traced.

***ALLINST**
All machine instructions in the specified program are traced.

start-statement-identifier stop-statement-identifier
Specify the HLL statement identifiers (or machine instruction numbers) at which tracing starts and, optionally, the identifier at which tracing stops. A maximum of five trace ranges can be defined at the same time for any program in debug mode. Each trace range begins with the specified starting statement, and all following statements are traced until the ending statement is reached. If only a starting statement identifier is specified for a range, the single statement specified is the only statement traced for that range. If machine instruction numbers are specified, a slash must be placed in front of the number, and both the slash and the number must be enclosed in apostrophes.

In high-level language programs, different statements and/or labels can be mapped to the same internal instruction. This happens when there are several statements that do not operate on variables directly (such as DO, END, or comments) following one another in a program. To determine which statements (labels) can be mapped to the same instruction, the intermediate representation of a program listing can be used.

Top

Program variables (PGMVAR)

Specifies the names of the variables whose values are recorded when a trace statement in a program is processed. Depending on the value supplied for the **When output (OUTVAR)** parameter, the values can be recorded for every trace statement processed or only when a variable changes value. The program variables can be specified either by their high-level language names or by their machine-interface object-definition-table-vector (MI ODV) numbers.

Single values

***NONE**
No program variables have their values recorded during tracing.

Other values (up to 10 repetitions)

Element 1: Program variable

***CHAR**
This special value is specified instead of a variable name if a basing pointer is also specified. This special value displays a character view of a pointer to be shown without the use of a based variable.

character-value
Specify the names of up to ten program variables whose values are recorded during tracing. If a variable name contains special characters, it must be enclosed in apostrophes.

If the program variable is an array, the subscripts representing the elements in the array can be specified. If an array name is specified without any subscripts, all of the array elements are recorded. A single-dimensional cross-section can also be specified. Up to 132 characters may be specified for this program variable entry. This includes any qualifiers, subscripts, blanks, parentheses, or commas. It does not include the enclosing apostrophes when special characters are used. An integer, MI ODV number, asterisk (single-dimensional cross-section), or a numeric variable name can be specified for a subscript.

For more information, refer to ″Parameter values used for testing and debugging″ in ″CL concepts and reference″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Element 2: Basing pointer variable

character-value
Specify the names of up to five basing pointers for each program variable specified for element 1 of this parameter. In some languages, the program variable may be based on a pointer variable. This set of values allows you to explicitly specify the basing pointers for the variable to be recorded. Each basing-pointer name must be enclosed in apostrophes if it contains special characters.

If the basing pointer is an array, the subscripts representing an element in the array must be specified. Up to 132 characters can be specified for a basing pointer name. This includes any qualifiers, subscripts, blanks, parentheses, and commas. It does not include the enclosing apostrophes when special characters are used. An integer, machine-interface object-definition-table-vector (MI ODV) number, or a numeric variable name can be specified for a subscript.

For more information, refer to ″Parameter values used for testing and debugging″ in ″CL concepts and reference″ in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Top

Output format (OUTFMT)

Specifies the format to use for recording the variables.

***CHAR**
Variables are recorded in character form.

***HEX** Variables are recorded in hexadecimal form.

Top

Program (PGM)

Specifies the program that contains the specified statement identifiers or the machine instruction numbers to trace.

***DFTPGM**
The program previously specified as the default program contains the statements to trace.

name Specify the name of the program that contains the statements to trace. The specified program must already be in debug mode.

Top

Char output start position (START)

Specifies, for string variables only, the starting position in the string from which its value is recorded during tracing. If more than one string variable is specified for the **Program variables (PGMVAR)** parameter, the same starting position value is used for each one. For a bit string, the value specifies the starting bit position. For a character string, the value specifies the starting character position.

1 The variable is recorded from the first position on through the length specified for the **Characters to display (LEN)** parameter.

integer
Specify the first position being recorded in the program variable.

The value supplied for the **Char output start position (START)** parameter must not be larger than the maximum string length for any variable specified, except that a value of **1** for the START parameter is allowed if the maximum length for a string is zero. The value supplied for the LEN parameter plus the value supplied for the START parameter minus one, must not be greater than the maximum string length. These checks are made for each string variable specified for the PGMVAR parameter.

Top

Characters to display (LEN)

Specifies, for string variables only, the length of the string being recorded during the trace, starting at the position specified for the **Char output start position (START)** parameter. If more than one string variable is specified by the **Program variables (PGMVAR)** parameter the same value is used for each one. For a bit string, the value specifies the number of bits shown, and for a character string, the value specifies the number of characters shown.

***DCL** The string variable is shown to the end of the string or for a value of 200 bytes, whichever is less. If the string variable has a maximum length of zero, the only allowable value for the **Characters to display (LEN)** parameter is *DCL.

integer
Specify the length of the data shown. The length (as well as the combination of values supplied for the START parameter and the LEN parameter must be no greater than the length of the shortest string specified by the PGMVAR parameter.

Top

When output (OUTVAR)

Specifies whether the values of the program variables are recorded only when their values change, or whether they are recorded regardless of any of their values being changed. This parameter is ignored if *NONE is specified or assumed for the **Program variables (PGMVAR)** parameter.

Note: Within each range, the values of all the traced variables are always recorded the first time a statement in the range is processed. The value supplied for the OUTVAR parameter determines when the variables are recorded for all following statements in the range.

***CHG** The system records the values of all the program variables when one or more of the values have changed since the last trace point. A variable is considered changed not only when its value is changed, but also when any of the displayed attributes change (such as length, lower and upper boundaries, and subscript values). For example, if an array is specified and the upper boundary changes for the array, the array is considered to have changed.

Note: The value might not appear to have changed if it contains characters that cannot be shown (a value less than 40 hex). The variable is still recorded even though you cannot see the change from what is shown. If *HEX is specified for the **Output format (OUTFMT)** parameter, the changes can be observed in the traced data.

***ALWAYS**
The system records the values of the specified variables every time any of the specified trace statements are processed, whether or not any variable had its value changed.

Top

Trace handling program (TRCPGM)

Specifies the user-supplied program that is called when a statement being traced is reached in the program specified on the **Program (PGM)** parameter. The program with the traced statement passes informational parameters to the trace-handling program when it is called. These parameters identify the program name, the recursion level, the high-level language statement identifier, the machine instruction number at which the trace statement occurred, and a changed variable indicator. The parameters have the following formats:

1. Program name (10 bytes). Specifies the name of the program in which the traced statement was reached.
2. Recursion level (5 bytes). Specifies the recursion level number of the program in which the traced statement was reached. This value is a 1- to 5-digit number padded on the right with blanks.
3. Statement Identifier (10 bytes). Specifies the high-level language program statement identifier that was reached. If the traced statement does not correspond to a statement identifier, the parameter contains a slash (/) followed by a 4-digit hexadecimal machine instruction number.
4. Instruction number (5 bytes). Specifies the machine instruction number that corresponds to the high-level language statement at which the traced statement was reached. No slash appears in front of the machine instruction number. The value consists of 1 to 4 hexadecimal characters representing the MI instruction number, followed by one or more blanks. If the program passes a machine instruction number on the third parameter, the values on the third and fourth parameters will be the same.

All the parameter values are left-justified and padded on the right with blanks. When control returns to the program with the traced statement, processing continues.

When a trace-handling program is specified and OUTVAR(*CHG) is specified, the trace-handling program is called only if a program variable specified on the **Program variables (PGMVAR)** parameter is changed. No trace data is recorded.

Single values

***NONE**
No trace-handling program is called when a traced point specified on this command is reached in a batch environment. The interrupted program continues processing.

Qualifier 1: Trace handling program

name Specify the name of the user-supplied program to be called when a traced statement is reached during debugging in a batch environment. The program specified must not be the same as the program specified on the **Program** (PGM) parameter. If the same program is specified for both the TRCPGM and PGM parameters, results can be unpredictable. After the program runs, control is returned to the interrupted program and processing continues.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Examples

```
ADDTRC   STMT((100 120) (150 200))
         PGMVAR('&CTR' '&BRCTR' '&SAM')
```

This command traces program statements in the default program between the ranges of statements 100 through 120 and 150 through 200. Also, whenever the values of any of the program variables &CTR, &BRCTR, and &SAM are changed by one of the traced statements within those ranges, the values of all three are recorded before the traced statement is processed. When all of the traced statements have been processed, or when a breakpoint is reached, the Display Trace Data (DSPTRCDTA) command can be used to show the trace data collected.

Top

Error messages

***ESCAPE Messages**

CPF1999
Errors occurred on command.

Top

Add Trace Filter (ADDTRCFTR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Trace Filter (ADDTRCFTR) command adds a new trace filter to the system. A trace filter identifies the trace flow (call/return) data that is to be collected during a trace session, and is meant to limit the amount of data collected by specifying a compare value. If the data in the trace record matches the compare value, then the data will be collected. If not, the data is discarded. The filter is specified on the STRTRC (Start Trace) command.

