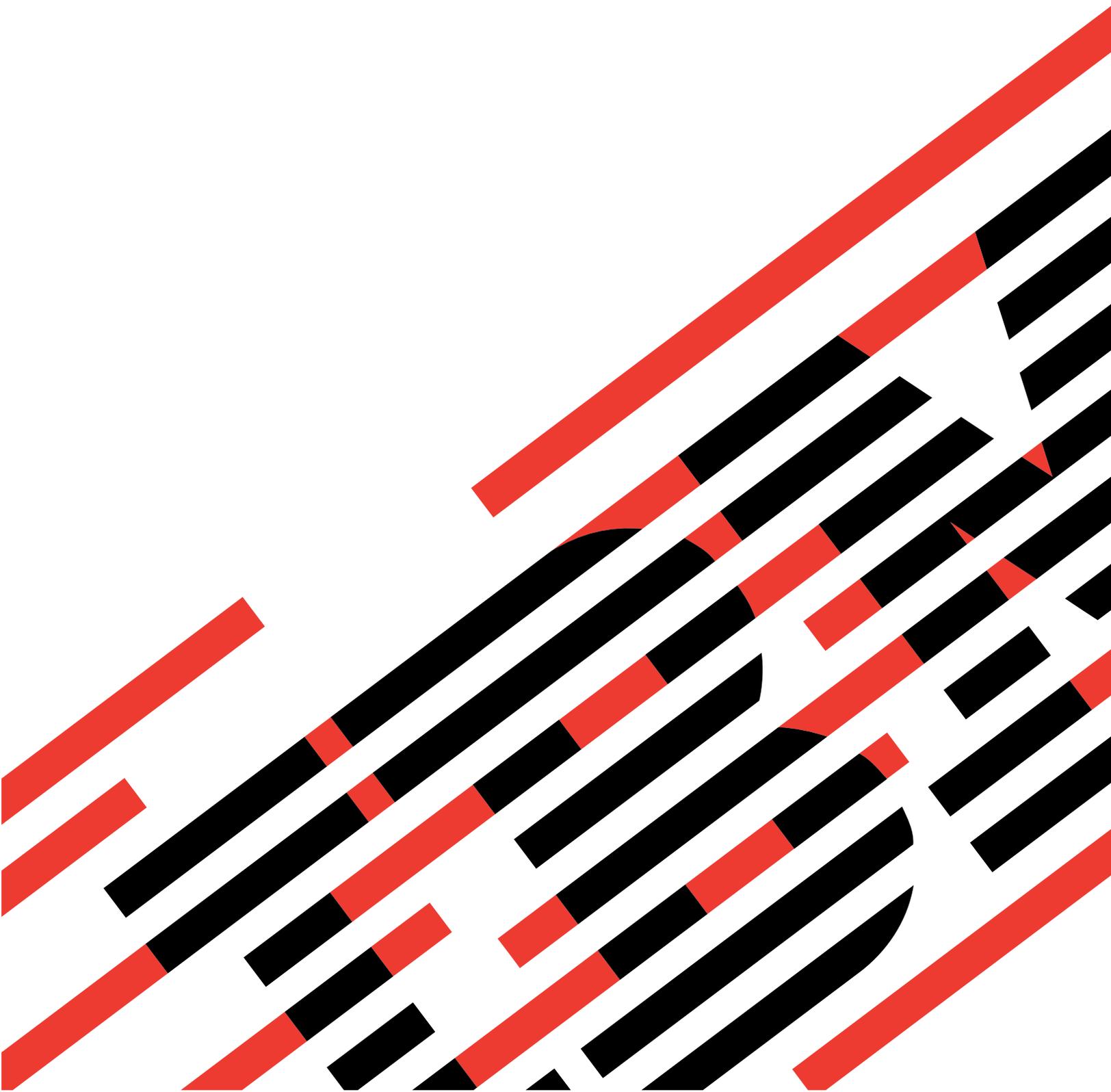


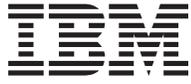
IBM

@server

iSeries

CL Commands Volume 20





@server

iSeries

CL Commands Volume 20

Contents

Command Descriptions	1
STRCRG (Start Cluster Resource Group) Command Description	1
STRCHTSVR (Start Clustered Hash Table Server) Command Description	3
STRCMTCTL (Start Commitment Control) Command Description	5
STRCMNSVR (Start Communications Server) Command Description	9
STRCMNTRC (Start Communications Trace) Command Description	9
STRCPYSCN (Start Copy Screen) Command Description	13
STRDBMON (Start Database Monitor) Command Description	16
STRDBRDR (Start Database Reader) Command Description	18
STRDBG (Start Debug) Command Description	21
STRDBGSVR (Start Debug Server) Command Description	27
STRDIRSHD (Start Directory Shadowing) Command Description	27
STRDSKRGZ (Start Disk Reorganization) Command Description	28
STRDKTRDR (Start Diskette Reader) Command Description	29
STRDKTWTR (Start Diskette Writer) Command Description	32
STREDU (Start Education) Command Description	35
STREXPBRM (Start Media Expiration Processing for BRM) Command Description	36
STRFMA (Start Font Management Aid) Command Description	36
STRGRPJS (Start Group Using Job Scheduler) Command Description	37
STRHOSTSVR (Start Host Server) Command Description	38
STRHTPCRL (Start HTTP Crawl) Command Description	41
STRINFSKR (Start InfoSeeker) Command Description	43
STRIDD (Start Interactive Data Definition Utility) Command Description	44
STRITF (Start Interactive Terminal Facility) Command Description	45
STRIPSIFC (Start IP over SNA interface) Command Description	45
STRJS (Start Job Scheduler) Command Description	46
STRJOBTRC (Start Job Trace) Command Description	46
STRJRN (Start Journal) Command Description	47
STRJRNP (Start Journal Access Path) Command Description	52
STRJRNOBJ (Start Journal Object) Command Description	55
STRJRNP (Start Journal Physical File) Command Description	58
STRMSF (Start Mail Server Framework) Command Description	61
STRMNTBRM (Start Maintenance for BRM) Command Description	62
STRMGDSYS (Start Managed System Services) Command Description	63
STRMGRSRV (Start Manager Services) Command Description	64
STRMGRBRM (Start Migration Using BRM) Command Description	65
STRMOD (Start Mode) Command Description	66
STRNFSSVR (Start Network File System Server) Command Description	67
STRNWCSL (Start Network Server Console) Command Description	70
STROBJCVN (Start Object Conversion) Command Description	71
STRPASTHR (Start Pass-Through) Command Description	73
STRPEX (Start Performance Explorer) Command Description	79
STRPFRG (Start Performance Graphics) Command Description	80
STRPFRT (Start Performance Tools) Command Description	82
STRPFRT (Start Performance Trace) Command Description	83
STRTCPPTP (Start Point-to-Point TCP/IP) Command Description	85
STRPJ (Start Prestart Jobs) Command Description	88
STRPRTWTR (Start Printer Writer) Command Description	89
STRPGMPRF (Start Program Profiling) Command Description	96
STRPGMMNU (Start Programmer Menu) Command Description	97