Restrictions:

- You must have service (*SERVICE) special authority, or be authorized to the Service trace function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.

Top

Parameters

Keyword	Description	Choices	Notes
FTR	Filter	*Name*	Required, Positional 1
PGMTRG	Program trigger	*Element list*	Optional
	Element 1: Filter entry	*Qualified object name*	
	Qualifier 1: Filter entry	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**	
	Element 2: Module	*Name*	
	Element 3: Procedure	*Character value*	
	Element 4: Type	***PGM**, *SRVPGM	
	Element 5: Trigger option	***ENTRYEXIT**, *ENTRY	
JVATRG	Java trigger	*Element list*	Optional
	Element 1: Java package	*Character value*, ***NONE**	
	Element 2: Java class	*Character value*, ***ALL**	
	Element 3: Java method	*Character value*, ***ALL**	
	Element 4: Trigger option	***ENTRYEXIT**, *ENTRY, *ENTRYEXITNOPGM, *ENTRYNOPGM	

Keyword	Description	Choices	Notes
PGMFTR	Program filter	*Element list*	Optional
	Element 1: Relational operator	***EQ**, *NE	
	Element 2:	Values (up to 16 repetitions): *Element list*	
	Element 1: Filter entry	*Qualified object name*	
	Qualifier 1: Filter entry	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**	
	Element 2: Module	*Name*, ***ALL**	
	Element 3: Procedure	*Character value*, ***ALL**	
	Element 4: Type	***PGM**, *SRVPGM	
JVAFTR	Java method filter	*Element list*	Optional
	Element 1: Relational operator	***EQ**, *NE	
	Element 2:	Values (up to 5 repetitions): *Element list*	
	Element 1: Java package	*Character value*	
	Element 2: Java class	*Character value*, ***ALL**	
	Element 3: Java method	*Character value*, ***ALL**	

Top

Filter (FTR)

Specifies the trace filter to be added. If the specified filter already exists, an error message is issued. An existing trace filter can be removed by using the Remove Trace Filter (RMVTRCFTR) command.

This is a required parameter.

name Specify the name of the new trace filter.

Top

Program trigger (PGMTRG)

If a procedure is called that matches this trigger specification, then STRTRC will begin collecting flow trace records for this STRTRC session. The flow trace records will be collected only for the thread where the trigger occurs.

When the procedure returns and it matches the trigger specification, STRTRC will stop collecting the flow trace records.

Element 1: Filter entry

Qualifier 1: Filter entry

name Specify the name of the program which, if called, will cause trace flow records to start being collected.

Qualifier 2: Library

***LIBL** The library list of the job that issues the STRTRC command is searched to find the specified program or service program.

name Specify the name of the library which contains the program or service program.

Element 2: Module

name Specify the name of the module within the program or service program that contains the procedure that is to be the trigger.

Element 3: Procedure

character-value
Specify the name of a specific procedure within the specified module that is to be the trigger. Specify the procedure name within single quotes if the procedure name contains lower case characters.

Element 4: Type

***PGM** The program being specified is a program (*PGM) object.

***SRVPGM**
The program being specified is a service program (*SRVPGM) object.

Element 5: Trigger option

***ENTRYEXIT**
The specified trigger procedure enables the collection of flow trace records at procedure entry time. At procedure exit, the collection of flow trace records is disabled.

***ENTRY**
The specified trigger procedure enables the collection of flow trace records at procedure entry time. The collection of flow trace records continues for the duration of the trace session.

Top

Java trigger (JVATRG)

If a Java method entry event (*JVAENTRY) occurs that matches this trigger specification, then STRTRC will begin collecting all events for this STRTRC session. The events will be collected only for the thread where the trigger occurs. When the Java method exit event (*JVAEXIT) occurs that matches the trigger specification, STRTRC will stop collecting the events.

The *JVAENTRY and *JVAEXIT hooks are always enabled in interpreted Java. You can also enable these hooks when running in JIT mode by specifying the property 'os400.enbpfrcol=1'.

For code compiled for direct execution, specify ENBPFRCOL(*ENTRYEXIT) on the Create Java Program (CRTJVAPGM) CL command.

Element 1: Java package

package-name
Specify the name of the Java package that contains the method to use as the trigger.

Element 2: Java class

class-name
Specify a class within the package that contains the method to use as the trigger.

Element 3: Java method

method-name
Specify a method to use as the trigger.

Element 4: Trigger option

***ENTRYEXIT**
The specified trigger method enables the collection of events at Java method entry time. At Java method exit, the collection of events is disabled.

***ENTRY**
The specified trigger method enables the collection of events at Java method entry time. The collection of events is enabled for the duration of the STRTRC session.

***ENTRYEXITNOPGM**
The specified trigger method enables the collection of events (except program events) at method entry time. At method exit, the collection of events is disabled. This option is not valid for STATS mode.

***ENTRYNOPGM**
The specified trigger method enables the collection of events (except program events) at method entry time. The collection of events is enabled for the duration of the STRTRC session. This option is not valid for STATS mode.

Top

Program filter (PGMFTR)

Specifies the program comparisons to use for this filter.

Element 1: Relational operator

***EQ** Flow trace records having program data that matches the specified program are included in the data collected.

***NE** Flow trace records having program data that matches the specified program are excluded from the data collected. These trace records will not show up.

Element 2:

You can specify up to 16 values for this element set.

Element 1: Filter entry

Qualifier 1: Filter entry

name Specify the name of the program to be used as a compare value for the program filter.

Qualifier 2: Library

***LIBL** The library list of the job that issues the STRTRC command is searched to find the specified program or service program.

name Specify the name of the library which contains the program or service program.

Element 2: Module

***ALL** All modules in the program or service program will pass the filter. If filtering an OPM (Original Program Model) program, specify *ALL for this element.

name Specify a specific module within the program or service program to be used as a compare value for the program filter.

Element 3: Procedure

***ALL** All procedures in the specified module are used as a compare value for the program filter.

character-value
Specify a procedure to use as the filter compare value. Specify the procedure name within single quotes if the procedure name contains lower case characters.

Element 4: Type

***PGM** The program being specified is a program (*PGM) object.

***SRVPGM**
The program being specified is a service program (*SRVPGM) object.

Top

Java method filter (JVAFTR)

Specifies the Java package, class, and methods to be used as compare values for the Java filter.

Element 1: Relational operator

***EQ** Flow trace records having Java data that match the specified packages, classes, and methods are included in the data collected.

***NE** Flow trace records having Java data that matches the specified packages, classes, and methods are excluded from the collection and will not show up.

Element 2:

You can specify up to 5 values for this element set.

Element 1: Java package

character-value
Specify the name of the Java package to be used as a compare value for the filter.

Element 2: Java class

***ALL** All classes in the specified package will pass the Java filter.

character-value
Specify the name of a class within the package to be used as a compare value for the filter.

Element 3: Java method

***ALL** All methods in the specified class and package will pass the filter.

character-value
Specify the name of a method to use as the filter compare value.

Top

Examples

Example 1: Adding a Trace Filter for a Program Trigger

```
ADDTRCFTR   FTR(PGMFTR)
            PGMTRG(MYLIB/MYPGM MYMODL *PEP *PGM *ENTRY)
```

This command adds a new trace filter named PGMFTR. If this filter is used in the Start Trace (STRTRC) command, the collection of data will begin when the program entry of MYMODL module of MYPGM is called.

Top

Error messages

Unknown

Top

Add Work Station Entry (ADDWSE)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Add Work Station Entry (ADDWSE) command adds a work station job entry to the specified subsystem description. Each entry describes one or more work stations that are controlled by the subsystem. The work stations identified in the work station entries are allowed to sign on or enter the subsystem and run jobs.

Restrictions:

1. To use this command, you must have:
   - object operational (*OBJOPR), object management (*OBJMGT), and read (*READ) authority to the specified subsystem description and execute (*EXECUTE) authority to the library containing that subsystem description.
   - object operational (*OBJOPR) and read (*READ) authority to the job description and execute (*EXECUTE) authority to the library containing that job description.
2. Only a user with all object (*ALLOBJ) special authority is allowed to add an entry for which the job description does not exist.

Top

Parameters

Keyword	Description	Choices	Notes
SBSD	Subsystem description	*Qualified object name*	Required, Positional 1
	Qualifier 1: Subsystem description	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
WRKSTN	Work station name	*Generic name, name*	Optional, Positional 2
WRKSTNTYPE	Work station type	*ALL, 3179, 3180, 3196, 3197, 3277, 3278, 3279, 3476, 3477, 3486, 3487, 5251, 5291, 5292, 5555, *ASCII, CONS, *CONS, *NONASCII	Optional, Positional 3
JOBD	Job description	Single values: ***USRPRF**, *SBSD Other values: *Qualified object name*	Optional, Positional 4
	Qualifier 1: Job description	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MAXACT	Maximum active jobs	0-1000, ***NOMAX**	Optional
AT	Allocation	***SIGNON**, *ENTER	Optional

Top

Subsystem description (SBSD)

Specifies the name and library of the subsystem description to which the work station job entry is added.