STRQSH (Start QSH) Command Description	101
STRQRY (Start Query) Command Description	103
STRQMPCR (Start Query Management Procedure) Command Description	103

STRQMQRV (Start Query Management Query) Command Description	106
STRQST (Start Question and Answer) Command Description	110
STRRCYBRM (Start Recovery Using BRM) Command Description	111
STRRMTSPT (Start Remote Support) Command Description	112
STRRMTWTR (Start Remote Writer) Command Description	118
STRREXPRC (Start REXX Procedure) Command Description	121
RPCBIND (Start RPC Binder Daemon) Command Description	123
STRSCHIDX (Start Search Index) Command Description	124
STRSRVJOB (Start Service Job) Command Description	125
STRSBS (Start Subsystem) Command Description	127
STRSBSBRM (Start Subsystems Using BRM) Command Description	129
STRSPTN (Start Support Network) Command Description	130
STRSYSMGR (Start System Manager) Command Description	130
STRSST (Start System Service Tools) Command Description	131
STRS36 (Start System/36) Command Description	132
STRS36PRC (Start System/36 Procedure) Command Description	133
STRTCP (Start TCP/IP) Command Description	134
STRTCPIFC (Start TCP/IP Interface) Command Description	137
STRTCP SVR (Start TCP/IP Server) Command Description	139
STRTIESSN (Start Technical Information Exchange Session) Command Description	146
STRTRC (Start Trace) Command Description	147
STRTRPMGR (Start Trap Manager) Command Description	151
STRUSF (Start Ultimeia System Facilities) Command Description	152

Command Descriptions



STRCRG (Start Cluster Resource Group) Command Description

STRCRG Command syntax diagram

Purpose

The Start Cluster Resource Group (STRCRG) command enables resiliency for the specified cluster resource group. The Start Cluster Resource Group command can be used to change cluster resource group from a status of Indoubt or Inactive to Active. Since some changes to the cluster resource group can only be performed when the cluster resource group is not active, ensure that the cluster resource group definition is correct before calling this command.

This command will do the following for all cluster resource group types:

- Set the cluster resource group status to Start Cluster Resource Group Pending.
- If the current recovery domain has more than one backup node and some backup nodes are not active, the recovery domain may be reordered so that all active backup nodes are ordered before inactive backup nodes. If the inactive backup nodes are already ordered after active backup nodes or if there are no inactive backup nodes, this command will not change the recovery domain.
- Call the exit program on all active nodes in the recovery domain with an action code of Start, if an exit program is specified for the cluster resource group.
- Set the cluster resource group status to Active if the exit program is successful on all active nodes in the recovery domain.
- If the cluster resource group exit program is unsuccessful on any active node in the recovery domain, the status of the cluster resource group will be set to Indoubt on all nodes.

This command will do the following for application cluster resource groups:

- Verify the takeover IP address has been configured on all nodes in the recovery domain except replicates.
- Start the TCP/IP interface on the primary node.
- If the exit program is unsuccessful on any active node in the recovery domain, the long running application exit program job on the primary node will be cancelled and the Takeover IP address will be ended on the primary node.

For an application cluster resource group, the exit program on the primary node is not expected to complete. The status of the cluster resource group will be set to Active when the exit program job has been started on the primary and the exit program has completed successfully on all other nodes in the recovery domain.

This CL command requires the following for resilient device cluster resource groups:

1. The cluster resource group must have at least one configuration object entry.
2. The configuration objects specified for the cluster resource group must exist on all active nodes in the recovery domain and the resource name specified in a configuration object must be the same on all active nodes in the recovery domain.
3. If a data base name has been specified for a configuration object, it must be the same on all active nodes in the recovery domain.
4. If a server takeover IP address is specified, it must exist on all nodes in the recovery domain.

5. The primary node must be the current owner of all IOPs or high-speed link I/O bridges for the devices in the cluster resource group.
6. The IOP or high-speed link I/O bridge controlling the devices specified in the cluster resource group must be accessible by all active nodes in the cluster resource group's recovery domain.
7. Starting the cluster resource group will not vary on the configuration objects or start the server takeover IP address.
8. All members of an auxiliary storage pool group must be configured in the cluster resource group.
9. A value of *PRIMARY for the device's configuration object online' attribute can be specified only for a secondary auxiliary storage pool.

Restrictions

1. To use this command you must have *IOSYSCFG authority.
2. This command cannot be called from a cluster resource group exit program.
3. Cluster Resource Services must be active on the node running the command.
4. The status of the node currently assigned the role of primary must be Active.
5. The cluster resource group status must be either Inactive or Indoubt.
6. For a device cluster resource group:
 - a. The cluster resource group must have at least one configuration object entry.
 - b. The configuration objects specified for the cluster resource group must exist on all active nodes in the recovery domain and the resource name specified in a configuration object must be the same on all active nodes in the recovery domain.
 - c. The primary node must be the current owner of all input/output processors or high-speed link I/O bridges for the devices in the cluster resource group.
 - d. Hardware configuration must be complete so that the physical hardware has been associated with the configuration object. In particular for auxiliary storage pools, the disk units must have been assigned.
 - e. The input/output processor or high-speed link I/O bridge controlling the devices specified in the cluster resource group must be accessible by all active nodes in the cluster resource group's recovery domain.

Required Parameters

CLUSTER

Specifies the name of the cluster containing the cluster resource group.

cluster-name: Specify the name of the cluster.

CRG Specifies the name of the cluster resource group which will be started.

cluster-resource-group-name: Specify the name of the cluster resource group to start.

Optional Parameter

EXITPGMDTA

Specifies up to 256 bytes of data that is passed to the cluster resource group exit program when it is called. This parameter may contain any scalar data except pointers. For example, it can be used to provide state information. This data will be stored with the specified cluster resource group and copied to all nodes in the recovery domain. Pointers in this area will not resolve correctly on all nodes and should not be placed in the data. The data specified will replace the existing exit program data stored with the cluster resource group. If blanks are specified, then the exit program data stored with the cluster resource group will be cleared. This parameter must be set to *SAME if no exit program is specified for the cluster resource group.

***SAME**: The exit program data stored with the cluster resource group specified will be passed to the exit program.

exit-program-data: Specify the data to be passed to the exit program.

Example for STRCRG

```
STRCRG CLUSTER(MYCLUSTER) CRG(MYCRG) EXITPGMDTA('important information')
```

This command enables resiliency of the cluster resource group MYCRG in the cluster MYCLUSTER. The exit program, when called, is passed 'important information' in the exit program data field. The cluster resource group will be ACTIVE as a result of this command.

Error messages for STRCRG

*ESCAPE Messages

CPF0001

Error found on &1 command.



STRCHTSVR (Start Clustered Hash Table Server) Command Description

STRCHTSVR Command syntax diagram

Purpose >>

The Start Clustered Hash Table Server (STRCHTSVR) command is used to start a clustered hash table server on each cluster node specified in the NODE parameter. Successful completion of this command results in a job started on each node defined in the cluster. Each node specified in the NODE parameter will be in the clustered hash table server domain.

The clustered hash table server enables sharing and replicating data between cluster nodes using the Clustered Hash Table APIs. The data is stored within the clustered hash table server in non-persistent storage.

You can use the End Clustered Hash Table Server (ENDCHTSVR) command to end the clustered hash table server.

Restrictions

- Cluster Resource Service must be active on the local node.
- All nodes specified in the NODE parameter must have Cluster Resource Services active.
- Requesting user profile must exist on all nodes specified in the NODE parameter.
- Requesting user must have *CHANGE authority to the authorization list, if specified, on the current node as well as all nodes specified in the NODE parameter.