This is a required parameter.

Qualifier 1: Subsystem description

name Specify the name of the subsystem description where the work station job entry is being added.

Note: The following IBM-supplied objects are not valid on this parameter:

- QLPINSTALL
- QSYSSBSD

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the library where the subsystem description is located.

Top

Work station name (WRKSTN)

Specifies the name of the work station used by the subsystem. The device description name that was specified in the Create Device Desc (Display) (CRTDEVDSP) command associated with the work station is the name used.

Double-Byte Character Set Considerations: For double-byte character set (DBCS), a work station whose type is 5555 must be specified for either this parameter or the **Work station type (WRKSTNTYPE)** parameter, but not for both.

generic-name
Specify a generic name. Examples include: DSP*, RMT*,...

Note: Specifying a generic work station name does not result in multiple entries being added, changed, or removed.

name Specify the name of a specific work station. Examples include: DSP10, DSP11, RMT55,...

A value must be specified on either this parameter or the **Work station type (WRKSTNTYP)** parameter, but not for both.

Top

Work station type (WRKSTNTYPE)

Specifies the type of work station associated with the entry being added, changed, or removed. This entry applies to all work stations of this type that do not have specific entries for an individual work station.

***ALL** All work station devices. This includes devices with 5250, ASCII, and 327x device types.

3179 3179 work station.

3180 3180 work station.

3196 3196 work station.

3197 3197 work station.

3277 3277 work station.

3278 3278 work station.

3279 3279 work station.

3476 3476 work station.

3477 3477 work station.

3486 3486 work station.

3487 3487 work station.

5251 5251 work station.

5291 5291 work station.

5292 5292 color work station.

5555 5555 double-byte character set (DBCS) capable work station.

***ASCII**
All ASCII work station device types.

CONS System console display. This entry overrides a device type entry that specifies the same device type as the device being used as the console.

***CONS**
System console display. This entry overrides a device type entry that specifies the same device type as the device being used as the console.

***NONASCII**
All work station devices that use the 5250 data stream, as well as, 327x device types.

A value must be specified on either this parameter or the **Work station name (WRKSTN)** parameter, but not for both.

Top

Job description (JOBD)

Specifies the name and library of the job description used for jobs started through this work station entry. If the job description does not exist when the entry is added, a library qualifier must be specified because the qualified job description name is kept in the subsystem description.

Note: Only a user with all object (*ALLOBJ) special authority is allowed to add or change an entry for which the job description does not exist.

Single values

***USRPRF**
The job description named in the user profile that is used to sign on at this work station (or at this type of work station) is used for jobs started through this entry.

***SBSD**
The job description having the same name as the subsystem description, specified on the **Subsystem description (SBSD)** parameter, is used for jobs started through this entry.

Qualifier 1: Job description

name Specify the name of the job description.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the library where the job description is located.

Top

Maximum active jobs (MAXACT)

Specifies, for work stations that use this work station job entry, the maximum number of work station jobs that can be active at the same time.

***NOMAX**
There is no maximum number of jobs (work stations) that can be active at the same time through this work entry.

0-1000 Specify the maximum number of jobs that can be active at the same time through this work entry.

Top

Allocation (AT)

Specifies how the work stations associated with this job entry are allocated. For more information on how work stations are allocated to subsystems, see the Start Subsystem (STRSBS) command.

***SIGNON**
The work stations are allocated when the subsystem is started if the work station is not already in use (signed on) in another subsystem. A sign-on prompt is displayed at each work station associated with this work entry. If a work station becomes allocated to a different subsystem, interactive jobs associated with the work station are allowed to enter this subsystem through the Transfer Job (TFRJOB) command.

***ENTER**
The work stations associated with this work entry are not allocated when the subsystem is started. However, the interactive jobs associated with the work stations are allowed to enter this subsystem through the TFRJOB command.

Top

Examples

Example 1: Adding a Work Station Job Entry

```
ADDWSE   SBSD(LIB7/ORDER)  WRKSTNTYPE(5251)  JOBD(QCTL)  AT(*SIGNON)
```

This command adds a work station job entry to a subsystem description named ORDER in library LIB7. All type 5251 work stations are allocated to this subsystem when the subsystem is started, unless they are already active in a previously started subsystem. After sign-on is complete, the IBM-supplied job description QCTL is used to start the routing step.

Example 2: Adding a Work Station Job Entry

```
ADDWSE   SBSD(LIB7/ORDER)  WRKSTN(A12)  JOBD(LIB7/ORDER)  AT(*ENTER)
```

This command adds a work station job entry for work station A12 to a subsystem description named ORDER in library LIB7. Interactive jobs associated with work station A12 are allowed to enter this subsystem through the Transfer Job (TFRJOB) command.

Top

Error messages

***ESCAPE Messages**

CPF1619
Subsystem description &1 in library &2 damaged.

CPF1691
Active subsystem description may or may not have changed.

CPF1697
Subsystem description &1 not changed.

Top

Allocate Object (ALCOBJ)

Where allowed to run: All environments (*ALL)
Threadsafe: Conditional

Parameters
Examples
Error messages

The Allocate Object (ALCOBJ) command is used in a job or thread to reserve an object or list of objects for use later in the job or thread. If an object that is needed in the job is not specified in an ALCOBJ command, an allocation is attempted automatically when the object is used.

Objects can be deallocated with the Deallocate Object (DLCOBJ) command. Allocated job-scoped locks are automatically released when the job ends. Allocated thread-scoped locks are automatically released when the thread ends. If a thread received a job-scoped lock, the job will continue to hold that lock after the requesting thread ends. Lock-space-scoped locks are not automatically released.

The DLCOBJ command should not be issued for an object that was not explicitly allocated by the ALCOBJ command. If the DLCOBJ command is used this way, internal locks on the object are released, making the object capable of being deleted.

NOTES:

1. When allocating database files, use the DLCOBJ command before deleting the file if the file being allocated is a logical file.
2. If a file is being allocated that is affected by a file override, the ALCOBJ command ignores the override and attempts to allocate the file named in the OBJ parameter.
3. When allocating distributed data management (DDM) files and distributed files, additional time is required for the command to complete because of the time required for communication and for allocating files on remote systems.
4. Work station message queues cannot be allocated. A work station message queue is associated with a work station device description of the same name. Therefore, to do an operation on a work station message queue that must be allocated, the user must allocate the associated device description. When the device description is allocated, the work station message queue is implicitly allocated.
5. When ALCOBJ is executed to get an EXCL lock on a program (*PGM), only the program object description is locked. The program code is not locked exclusively. Therefore, the program may still be run by another user. Changes are not allowed for the program object description while the actual program can still be used.
6. The system does not lock programs when calling them.
7. When ALCOBJ is executed to get an EXCL lock on a logical file member (*FILE), the lock occurs on both the logical file member and the associated physical file members. No other user can use the physical file members (not even through some other logical file member).
8. Allocating an object by specifying *LIBL for the object's library, changing the thread's library list, and then attempting to deallocate the object by specifying *LIBL for the object's library can result in issuing the deallocate against the wrong object. This could release internal locks.

Restrictions:

1. This command cannot be used to allocate a device description, *DEVD, for an advanced program-to-program communications (APPC) device or for an intrasystem (INTRA) device.
2. This command can be used to allocate only the following database *FILE types:
   - Physical files
   - Logical files
   - Distributed files

This allocates the piece of the file on each node in the node group.

- DDM files

  This allocates both the DDM file on the local system and the file on the remote system that is identified in the DDM file.

3. The object must exist on the system.
4. The user issuing the command must have object operational (*OBJOPR) authority to the object and execute (*EXECUTE) authority to the object's library.
5. If the allocation cannot be completed, none of the locks are granted, and a message is sent to the thread that issued the command. If the command is issued from a program, the Monitor Message (MONMSG) command can be used to determine whether the allocation was successful.
6. In multithreaded jobs, this command is not threadsafe for distributed files. This command is also not threadsafe for distributed data management (DDM) files of type *SNA.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object specifications	Values (up to 50 repetitions): *Element list*	Required, Positional 1
	Element 1: Object	*Qualified object name*	
	Qualifier 1: Object	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 2: Object type	*AUTL, *BNDDIR, *CLD, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DEVD, *DTAARA, *DTADCT, *DTAQ, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *IMGCLG, *IPXD, *LIB, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGQ, *NODL, *NTBD, *NWSCFG, *NWSD, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PSFCFG, *QMFORM, *QMQRY, *QRYDFN, *SBSD, *SCHIDX, *SQLPKG, *SRVPGM, *SSND, *S36, *TIMZON, *USRIDX, *USRQ, *USRSPC, *VLDL, *WSCST	
	Element 3: Lock state	*SHRRD, *SHRNUP, *SHRUPD, *EXCLRD, *EXCL	
	Element 4: Member, if data base file	*Name*, *FIRST	
WAIT	Wait time	*Integer*, ***CLS**	Optional, Positional 2
SCOPE	Lock scope	***JOB**, *THREAD, *LCKSPC	Optional
CONFLICT	Lock conflict action	***NORQSRLS**, *RQSRLS	Optional

Top

Object specifications (OBJ)

Specifies the name and library of one or more objects that are allocated to the job, thread, or lock space, the type of each object specified, the lock state of each object, and the member name (if the object is a database file or distributed data management (DDM) file).