Required Parameter

SERVER

Specifies the clustered hash table server to be started.

server-name: Specify the name of the clustered hash table server to be started.

Optional Parameters

RSPTIMO

Specifies the amount of time (in seconds) that the clustered hash table server has to complete a

retrieve request. An error will be returned to the requestor of the retrieve if the clustered hash table server does not get a response before the specified length of time. This parameter is ignored when the current cluster version is 3 or greater.

60: The clustered hash table server will wait 60 seconds for a response.

response-timeout-value: Specify the amount of time (in seconds) the clustered hash table server waits for a response before returning an error. Valid values range from 1 through 300 seconds.

AUTL Specifies an authorization list name for the clustered hash table server. This defines the list of users authorized to start, end and connect to the clustered hash table server. An authorization list by the same name must exist on all nodes in the NODE parameter. The same authorization list name must be specified when starting a clustered hash table server that is already running on other nodes in the cluster.

***SERVER:** Use the same authorization list as the specified clustered hash table server if it is already active on one of the nodes in the cluster. If the server does not exist in the cluster and this value is specified then no special authority will be needed to access, start, end, or connect to the server. This value must be specified if the current cluster version is 2 or less .

authorization-list-name: The name specified defines the list of users authorized to the clustered hash table server. Users must have *CHANGE authority to the named authorization list to start and end the clustered hash table server. Users must have *USE authority to the named authorization list to connect to the clustered hash table server. Management of the authorization list is the users responsibility.

NODE Specifies the list of cluster nodes that compose the clustered hash table domain. A server job will be started on each of the cluster nodes specified. If the clustered hash table server already exists in the cluster, the cluster nodes specified will be added to the clustered hash table domain and a job will be started on the node specified. Nodes in this list must be unique. The nodes must be active in the cluster.

***LOCAL:** A clustered hash table server will be started on the local node only. *LOCAL can be specified only once.

node-identifier: Specify the name of each cluster node that defines the clustered hash table domain. Up to 128 cluster nodes can be specified. <<

Examples for STRCHTSVR

Example 1: Starting a local clustered hash table server >>

```
STRCHTSVR SERVER(CHTSVR1)
```

This command starts clustered hash table server CHTSVR1 only on the local node. There is no authority restriction on the clustered hash table server.

Example 2: Starting a clustered hash table server on multiple nodes

```
STRCHTSVR SERVER(CHTSVR2) AUTL(AUTHLIST) NODE(FRED BARNEY)
```

This command starts clustered hash table server CHTSVR2 on nodes FRED and BARNEY. The clustered hash table server has access restricted by AUTHLIST. <<

Error messages for STRCHTSVR

*ESCAPE Messages

CPFBD02

Start Clustered Hash Table Server failed with reason code &1.

STRCMTCTL (Start Commitment Control) Command Description

STRCMTCTL Command syntax diagram

Purpose

The Start Commitment Control (STRCMTCTL) command is used to establish either a job level or activation group level commitment definition.

This command also specifies the level of record locking that occurs for the commitment definition to be started. Also, a notify object can be specified.

A default journal can be specified. Entries that describe all journals and systems involved in a commitment control operation can be placed in this journal.

More information on the use of journal management is in Journal management article in the Information Center or the use of commitment control is in the Commitment control article in the Information Center.

Restrictions:

1. The user must have object operational and add authority to the object named on the NFYOBJ parameter, if an object is specified.
2. The user must have object operational and add authority to the object named on the DFTJRN parameter, if an object is specified.

Required Parameter

LCKLVL

Specifies the default level of record locking that occurs for the commitment definition to be started.

***CHG:** Every record read for update (for a file opened under commitment control) is locked. If a record is updated, added, or deleted, that record remains locked until the transaction is committed or rolled back. Records that are accessed for update but are released without being updated are unlocked.

***CS:** Every record accessed for files opened under commitment control is locked. Records that are not updated or deleted are locked only until a different record is accessed. Records that are updated, added, or deleted are locked until the transaction is committed or rolled back.

***ALL:** Every record accessed for files opened under commitment control is locked until the transaction is committed or rolled back.

Optional Parameters

NFYOBJ

Specifies the name and type of the object where notification is sent regarding the status of a transaction for a commitment definition. The commitment identifier of the last successful commit operation is sent to the notify object only for the following conditions:

- For a job level commitment definition, if any of the following are true:
 - A system failure occurs
 - The job ends with uncommitted changes
 - The job ends with a nonzero completion code
- For an activation group level commitment definition, if any of the following are true:
 - A system failure occurs

- The job ends with uncommitted changes
- The job ends with a nonzero completion code
- The activation group ends abnormally
- The activation group ends with uncommitted changes and the uncommitted changes are rolled back

For a system failure, the commitment identifier is placed in the notify object after the next successful initial program load (IPL). For a job that ends with uncommitted changes or with a nonzero completion code, the commitment identifier is placed in the notify object during end job processing. For an activation group that ends with uncommitted changes or ends abnormally, the commitment identifier is placed in the notify object during activation group end processing.

A commitment identifier (specified on the CMTID parameter on the Commit (COMMIT) command) can be specified on each commit operation performed for a commitment definition. If more than one job is concurrently using commitment control or there is more than one commitment definition being used concurrently within a single job, then each commitment definition for each job should use a unique notify object or the specified commit identifier should contain unique text such that the text identifies a single commitment definition for a single job. If COMMIT CMTID(*NONE) is specified, this parameter is ignored.

***NONE:** No notification is sent after a system failure or during job end.

Element 1: Name of Object to Receive Notification

The name of the object 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.

object-name: Specify the name of the object to receive notification of the last transaction that is successfully committed. The user must have the required authority for the object specified.

Element 2: Location of Text

***MSGQ:** The text identifying the last commitment boundary is placed on the specified message queue.

***DTAARA:** The text identifying the last commitment boundary is placed in the specified data area. The data area specified must be of type character, and unique to this job. The text is padded or truncated to fit the data area.

***FILE:** The text identifying the last commitment boundary is added to the specified physical file.

Element 3: Member to Receive Notification

This element is valid only when *FILE is specified.

***FIRST:** The first member of the physical file receives the notification.

member-name: Specify the name of the member of the physical file that receives the notification.

CMTSCOPE

Specifies the scope for the commitment definition to be started.

***ACTGRP**: An activation-group-level commitment definition is started for the activation group associated with the program issuing the command.

***JOB**: The job-level commitment definition is started for the job.

TEXT Specifies the text that briefly describes the commitment definition to be started. More information on this parameter is in Commonly used parameters.

***DFTTEXT**: The system is to provide a default text description for the commitment definition.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

DFTJRN

Specifies the default journal. The default journal contains entries identifying each of the resources involved in a unit of work. Entries can also be placed when each unit of work starts or ends due to a commit or rollback operation depending on the OMTJRNE parameter value. See the Backup, Recovery, and Availability topic in the Information Center for information on how the system performs the rollback operation under commitment control.

The default journal can be used when adding a resource through the Add Committable Resource (QTNADDCR) Application Program Interface (API). If the special value *DFTJRN is specified for the journal name when calling the API, the name specified on this DFTJRN parameter is used.

***NONE**: No default journal is specified.

The name of the default journal 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.

journal-name: Specify the name of the default journal.

OMTJRNE

Specifies the journal entries to omit from the default journal. If *NONE is specified on the DFTJRN parameter, this parameter is ignored.

***NONE**: No journal entries are omitted.

***LUWID**: The journal entry that contains the Logical Unit of Work Identifier (LUWID) and all the resources involved in the logical unit of work are omitted, if the logical unit of work is committed or rolled back successfully. If an error occurs while committing or rolling back the logical unit of work, the entry will always be sent regardless of this value.

Examples for STRCMTCTL

Example 1: Defining Activation Group Level Commitment Control

```
STRCMTCTL LCKLVL(*CHG)
CMTSCOPE(*ACTGRP)
TEXT('Blue Commit Group')
```

This command described by the user as the Blue Commit Group starts the activation group level commitment for the activation group associated with the program issuing the command.

Only records that are updated, inserted, or deleted are locked until the transaction is ended by a commit or rollback operation. No identification for the commitment boundary is sent after the initial program load (IPL) following an abnormal system end, after an abnormal end to an activation group for the job, or when the job or activation group ends either with uncommitted changes or with a nonzero completion code.

Example 2: Defining Job Level Commitment Control

```
STRCMTCTL LCKLVL(*ALL)
          NFYOBJ(RCVLIB/MYFILE *FILE IDSAVE)
          CMTSCOPE(*JOB)
          DFTJRN(MGWLIB/MYJRN)
```

This command starts the job level commitment definition. All records accessed in files opened under commitment control are locked until the commitment transaction is ended by a commit or rollback operation. If a commitment transaction ends in a manner that a notify object is to be updated with the commitment identifier of the last successful commit operation, the notify object to be updated is member IDSAVE of file MYFILE in the library RCVLIB. When a commit or rollback is done, an entry that lists information about all the resources involved in the logical unit of work is put into journal MYJRN in library MGWLIB.

Error messages for STRCMTCTL

***ESCAPE Messages**

CPF8351

Commitment control already active.

CPF8352

Attribute in notify object &1 type *&4 not valid.

CPF8360

Not enough storage for commitment control operation.

CPF8366

Commitment definition &2 not created.

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &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.

CPF9815

Member &5 file &2 in library &3 not found.

CPF9820

Not authorized to use library &1.

CPF9830

Cannot assign library &1.