This is a required parameter.

You can specify 50 values for this parameter.

Element 1: Object

Qualifier 1: Object

name Specify the name of the object.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the name of the library where the object is located.

Element 2: Object type

object-type
Specify the type of object to be allocated. Refer to the figure Valid Lock States by Object Type for more information.

Element 3: Lock state

***SHRRD**
The lock state is shared for read.

***SHRNUP**
The lock state is shared, no update.

***SHRUPD**
The lock state is shared for update.

***EXCLRD**
The lock state is exclusive, allow read.

***EXCL**
The lock state is exclusive, no read.

Element 4: Member, if data base file

Note: The following values can only be specified if the object type is a database file.

***FIRST**
The first member of the database file is allocated.

name Specify the name of the member to be allocated. If the specified file is a logical file, the physical file members associated with the members of the logical file are also allocated.

You can specify all five lock states (*EXCL, *EXCLRD, *SHRUPD, *SHRNUP, and *SHRRD) for most, but not all, object types.

Table 2. **Figure: Valid Lock States by Object Type**

Object Type	Object Type Definition	Lock States *EXCL	*EXCLRD	*SHRUPD	*SHRNUP	*SHRRD
*AUTL	Authorization List	x	x	x	x	x
*BNDDIR	Binding directory	x	x			x
*CLD	C Locale description	x	x	x	x	x
*CRQD	Change request description	x	x	x	x	x
*CSI	Communications side information	x	x	x	x	x
*DEVD	Device Description		x	x		
*DTAARA	Data area	x	x	x	x	x
*DTADCT	Data dictionary	x	x	x	x	x
*DTAQ	Data queue	x	x	x	x	x
*FCT	Forms control table	x	x	x	x	x
*FILE	File	x	x	x	x	x
*FNTRSC	Font resource	x	x	x	x	x
*FNTTBL	Font mapping table	x	x	x	x	x
*FORMDF	Form definition	x	x	x	x	x
*IMGCLG	Image catalog	x	x	x	x	x
*IPXD	Internet packet exchange description	x	x	x	x	x
*LIB	Library		x	x	x	x
*LOCALE	Locale space object	x	x	x	x	x
*MEDDFN	Media definition	x	x	x	x	x
*MENU	Menu	x	x	x	x	x
*MGTCOL	Management collection	x	x	x	x	x
*MODULE	Module	x	x			x
*MSGQ	Message queue	x				x
*NODL	Node list	x	x	x	x	x
*NTBD	NetBIOS description	x	x	x	x	x
*NWSCFG	Network server configuration	x	x	x	x	x
*NWSD	Network server description	x	x	x	x	x
*OVL	Overlay	x	x	x	x	x
*PAGDFN	Page definition	x	x	x	x	x
*PAGSEG	Page segment	x	x	x	x	x
*PDFMAP	PDF Map	x	x	x	x	x
*PDG	Print descriptor group	x	x	x	x	x
*PGM	Program	x	x			x
*PNLGRP	Panel group	x	x	x	x	x
*PSFCFG	Print service facility configuration	x	x	x	x	x
*QMFORM	Query management form	x	x	x	x	x
*QMQRY	Query management query	x	x	x	x	x
*QRYDFN	Query definition	x	x	x	x	x
*S36	S/36 machine description	x	x	x	x	x
*SBSD	Subsystem description	x				
*SCHIDX	Search index	x	x	x	x	x
*SQLPKG	Structured Query Language package	x	x	x	x	x
*SRVPGM	Service program	x	x	x	x	x
*SSND	Session description	x	x	x	x	x
*TIMZON	Time zone description	x	x	x	x	x
*USRIDX	User index	x	x	x	x	x
*USRQ	User queue	x	x	x	x	x
*USRSPC	User space	x	x	x	x	x
*VLDL	Validation list object	x	x	x	x	x
*WSCST	Workstation customizing object	x	x	x	x	x

'x' indicates the lock state is allowed for the object type.

Note: Additional details about lock states can be found in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Multiple locks can be specified for the same object in the same job with duplicate or different lock states. Each lock is held separately. For example, if an *EXCL lock is already held for an object, and a second *EXCL lock request occurs, the second lock is acquired. Both locks must be released in the job (deallocated with the Deallocate Object (DLCOBJ) command) before another job can access the same object. If an object is already allocated with one lock state and user want to use a different lock state, first use the ALCOBJ command to request the new lock with the desired lock state and then use the DLCOBJ command to release the old lock (with the old lock state).

When an exclusive lock is requested on a logical file member, the lock occurs on both the logical file member and the associated physical file members. No other user can use the physical file members (not even through some other logical file member).

To determine whether a device description can be allocated, use information from the Work with Configuration Status (WRKCFGSTS) command.

Top

Wait time (WAIT)

Specifies the number of seconds that the program waits for the object to be allocated. If the object cannot be allocated in the specified wait time, a message, which can be detected by a Monitor Message (MONMSG) command, is sent to the program. If one or more device descriptions are in the list of objects to be allocated, the system may wait more than the specified amount of time to attempt the allocation.

When allocating distributed data management (DDM) files and distributed files, additional time is required for communications and for allocating files on remote systems. A separate wait time is used for each remote system. When allocating objects with a lock-space-scope, the lock space may override the wait time specified.

***CLS** The default wait time specified in the class description used by the routing step is used as the wait time for the object to be allocated.

integer-number
Specify the number of seconds that the program waits for all of the specified objects to be allocated. Valid values include 0 and numbers ranging from 30 through 32767. A value of 0 indicates no wait time.

Top

Lock scope (SCOPE)

Specify the scope for this lock request.

***JOB** The lock is scoped to the job.

***LCKSPC**
The lock is scoped to the lock space attached to the current thread. If no lock space is attached, the lock is scoped to the job.

***THREAD**
The lock is scoped to the thread.

All objects types supported by the OBJ parameter support job-scoped locks. All object types supported by the OBJ parameter support lock-space-scoped locks. When allocating DDM objects with a lock-space-scope, the lock on the remote system is scoped to the job.

Locks scoped to a thread can never conflict with a lock scoped to its containing job, but may conflict with a lock scoped to a different job or any other thread (depending on the lock states involved).

Table 3. **Figure: Object Types that Support Thread Scope Locks**

Object Type	Object Type Definition	Thread Scope
*AUTL	Authorization list	
*BNDDIR	Binding directory	
*CLD	C Locale description	
*CRQD	Change request description	
*CSI	Communications side information	
*DEVD	Device Description	x
*DTAARA	Data area	x
*DTADCT	Data dictionary	x
*DTAQ	Data queue	x
*FCT	Forms control table	
*FILE	File	x
*FNTRSC	Font resource	
*FNTTBL	Font mapping table	
*FORMDF	Form definition	
*IMGCLG	Image catalog	x
*IPXD	Internet packet exchange description	x
*LIB	Library	x
*LOCALE	Locale space object	x
*MEDDFN	Media definition	
*MENU	Menu	
*MGTCOL	Management collection	x
*MODULE	Module	
*MSGQ	Message queue	x
*NODL	Node list	
*NTBD	NetBIOS description	x
*NWSCFG	Network server configuration	
*NWSD	Network server description	x
*OVL	Overlay	
*PAGDFN	Page definition	
*PAGSEG	Page segment	
*PDFMAP	PDF Map	x
*PDG	Print descriptor group	
*PGM	Program	x
*PNLGRP	Panel group	
*PSFCFG	Print service facility configuration	
*QMFORM	Query management form	
*QMQRY	Query management query	
*QRYDFN	Query definition	
*S36	S/36 machine description	
*SBSD	Subsystem description	x
*SCHIDX	Search index	
*SQLPKG	Structured Query Language package	
*SRVPGM	Service program	x
*SSND	Session description	
*TIMZON	Time zone description	x
*USRIDX	User index	x
*USRQ	User queue	x
*USRSPC	User space	x
*VLDL	Validation list object	x
*WSCST	Workstation customizing object	

'x' indicates a thread-scoped lock is allowed for the object type.