STRCMNSVR (Start Communications Server) Command Description

STRCMNSVR Command syntax diagram

Purpose

The Start Communications Server (STRCMNSVR) command is used to start the iSeries 400 target display station pass-through server. The target display station pass-through server processes iSeries 400 display station pass-through, iSeries 400 Client Access work station function (WSF), and other 5250 emulation programs on programmable workstations.

Restriction: You must have job control (*JOBCTL) special authority to use this command.

Optional Parameters

NBRPASTHR

Specifies the number of target display station pass-through server jobs to be started.

***SYSVAL:** Specifies that the system value (QPASTHRSVR) is to be used to determine the number of servers.

number-pass-through-servers: Specify the number of servers. This will override the current system value (QPASTHRSVR). Valid values range from 1 to 100.

Example for STRCMNSVR

Example 1: Starting Target Display Station Pass-through Server

STRCMNSVR

This command starts the target display station pass-through server. System value QPASTHRSVR is used to determine how many server jobs are started.

No error messages.

STRCMNTRC (Start Communications Trace) Command Description

STRCMNTRC Command syntax diagram

Purpose

The Start Communications Trace (STRCMNTRC) command initiates a communications trace for a specified line, network interface, or network server description.

A communications trace continues until:

- The End Communications Trace (ENDCMNTRC) command is run
- The Communications Trace function of the Start System Service Tools (STRSST) command is used to end the trace
- A physical line problem causes the trace to end
- TRCFULL(*STOPTRC) is specified and the buffer becomes full

Restrictions:

1. The user must have *USE authority to the line, network interface, or network server to be traced.
2. To use this command you must have *SERVICE special authority, or be authorized to the Service Trace function of Operating System/400 through iSeries Navigator's Application Administration support.

The Change Function Usage Information (QSYCHFUI) API, 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. This command is shipped with public *EXCLUDE authority.
4. The following user profiles have authority to this command:
 - QSECOFR
 - QSRV

Required Parameters

CFGOBJ

Specifies the name of the configuration object to trace. The object is either a line description, a network interface description, or a network server description.

CFGTYPE

Specifies the type of configuration description to trace.

***LIN:** The type of configuration object is a line description.

***NWI:** The type of configuration object is a network interface description.

***NWS:** The type of configuration object is a network server description.

Optional Parameters

MAXSTG

Specifies the trace buffer size.

128K: A trace buffer of 128 kilobytes is used.

***MIN:** The minimum trace buffer size is used.

***MAX:** The maximum trace buffer size is used.

buffer-size: Specify the trace buffer size. Valid buffer sizes may be specified either as number of kilobytes, or as one of the following special values which has a one-letter suffix of 'K' for kilobytes, 'M' for megabytes, or 'G' for gigabytes: 128K, 256K, 2M, 4M, 6M, 8M, 16M, 32M, 64M, 128M, 256M, 512M, 1G. The minimum trace buffer size is 128 kilobytes.

DTADIR

Specifies the communication data to trace.

Note:

For network server description traces, this parameter is ignored and *BOTH is used.

***BOTH:** Data sent and received by the system is traced.

***SND:** Data sent by the system is traced.

***RCV:** Data received by the system is traced.

TRCFULL

Specifies the action the system takes when the trace buffer is full of data.

***WRAP:** The trace continues and overwrites the data in the buffer.

***STOPTRC:** The trace stops.

USRDTA

Specifies the amount of beginning and ending user data to trace.

Note:

For network server description traces, this parameter is ignored and *CALC is used.

***CALC:** The system determines the number of bytes that are traced. For LAN lines, this is the first 100 bytes. For other line types, the whole frame is traced.

***MAX:** Trace as much of frames as possible. For non-LAN, *MAX will be the equivalent of *CALC.

Element 1: Beginning Bytes

beginning-bytes: Specify the amount (in bytes) of beginning user data to be traced.

Element 2: Ending Bytes

ending-bytes: Specify the amount (in bytes) of ending user data to be traced.

CMNTRCOPTS

Specifies the type of data to be traced.

***ALLDTA:** All data is traced. No filtering is specified.

***RMTCTL:** The data traveling to and from a remote controller is traced.

***RMTMAC:** The data traveling to and from a remote medium access control (MAC) address is traced.

***RMTSAP:** The data traveling to and from a remote service access point (SAP) is traced.

***LCLSAP:** The data traveling to and from a local service access point (SAP) is traced.

***IPPCLNUM:** The data within an Internet Protocol (IP) number is traced.

***RMTIPADR:** The data traveling to and from a remote IP address is traced.

DDITRCOPTS

The DDITRCOPTS parameter is supported for upward compatibility of CL programs which contain the STRCMNTRC command. The CMNTRCOPTS parameter provides all of the same function as DDITRCOPTS and should be used instead of DDITRCOPTS.

RMTCTL

Specifies the remote controller receiving and sending the data to be traced.

RMTMAC

Specifies the remote medium access control address receiving and sending the data to be traced.

RMTSAP

Specifies the remote service access point receiving and sending the data to be traced.

LCLSAP

Specifies the local service access point receiving and sending the data to be traced.

IPPCLNUM

Specifies the Internet Protocol (IP) number to be traced.

***ICMP:** The Internet control message group is traced.

***IGMP:** The Internet group management group is traced.

***TCP:** The transmission control group is traced.

***EGP:** The exterior gateway protocol group is traced.

***IGP:** A private interior gateway group is traced.

***UDP:** The user datagram group is traced.

IP-protocol-number: Specify the Internet Protocol (IP) number to trace. Values range from 0 through 255.

RMTIPADR

Specifies the remote Internet Protocol (IP) address to be traced.

LMITRCOPTS

Specifies the type of data to be placed in the trace buffer.

***ALLDTA**: All data, including the local management interface (LMI), is placed in the trace buffer.

***NOLMI**: All data, except LMI data, is placed in the trace buffer.

***LMIONLY**: Only LMI data is placed in the trace buffer.

NWSTRCOPTS

Specifies the type of data to be placed in the trace buffer.

***NETBIOS**: All NetBIOS data is placed in the trace buffer.

***INTERNAL**: The communications processor operating system data is placed in the trace buffer.

***IPX**: The Internetwork Packet Exchange data is placed in the trace buffer.

Note:

This value is only used for the Integrated PC Server, also known as File Server I/O Processor (FSIOP), that is enabled for Netware processing. This value is not used by native IPX support.

TEXT Specifies the text that briefly describes the trace. More information on this parameter is in *Commonly used parameters*.

Note:

The text is used on spooled output and the System Service Tools (SST) Communications Trace screen displays.

***BLANK**: Text is not specified.

description: Specify up to 20 characters of text.

Example for STRCMNTRC

```
STRCMNTRC CFGOBJ(*QESLINE) CFGTYPE(*LIN)
```

This command starts a communications trace of line description QESLINE.

Error messages for STRCMNTRC

***ESCAPE Messages**

CPF2601

Line description &1 not found.

CPF2634

Not authorized to object &1.

CPF39AA

Trace &1 type &2 already exists

CPF39AB

Beginning or ending bytes exceeds maximum value

- CPF39AC**
Total of beginning and ending bytes exceeds maximum value
- CPF39AD**
&1 type &2 cannot be traced
- CPF39A6**
Storage could not be allocated
- CPF39A7**
Trace storage not available in communications processor
- CPF39A8**
Not authorized to communications trace service tool
- CPF39A9**
Error occurred during communications trace function
- CPF39BD**
Network interface description &1 not found
- CPF39BF**
Remote IP address not valid.
- CPF39B6**
Communications trace function cannot be performed
- CPF39C0**
Controller description &1 not found.
- CPF39C1**
Controller description &1 not valid.
- CPF39C2**
Number of user bytes to trace must be *CALC.

STRCPYSCN (Start Copy Screen) Command Description

STRCPYSCN Command syntax diagram

Purpose

The Start Copy Screen (STRCPYSCN) command allows the user to copy the displays of another display station to observe what is happening and diagnose problems.

If the STRCPYSCN command is used to copy displays from a source device that has the wide-display feature to an output device with a regular-width display, the command is accepted, but wide-display images are not shown and an informational message is sent to the target work station indicating that the display was not shown.

If the STRCPYSCN command is used to copy displays from a source device that supports graphic DBCS characters, the command is accepted and character information is shown, but graphic DBCS characters appear as single byte. No message is sent.

If the output device is not the requesting device, then the output device cannot be signed on. If the output device is signed on, a message is sent to the requester indicating that the device is not available for copying. If the source device is signed off after display copy has begun, the function automatically ends.

Note: The copy display function can be ended by either the source or target device.

Restrictions:

1. Permission must be given from the user of the source work station.
2. When a request is made to begin display image copying, a break message is sent to the user of the source work station to inform the user that the displays are going to be copied. The user must reply to this message before any displays are copied.
3. *REQUESTER is not valid for the SRCDEV or OUTDEV parameters when the command is submitted to batch.

Required Parameters

SRCDEV

Specifies the display station that is used as the source for the display images being copied.

***REQUESTER:** The displays are copied from the display station that issued this command.

source-device-name: Specify the name of the display station name (other than the *REQUESTER) whose displays are being copied.

Note:

If the display station specified for this parameter is a personal computer, it cannot be running PC organizer.

OUTDEV

Specifies the output device for the copying process.

***REQUESTER:** The displays are copied to the work station from which this command is issued. If *REQUESTER is specified on the SRCDEV parameter, *REQUESTER cannot be specified on the OUTDEV parameter.

***NONE:** The copied displays are not sent to a display station. If OUTDEV is *NONE, then OUTFILE must be specified.

display-device-name: Specify the name of the display station (other than the *REQUESTER) that shows the copied displays.

Optional Parameters

JOBQ Specifies the job queue that is used to submit the job that shows the displays from the source device on the target device when the requesting device is not the target device. When OUTDEV(*REQUESTER) is specified, this parameter is ignored, since it defaults to the values for the target display station and then a submit job command is not necessary.