Top

Lock conflict action (CONFLICT)

Specify the action to be taken if a lock conflict exists. This parameter is only supported for database files and is ignored for all other objects. The supported database *FILE objects are:

- physical file
- logical file
- distributed file

***NORQSRLS**
No requests are sent to other jobs or threads which are holding conflicting locks.

***RQSRLS**
A request is sent to the system code running in each job or thread that is holding a conflicting lock for the specified object. Notification of lock contention is not visible to user applications which hold conflicting locks. Only locks which are acquired implicitly by system code are eligible to be released. Locks acquired explicitly by user application code are not eligible to be released. If *RQSRLS specified for a distributed file, the request to release the lock is sent to each node in the node group that holds a conflicting lock.

Top

Examples

Example 1: Allocate File for Job

```
ALCOBJ   OBJ((LIBB/FILEA  *FILE *EXCL MEMBERA))  SCOPE(*JOB)
         WAIT(60)
```

This command exclusively allocates member MEMBERA of file FILEA in library LIBB to the job in which the ALCOBJ command is used. If MEMBERA is unavailable, the job will wait 60 seconds for it to become available.

Example 2: Allocate Data Area for a Thread

```
ALCOBJ   OBJ((LIBY/DATAAREAX  *DTAARA  *EXCL ))
         SCOPE(*THREAD)
```

This command exclusively allocates data area DATAAREAX in library LIBY to the requesting thread in which the ALCOBJ command is used.

Example 3: Allocate File for Lock Space

```
ALCOBJ   OBJ((LIBB/FILEA  *FILE *EXCL MEMBERA))
         SCOPE(*LCKSPC)
```

This command exclusively allocates member MEMBERA of file FILEA in library LIBB to the lock space attached to the current thread. If no lock space is attached, the lock is scoped to the job.

Top

Error messages

***ESCAPE Messages**

CPF1002
Cannot allocate object &1.

CPF1040
Maximum number of objects allocated on system.

CPF1085
Objects not allocated.

Top

Answer Line (ANSLIN)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Answer Line (ANSLIN) command prepares the system for an incoming call on a line that is not configured for auto-answer. When a call arrives, the system operator manually answers, validates the requirements of the caller, and then enters the Answer Line (ANSLIN) command. When the system is ready to receive the call, a message tells the operator to **select data mode on the modem**.

Top

Parameters

Keyword	Description	Choices	Notes
LINE	Line	*Name*	Required, Positional 1

Top

Line (LINE)

Specifies the communications line the call is coming in on.

This is a required parameter.

line-description-name
Specify the name of the communications line description.

Top

Examples

```
ANSLIN   LINE(LINE01)
```

This command answers an incoming call on a line named LINE01.

Top

Error messages

***ESCAPE Messages**

CPF2704
Line description &1 not found.

CPF5914
Answer Line (ANSLIN) command for line &1 failed.

CPF5915
Line &23 not in a valid state for answering.

CPF5917
Not authorized to line description &1.

CPF5919
Line &1 not available.

CPF5935
Error occurred during command processing.

CPF5938
Another job using line &1.

CPF5939
Another job using line &1.

CPF5945
ANSLIN command not valid for line &1.

Top

Answer Questions (ANSQST)

Where allowed to run: Interactive environments (*INTERACT *IPGM *IREXX *EXEC)
Threadsafe: No

Parameters
Examples
Error messages

The question-and-answer (Q & A) database coordinator uses the Answer Questions (ANSQST) command to display and answer questions asked by users of a Q & A database. More information is available in the Basic system operations topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. A user must have authority to the command and be a Q & A coordinator for any Q & A database referred to by the command.
3. This command can only be used interactively.

Top

Parameters

Keyword	Description	Choices	Notes
QSTDB	Q/A database	*Name*, ***SELECT**	Optional, Positional 1
LIB	Lib containing Q/A database	*Name*, ***QSTLIB**	Optional, Positional 2

Top

Q/A database (QSTDB)

Specifies the Q & A database with which to show and answer questions.

The possible values are:

***SELECT**
You are asked to specify a Q & A database. If only one Q & A database exists on the system, it is the default.

question-database
Specify the name of the Q & A database with which to display and answer questions.

Top

Lib containing Q/A database (LIB)

Specifies the name of the library that contains the Q & A database.

The possible library values are:

***QSTLIB**
The library containing the specified Q & A database is searched. If *SELECT is specified on the QSTDB parameter, any Q & A database in any library to which you are authorized can be selected.

library-name
Specify the name of the library to be searched. If *SELECT is specified on the QSTDB parameter, any database in the library to which you are authorized can be selected.

Top

Examples

```
ANSQST
```

This command shows the Select Question Status display. If more than one Q & A database is available for selection, the Select Q and A Database display is shown first.

Top

Error messages

None

Top

Analyze Command Performance (ANZCMDPFR)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Analyze Command Performance (ANZCMDPFR) command measures the performance of a single CL command or a set of CL commands. For a single command, performance information is collected from when the command starts until the command ends. For a set of commands, performance information is collected from the start of the first command in the set until the last command in the set ends.

Top

Parameters

Keyword	Description	Choices	Notes
CMD	Command	*Command string*	Optional
CMDFILE	Command file	*Qualified object name*	Optional
	Qualifier 1: Command file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
CMDMBR	Command member	*Name*, ***FIRST**	Optional
OUTPUT	Output	***MSG**, *OUTFILE	Optional
OUTFILE	File to receive output	*Qualified object name*	Optional
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member to receive output	*Name*, ***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	

Top

Command (CMD)

Specifies a CL command whose performance is to be analyzed.

Note: A value must be specified for either this parameter or the **Command file (CMDFILE)** parameter. You cannot specify a value for both the CMD and CMDFILE parameters.

command-string
Specify a syntactically-correct CL command string.

Top

Command file (CMDFILE)

Specifies the database source file that contains the source member that contains the CL commands whose performance is to be analyzed.

Note: A value must be specified for either this parameter or the **Command (CMD)** parameter. You cannot specify a value for both the CMD and CMDFILE parameters.

Qualifier 1: Command file

name Specify the name of the file to be used.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the thread is used to locate the file. If no current library entry exists in the library list, the QGPL library is used.

name Specify the name of the library where the file is located.

Top

Command member (CMDMBR)

Specifies the database source file member that contains the CL commands whose performance is to be analyzed.

***FIRST**
The first member of the database source file specified for the **Command file (CMDFILE)** parameter contains the CL commands to be run.

name Specify the name of the file member.

Top

Output (OUTPUT)

Specifies where the output from the command is sent.

***MSG** The output is displayed in a message (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***OUTFILE**
The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Top

File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library.

Note: If a new file is created, the text describing that file is ″Output file for ANZCMDPFR″ and the public authority is the same as the create authority specified for the library in which the file is created.

Use the Display Library Description (DSPLIBD) command to show the library's create authority. The database format (QAPYANZR) of the output file is the same as that used in the IBM-supplied database file QAPYANZC.

Qualifier 1: File to receive output

name Specify the name of the file to which the output of the command is directed.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found.

***CURLIB**
The current library for the job is searched to find the file. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched to find the file.

Top

Output member options (OUTMBR)

Specifies the name of the database file member that receives the output of the command.

Element 1: Member to receive output

***FIRST**
The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter. If the member already exists, you have the option to add new records to the end of the existing member or clear the member and then add the new records.

name Specify the name of the file member that receives the output. If it does not exist, the system creates it.

Element 2: Replace or add records

***REPLACE**
The system clears the existing member and adds the new records.

***ADD** The system adds the new records to the end of the existing records.

Top

Examples

Example 1: Analyzing Performance of a Single Command

```
ANZCMDPFR   CMD(CRTLIB LIB(TST))
```

This command analyzes the performance of the CRTLIB LIB(TST) command. The results are sent as messages.

Example 2: Analyzing a Command and Writing Results to a File

```
ANZCMDPFR   CMD(DSPLIB LIB(QGPL))  OUTPUT(*OUTFILE)
            OUTFILE(MYLIB/MYFILE)
```

This command analyzes the performance of the DSPLIB LIB(QGPL) command. The results are written to the first member of database file MYFILE in library MYLIB.

Example 3: Analyzing Performance of a Set of CL Commands

```
ANZCMDPFR   CMDFILE(MYLIB/MYCMDFILE)
            CMDMBR(CMDS)
```

This command analyzes the performance of the set of CL commands located in the source member named CMDS in file MYCMDFILE in library MYLIB. The results are sent as messages.

Top

Error messages

***ESCAPE Messages**

CPF4104
User not authorized to operation on file &2 in &3, member, device, or program device &4.

CPF4128
Not able to allocate objects needed for file &2 in library &3 member or program device &4.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

Top

Analyze Default Passwords (ANZDFTPWD)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Analyze Default Passwords (ANZDFTPWD) command allows you to print a report of all the user profiles on the system that have a default password and to take an action against the profiles. A profile has a default password when the profile's password matches the user profile name.

When the system is operating at password level 2 or 3, both the uppercase and lowercase values of the user profile name are checked. However, mixed case values of the user profile name will not be checked. For example, if the user profile JAMES has a password of 'JAMES' or 'james' it will be detected as having a default password; but passwords of 'JaMeS' or 'James' will not be detected as default passwords.

Restriction: You must have *ALLOBJ and *SECADM special authorities to use this command.

The format of the report depends on what action is taken against the profiles. When no action is taken, each entry will contain the user profile name, the user profile's status (STATUS), whether the profile's password is expired (PWDEXP), and the text description associated with the profile (TEXT). When an action is taken against the profiles, each entry will also contain the user profile's STATUS and PWDEXP values after the profile has been changed.

The list of user profiles with default passwords is also put in the system file QASECPWD in library QUSRSYS. Each entry contains the user profile name, the user profile STATUS and PWDEXP values before and after the profile is changed, and the user profile TEXT value. If no action was requested, the second set of STATUS and PWDEXP values will be blank.

Top

Parameters

Keyword	Description	Choices	Notes
ACTION	Action taken against profiles	Single values: ***NONE** Other values (up to 2 repetitions): *DISABLE, *PWDEXP	Optional

Top

Action taken against profiles (ACTION)

The action to be taken against the user profiles that have a default password.

***NONE**
No action is taken against profiles with a default password.

***DISABLE**
The user profile STATUS field is set to *DISABLED.

***PWDEXP**
The user profile PWDEXP field is set to *YES.

Top

Examples

```
ANZDFTPWD   ACTION(*DISABLE *PWDEXP)
```

This command analyzes all user profiles on the system. **Any user profiles on the system that have a default password will be disabled and their passwords will be set to expired.**

Top

Error messages

***ESCAPE Messages**

CPFB301
Cannot open file &2 in library &3.

CPFB302
Not authorized to check for default passwords.

Top

Analyze Java Program (ANZJVAPGM)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Analyze Java Program (ANZJVAPGM) command analyzes a Java program, lists its classes and shows the current status of each class.

Restrictions: The file must be in one of the following file systems: QOpenSys,"root", or a user-defined file system.

Top

Parameters

Keyword	Description	Choices	Notes
CLSF	Class file or JAR file	*Path name*	Required, Positional 1
CLASSPATH	Classpath	*Path name*, ***PGM**, *ENVVAR	Optional
JDKVER	Java developer kit version	*Character value*, ***PGM**, *CURRENT	Optional
DETAIL	Detail	***NONCURRENT**, *FULL	Optional
OUTPUT	Output	*, *PRINT	Optional

Top

Class file or JAR file (CLSF)

Specifies the class file , JAR file, or ZIP file name which has a Java program to analyze. The file name may be qualified by one or more directory names.

class-file-name
Specify the name of the class file. If the name is qualified it must be enclosed in apostrophes. An example of a qualified class file name is '/directory1/directory2/myclassname.class'.

JAR-file-name
Specify the name of the Java archive (JAR) file. A file is assumed to be a JAR file if the file name ends with '.jar' or '.zip'. If the name is qualified it must be enclosed in apostrophes. An example of a qualified JAR file name is '/directory1/directory2/myappname.jar'.

Top

Classpath (CLASSPATH)

Specifies the path used to locate classes referenced by the Java program being analyzed. Directories are separated by colons.

***PGM** The class path to use is the same one used to create the Java program being analyzed.

***ENVVAR**
The class path is determined by the environment variable CLASSPATH.

class-path
Path used to locate classes. An example class path is '/directory1/directory2:/QIBM/ProdData/Java400'.

Top

Java developer kit version (JDKVER)

Specifies the Java Development Kit (JDK) version to use when analyzing the Java program.

***PGM** The JDK version path to use is the same one used to create the Java program being analyzed.

***CURRENT**
The JDK version currently installed as the system default is to be used.

Java-Development-Kit-version
The JDK version to be used. An example JDK version is '1.2.2'.

Top

Detail (DETAIL)

Specifies the amount of detail to show.

***NONCURRENT**
Show only those classes which are not current with respect to the given classpath.

***FULL** Show all classes included in the Java stand-alone program. A status of **current** or **not current** will be shown for each class.

Top

Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting work station or printed with the job's spooled output.

***** Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

***PRINT**
The output is printed with the job's spooled output.

Top

Examples

```
ANZJVAPGM   CLSF('/projectA/myJavaclassname.class')
            DETAIL(*FULL)  OUTPUT(*)
```

This command will analyze the Java program associated with the class file myJavaclassname. The status for all classes included in the Java program will be listed. The Java program information will be output to the display.

Top

Error messages

***ESCAPE Messages**

JVAB278
No Java program associated with the file.

JVAB544
Unmonitored exception received.

Top

Analyze Java Virtual Machine (ANZJVM)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Analyze Java Virtual Machine (ANZJVM) command collects information about the Java Virtual Machine (JVM) for a specified job. A set of JVM information is collected immediately when the command is run. This collected JVM data is called a snapshot. A second snapshot is taken a specified amount of time later. By taking a snapshot of the JVM and comparing the data with a snapshot taken at a later time, the data can be analyzed to help find object leaks. The information is dumped using printer file QSYSPRT. The user data for the QSYSPRT file is 'ANZJVM'. The dump includes formatted information about the JVM heap. Details include names of classes, number of active objects per class, and the class loader used to load each class.

Restrictions:

- This command uses the Start Service Job (STRSRVJOB) and Start Debug (STRDBG) commands. The user of this command must be authorized to those commands.
- This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.
- This command must be run under a user profile that is the same as the job user identity of the JVM job, or that has use (*USE) authority to the job user identity of the JVM job.
- This command is not allowed if the remote service operation has been started for another job and that job is not the same job specified on this command.
- This command is not allowed if the JVM job is held, suspended, or ending.

Top

Parameters

Keyword	Description	Choices	Notes
JOB	Job name	*Qualified job name*	Optional, Positional 1
	Qualifier 1: Job name	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000000-999999	
INTERVAL	Time interval	0-3600, **60**	Optional
FRCGC	Force garbage collection	***YES**, *NO	Optional
SORT	Sort by	***NUMOBJCHG**, *NUMOBJ, *SIZECHG, *SIZE, *NAME	Optional
DUPJOBOPT	Duplicate job option	***SELECT**, *MSG	Optional

Top

Job name (JOB)

Specifies the name of the job where the JVM is running. If no job number is given, all of the jobs currently in the system are searched for the simple job name. The job name entered must be a job in which a JVM is currently running.

***SRVJOB**
Information about the JVM in the job currently being serviced will be dumped. If no job is currently being serviced, then a job identifier is required.

job-name
Specify the name of the JVM job.

user-name
Specify the name of the user of the JVM job.

job-number
Specify the number of the JVM job.

Top

Time interval (INTERVAL)

Specifies the time interval in seconds between the snapshots of the JVM to be analyzed.

60 Sixty seconds will pass between JVM snapshots of data.

0-3600 Specify the maximum number of seconds that will pass between JVM snapshots of data.

Top

Force garbage collection (FRCGC)

Specifies if a garbage collection cycle should be forced to take place.

***YES** A garbage collection cycle will take place before each snapshot of data is collected.

***NO** No garbage collection cycle will be forced to take place while collecting the data.

Top

Sort by (SORT)

Specifies the order in which the information is sorted.

***NUMOBJCHG**
Information is sorted by the number of objects changed.

***NUMOBJ**
Information is sorted by the number of objects in the first snapshot.

***SIZECHG**
Information is sorted by the change in the amount of space used by the object.

***SIZE** Information is sorted by the amount of space used by the object.

***NAME**
Information is sorted by the class name.

Top

Duplicate job option (DUPJOBOPT)

Specifies the action taken when duplicate jobs are found by this command.

***SELECT**
The selection display is shown when duplicate jobs are found during an interactive session. Otherwise, an escape message is issued.

***MSG** An escape message is issued when duplicate jobs are found.

Top

Examples

```
ANZJVM   JOB(099246/FRED/QJVACMDSRV)  INTERVAL(60)
```

This command will collect two snapshots of the Java Virtual Machine (JVM), 60 seconds apart, for the job with job name QJVACMDSRV, user name FRED, and job number 099246. The analyzed data from the snapshots is written to a spooled file. The spooled file name will be QSYSPRT and the spooled file user data text will be ANZJVM.

Top

Error messages

***ESCAPE Messages**

JVAB602
Job parameter required.

JVAB603
Unable to open print file.

JVAB605
ANZJVM failed with reason code &1.

JVAB60A
Job not found.

CPF1938
Command is not allowed while serviced job is not active.

CPF3524
More than one job with specified name found.

CPF3536
Job completed and cannot be serviced.

CPF3938
Already servicing another job.

CPF9824
Not authorized to command &1 in library &2.

Top

Analyze Object Conversion (ANZOBJCVN)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Analyze Object Conversion (ANZOBJCVN) command collects or reports information for user-created objects on the system. It can be used to plan the conversion of user objects to a new release of i5/OS operating system.

The following types of objects in libraries are analyzed and reported by this command:

- Programs (original program model (OPM) and Integrated Language Environment (ILE))
- Service programs
- Modules
- SQL packages

The following types of objects in directories are analyzed and reported by this command:

- Stream files with attached Java programs that were optimized to run in the i5/OS operating system. Only stream files in the ″root″ (/), QOpenSys and user-defined file systems are analyzed.
- Names of integrated file system objects in the ″root″ (/) file system as well as user-defined file systems created with option CASE(*MONO).

In addition, this command will determine the total number of spooled files on each auxiliary storage pool (ASP) that need to be converted and the estimated time to perform this conversion. This collection can be turned on or off with **Check spooled files (SPLFILE)** parameter.

The following information is collected or reported for each object:

- Object name
- Object type
- System level (the release of the operating system used when this user object was created).
- Creation data (indicates if an object has all the required data to be converted to the new release of the operating system).
- Digitally signed (indicates if an object has a digital signature or not. Program object conversion will lose any digital signatures associated with the object).
- Profiling (indicates if an object has profiling data or not. If an object has profiling data and its encapsulated profiling data was removed, this program will lose its profiling data during program object conversion).
- State (shows the current state of a user object. Objects with a current state of *SYSTEM or *INHERIT will change to *USER during program object conversion).
- Conversion time estimate (shows the estimated conversion time for the object. The estimates are calculated in seconds. If the object cannot be converted, the estimated time is 0).

  Note: All the references to conversion time in the Object Summary and Object Detail reports do not include time for case mapping conversions because it is done automatically by the operating system.
- The object name contains at least one character which is affected by new Unicode characters and casing rules in the target release. This object could potentially be renamed in the target release when the automatic conversion of the directories is performed for the new Unicode characters and casing rules.

Note: In most cases, the objects analyzed by ANZOBJCVN can be converted by Start Object Conversion (STROBJCVN) command or on first touch of the system after a software upgrade. However, STROBJCVN

will not convert the directories to support the new Unicode characters and casing rules since this conversion is automatically done by the operating system. Also STROBJCVN will not convert spooled database members because this conversion is automatically done based on the existence of the QSPL/QSPLDFRCVN data area. The default is to convert spooled files during IPL for all the spooled files stored in the system auxiliary storage pool (ASP) and all basic user ASPs or during the vary on of an independent ASP for spooled files stored in a primary or secondary ASP if the QSPL/QSPLDFRCVN data area does not exist after ASP.

Restrictions:

- You must have all object (*ALLOBJ) special authority to run this command.
- This command, especially the *COLLECT stage, can be very long running. For this reason, it is suggested that this command be run in a batch job.
- Any primary or secondary independent ASP (ASPs 33-255) referenced by this command must be varied on and have a status of 'Available' before running this command.
- Any user-defined file systems referenced by this command must be mounted before running this command.
- This command only can be run in two ways:
  - Having only one active job with ANZOBJCVN working with libraries and directories at the same time.
  - Having two active jobs running ANZOBJCVN at the same time: one job running ANZOBJCVN only for libraries and the other running ANZOBJCVN only for directories.
- If the QUSRSYS library is deleted after information is collected using ANZOBJCVN then the information will be lost.

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	Option	*COLLECT, *REPORT	Required, Positional 1
RPTTYPE	Type of report	Single values: ***ALL** Other values (up to 4 repetitions): *LIBSUM, *LIBDTL, *OBJSUM, *OBJDTL	Optional
LIB	Libraries to analyze	*Generic name, name*, ***ALLUSR**, *NONE	Optional
ASPDEV	ASP device	*Name*, *****, *ALLAVL, *CURASPGRP, *SYSBAS	Optional
SPLFILE	Check spooled files	***YES**, *NO	Optional
OBJ	Object	*Path name*, ***NONE**	Optional
SUBTREE	Directory subtree	***ALL**, *NONE	Optional
TGTRLS	Target release	*Character value*, **V6R1M0**	Optional

Top

Option (OPTION)

Specifies whether you want to collect information or generate reports using the collected information.

This is a required parameter.

***COLLECT**
Information for objects in the specified libraries and directories is collected for analysis. If Check Spooled File (SPLFILE) is set to *YES, spool database members info will be collected when collecting libraries.

***REPORT**
Collected object information is analyzed and reports are generated for objects in the specified libraries and directories as well as spool files.

Top

Type of report (RPTTYPE)

Specifies the types of object reports to generate. The reports are written to the printer file QPIZARPT.

If *COLLECT is specified for the Option (OPTION) parameter, specification of this parameter is ignored.

Single values

***ALL** All summarized and detail object reports are generated for all objects in the libraries and directories specified in the previous collection. If Check Spooled File (SPLFILE) was set to *YES during the previous collection, spool database members information will be generated in the Library Summary report.

Other values (up to 4 repetitions)

***LIBSUM**
A summary report is generated for each library specified in the previous collection. If SPLFILE(*YES) was specified, spool database members information will be generated in this report.

***LIBDTL**
A detailed report is generated for each library specified in the previous collection.

***OBJSUM**
A summary report is generated for integrated file system objects specified in the previous collection.

***OBJDTL**
A detailed report is generated for integrated file system objects specified in the previous collection.

Top

Libraries to analyze (LIB)

Specifies the libraries to be analyzed.

If *REPORT is specified for the Option (OPTION) parameter, specification of this parameter is ignored.

***ALLUSR**
All user libraries are analyzed. All libraries with names that do not begin with the letter Q are analyzed except for the following:

```
#CGULIB     #DSULIB     #SEULIB
#COBLIB     #RPGLIB
#DFULIB     #SDALIB
```

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered user libraries and are also analyzed:

```
QDSNX       QRCLxxxxx   QUSRDIRDB   QUSRVI
QGPL        QSRVAGT     QUSRIJS     QUSRVxRxMx
QGPL38      QSYS2       QUSRINFSKR
QMGTC       QSYS2xxxxx  QUSRNOTES
QMGTC2      QS36F       QUSROND
QMPGDATA    QUSER38     QUSRPOSGS
QMQMDATA    QUSRADSM    QUSRPOSSA
QMQMPROC    QUSRBRM     QUSRPYMSVR
QPFRDATA    QUSRDIRCF   QUSRRDARS
QRCL        QUSRDIRCL   QUSRSYS
```

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

***NONE**
No user library is analyzed. You can specify LIB(*NONE) if you only want to analyze objects that are located in directories.

generic-name
Specify the generic name of the libraries to be analyzed. A generic name is a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, all libraries that have names with the same prefix as the generic name are analyzed.

name Specifies the name of the user library whose objects are to be analyzed.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device where storage is allocated for the library and spooled files to be analyzed. If the library is in an ASP that is not part of the thread's library name space, this parameter must be specified to ensure the correct library is analyzed.

Note: This parameter does not apply for the objects specified in the OBJ parameter since the independent ASP name is part of the path name of the object.

If *REPORT is specified for the Option (OPTION) parameter, specification of this parameter is ignored.

*****
The ASPs that are currently part of the thread's library name space will be searched to find the library. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and, if the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group.

***ALLAVL**
All available ASPs will be searched. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and all available primary and secondary ASPs (ASPs 33-255 with a status of 'Available').

***CURASPGRP**
If the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group will be searched to find the library. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched. If no ASP group is associated with the thread an error will be issued.

***SYSBAS**
The system ASP (ASP 1) and all defined basic user ASPs (ASPs 2-32) will be searched to find the library. No primary or secondary ASPs will be searched, even if the thread has an ASP group.

name

Specify the name of the primary or secondary ASP device to be searched. The primary or secondary ASP must have been activated (by varying on the ASP device) and have a status of 'Available'. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched.

Top

Check spooled files (SPLFILE)

Specifies whether to analyze the total number of spooled files on each auxiliary storage pool (ASP) that need to be converted and the estimated time to perform this conversion.

Additionally, one cannot specify LIB(*NONE) if SPLFILE(*YES) is specified.

***YES** Spool files on each auxiliary storage pool (ASP) are analyzed.

***NO** Spool files on each auxiliary storage pool (ASP) are not analyzed.

Note: The ASP device selected (ASPDEV) affects the collection of spooled files.

Top

Object (OBJ)

Specifies the path name of the object, or a pattern to match the name, of the objects to be analyzed.

If *REPORT is specified for the Option (OPTION) parameter, specification of this parameter is ignored.

The object path name can be either a simple name or a name that is qualified with the name of the directory in which the object is located. A pattern can be specified in the last part of the path name. An asterisk (*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/.

Note: If the last component in the path name is a symbolic link, the symbolic link object is analyzed, but the object pointed to by the symbolic link is not analyzed.

***NONE**
No object is analyzed.

path-name
Specify the path name of the object to be analyzed.

Top

Directory subtree (SUBTREE)

Specifies whether or not to analyze the objects within the subtree if the object specified by the **Object (OBJ)** parameter is a directory.

If *REPORT is specified for the Option (OPTION) parameter, specification of this parameter is ignored.

***ALL** The objects specified by OBJ will be analyzed, if appropriate. If the object is a directory, its contents as well as the contents of all of its subdirectories will be analyzed.

Note: Pattern matching from the OBJ parameter only applies to the first level objects. If the first level object is a directory, the pattern matching does not apply to its contents or the contents of its subdirectories.

***NONE**
The objects specified by OBJ will be analyzed, if appropriate. If the object is a directory, it is analyzed if appropriate, but its contents are not analyzed.

Once the command has begun processing a specific directory subtree, the objects which will be found and processed may be affected by operations that update the organization of objects within the specified directory tree. This includes, but is not limited to, the following:

- Adding, removing, or renaming object links
- Mounting or unmounting file systems
- Updating the effective root directory for the process calling the command
- Updating the contents of a symbolic link

In order to process the directory subtree, the system code may increase the process-scoped maximum number of file descriptors that can be opened during processing. This is done so that the command is not likely to fail due to a lack of descriptors. This process-scoped maximum value is not reset when the command completes.

Top

Target release (TGTRLS)

Specifies the release to which the data and programs will be migrated. This information is used to determine the specific analysis to be performed. This is dependent on the release where the command is being run.

If *REPORT is specified for the Option (OPTION) parameter, specification of this parameter is ignored.

VxRyMz
Specifies the release to which data and programs will be migrated.

Top

Examples

Example 1: Collect Library Information

```
ANZOBJCVN   OPTION(*COLLECT)
```

This command will collect object conversion information for all objects in all user libraries in the system auxiliary storage pool (ASP), all configured basic user ASPs, and all independent ASPs that are varied on

and have a status of 'Available'. If the library is in an ASP that is not part of the thread's library name space, parameter ASPDEV must be specified to ensure the desired library is analyzed. No information is collected for directories.

Example 2: Generate Library Information Reports

```
ANZOBJCVN   OPTION(*REPORT) RPTTYPE(*LIBSUM *LIBDTL)
```

This command will report previously collected library information. The library summary and library detail reports will be generated.

Example 3: Collect Object Information from Directories

```
ANZOBJCVN   OPTION(*COLLECT)  LIB(*NONE)
            OBJ('/MYDIR/MYDOCS')  SUBTREE(*ALL)
```

This command will collect object conversion information for the /MYDIR/MYDOCS directory, its contents as well as the contents of all of its subdirectories will be analyzed. No information is collected for objects in libraries.

Example 4: Generate Object Detailed Report

```
ANZOBJCVN   OPTION(*REPORT)  RPTTYPE(*OBJDTL)
```

This command will report previously collected object information for directories. The object details report will be generated.

Example 5: Collect Library Information and Directory Information

```
ANZOBJCVN OPTION(*COLLECT) LIB(MYLIB) SPLFILE(*NO)
          OBJ('/MYDIR') SUBTREE(*ALL)
```

This command will collect object conversion information for the MYLIB library and for /MYDIR directory, its contents as well as the contents of all of its subdirectories will be analyzed. No spool database members information will be collected.

Top

Error messages

***ESCAPE Messages**

CPF218C
&1 not a primary or secondary ASP.

CPF222E
&1 special authority is required.

CPF9810
Library &1 not found.

CPF9833
*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group.

CPFB0DC
ANZOBJCVN OPTION (*REPORT) was not successful.

CPFB0DE
Not all eligible objects were analyzed.

CPFB0DF
ANZOBJCVN found locked objects in library QUSRSYS.

Top

Analyze Problem (ANZPRB)

Where allowed to run: Interactive environments (*INTERACT *IPGM *IREXX *EXEC)
Threadsafe: No

Parameters
Examples
Error messages

The Analyze Problem (ANZPRB) command allows you to analyze, create problem records for, or report problems that are not detected by the system. For example, you can analyze or report:

- Job or programming problems.
- Equipment or communications problems.
- Problems that made it necessary to do an initial program load (IPL) of the system again.
- Problems on a device or system not attached to your system.

Restriction: This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.

Top

Parameters

Keyword	Description	Choices	Notes
ANZTYPE	Analysis type	*REMOTE, *LOCAL, ***MENU**	Optional, Positional 1
RCPNAME	Remote control point	*Name*	Optional
NETID	Network ID	*Name*, ***NETATR**	Optional
USERID	User ID	*Name*	Optional
PASSWORD	Password	*Character value*, **X''**, *NONE	Optional

Top

Analysis type (ANZTYPE)

Specifies the type of analysis. The default value, *MENU allows you to do local or remote analysis on a System i5 or to analyze problems on another system which is not a System i5. If you want to analyze problems on a local or remote System i5, you can bypass the initial menu by specifying *LOCAL or *REMOTE.

Note: You cannot do remote analysis if IBM System Manager for i5/OS is not installed.

This is a required parameter.

The possible values are:

***MENU**
The Select Type of System menu is shown. You may choose to analyze problems on:

- This System i5
- Another System i5 that is enrolled as a service requester
- Another System i5that is not enrolled as a service requester
- Another system that is not a System i5

***LOCAL**
Local analysis is selected. Problem analysis is done on this System i5.

***REMOTE**
Remote analysis is selected. Problem analysis is done for another System i5 that is enrolled as a service requester.

Top

Remote control point (RCPNAME)

Specifies the remote control point name for the service requester system where the remote analysis is done.

Note: This parameter is valid only if *REMOTE is specified for the **Analysis type** prompt (ANZTYPE parameter).

Top

Network ID (NETID)

Specifies the network identifier (ID) for the service requester system where the remote analysis is done.

Note: This parameter is valid only if *REMOTE is specified for the **Analysis type** prompt (ANZTYPE parameter).

The possible values are:

***NETATR**
The network ID of the service provider is used.

network-ID
Specify the network ID.

Top

User ID (USERID)

Specifies the user identifier (ID) used to access the remote system.

Note: This parameter is valid only if *REMOTE is specified for the **Analysis type** prompt (ANZTYPE parameter).

Top

Password (PASSWORD)

Specifies the password used to access the remote system.

Note: This parameter is valid only if *REMOTE is specified for the **Analysis type** prompt (ANZTYPE parameter).

The possible values are:

***NONE**
No password is needed to access the remote system because the remote system has a security level of 10.

password
Specify the password.

Top

Examples

Example 1: Displaying the Menu

```
ANZPRB
```

This command shows the Analyze Problem menu.

Example 2: Starting Remote Analysis

```
ANZPRB   ANZTYPE(*REMOTE)
```

This command shows the display which prompts for the remaining values of the command. After you specify the appropriate values, remote analysis begins.

Example 3: Accessing Remote System with User ID and Password

```
ANZPRB   ANZTYPE(*REMOTE)  RCPNAME(RCH38377)  USERID(JON)
         PASSWORD(MY1STPWD)
```

This command shows the display which prompts for the remaining values of the command. After you specify the appropriate values beyond the ones specified on the command example, remote analysis begins.

Example 4: Remote Analysis has Security Level of 10

```
ANZPRB   ANZTYPE(*REMOTE)  RCPNAME(RCH38377)  USERID(JON)
```

This command is slightly different than the preceding example. The same display prompt appears; however, if PASSWORD is not specified, the system assumes that the remote system has a security level of 10, that is, it does not use passwords. After you specify the appropriate values beyond the ones specified on the command example, remote analysis begins.

Example 5: Displaying the Menu

```
ANZPRB   ANZTYPE(*MENU)
```

This command shows a menu prompting you for the type of analysis to be done. The remaining parameters do not appear on the display.

Example 6: Starting Local Analysis

ANZPRB ANZTYPE(*LOCAL)

This command begins analysis on the local device. The remaining parameters do not appear on the display.

Top

Error messages

***ESCAPE Messages**

CPF2B01
Problem analysis cannot continue because of an error in the program.

CPF2B3C
Licensed program &1 not installed.

CPF9308
Unable to complete problem analysis. Reason code &1.

Top

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This i5/OS commands publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing
AFP
AS/400
CICS
COBOL/400
C/400
DataPropagator
DB2
Distributed Relational Database Architecture
Domino
DRDA
IBM
Infoprint
InfoWindow
i5/OS
iSeries
Integrated Language Environment
Lotus
LPDA
OfficeVision
Print Services Facility
RPG/400
System i
System x
SystemView
System/36
TCS
Tivoli
WebSphere
z/OS

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Code license and disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Printed in USA