The name of the job queue 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 library list is used to locate the job queue.

QCTL: The default job queue, QCTL, is used.

job-queue-name: Specify the name of the job queue where the job of processing the copied displays is submitted.

OUTFILE

Specifies the qualified name of the database file where the copied displays are directed. If the database file does not exist, this command creates a database file in the specified library. Displays can be copied to a database file instead of, or in addition to, being shown on another work station, thereby keeping a record of the displays that were copied and shown.

Note: A library name of QTEMP cannot be used.

***NONE:** The output is not directed to a database file.

The name of the database 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.

file-name: Specify the name of the database file that receives the output of the display.

If the user qualifies the database file name with *LIBL, but the system cannot find the file, the file is created in the QGPL library. Or, if a library name is specified, the file is created in the specified library.

OUTMBR

Specifies the name of the database file member to which the output is directed.

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 on the OUTFILE parameter.

member-name: Specify the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

Element 2: Operation to Perform on Member

***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.

Example for STRCPYSCN

```
STRCPYSCN SRCDEV(WS2) OUTDEV(*REQUESTER)
```

This command sends an inquiry message to the user of work station, WS2. The message indicates that the display station displays are about to be copied to another display station. If the user of that display does not wish this to happen, then a cancel (C) reply prevents the operation from beginning. To allow the operation to begin, the user responds with a go (G) to the message.

Error messages for STRCPYSCN

*ESCAPE Messages

CPF2207

Not authorized to use object &1 in library &3 type *&2.

CPF7AF4

Library QTEMP is not valid for OUTFILE keyword.

CPF7AF5

From device cannot be used with to device.

CPF7AF6

Device &1 not available.

CPF7AF7

Device name &1 not correct.

CPF9845

Error occurred while opening file &1.

CPF9860

Error occurred during output file processing.

STRDBMON (Start Database Monitor) Command Description

STRDBMON Command syntax diagram

Purpose

The Start Database Monitor (STRDBMON) command starts the collection of database performance statistics for a specified job or all jobs on the system. The statistics are placed in a specified database file and member. If the file or member do not exist, one is created based on the QAQQDBMN file in library QSYS. If the file or member do exist, the record format of the file is checked to see if it is the same.

Restrictions:

1. You cannot specify *ALL on the JOB parameter if the monitor has already been started with the JOB(*ALL) attribute. Only one specific monitor can be started on a specific job at a time. For example, STRDBMON JOB(*) followed by another STRDBMON JOB(*) within the same job is not allowed unless the job was ended using ENDDDBMON between the two STRDBMON command operations.
2. QTEMP cannot be specified as the library on the OUTFILE parameter unless JOB(*) was also specified.
3. You must have job control (*JOBCTL) special authority to start a monitor for a job other than the current job.
4. For multithreaded jobs, this command is not threadsafe and may fail when the OUTFILE parameter is a distributed file or a Distributed Data Management (DDM) file of type *SNA.
5. ➤ JOB(*ALL) requires the file specified on the OUTFILE parameter to be in a library that resides in the system ASP.◀

OUTFILE

Specifies the library and file name to which the performance statistics are written. If the file does not exist, one is created based on the QAQQDBMN file in library QSYS.

The name of the 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.

file-name: Specify the name of the file.

OUTMBR

Specifies the member name of the specified file to use for output. If the specified member does not exist, one is created.

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 on the OUTFILE parameter.

member-name: Specify the member name.

Element 2: Action on Member

***REPLACE**: The system clears the existing member and adds the new records.

***ADD**: The new records are added to the existing information in the specified database file member.

JOB Specifies the qualified name of the job and consists of as many as three elements. For example:

job-name

user-name/job-name

job-number/user-name/job-name

*N may be used in place of the user-name element to maintain position in the sequence. More information on this parameter is in Commonly used parameters.

***:** The current job is monitored.

***ALL**: All jobs on the system are monitored, including jobs waiting on job queues.

job-name: Specify the name of the job whose monitor is started.

user-name: Specify the name of the user of the job whose monitor is started.

job-number: Specify the number of the job whose monitor is started.

TYPE Specifies the type of database records to place in the outfile.

***SUMMARY**: Only the summary database monitor records are collected.

***DETAIL**: Both detail and summary database monitor records are collected.

FRCRCD

Specifies the number of records to hold in the buffer before forcing the records to be written to the file.

***CALC**: The command will calculate the number of records to use as this value.

number-of-records: Specify the number of records to hold. Valid values range from 0 through 32767.

COMMENT

Specifies the description that is associated with the database monitor record whose ID is 3018.

***BLANK**: Text is not specified.

text: Specify up to 100 characters of text.

Examples for STRDBMON

Example 1: Starting Database Monitoring for All Jobs

```
STRDBMON  OUTFILE(QGPL/FILE1)  OUTMBR(MEMBER1 *ADD)
          JOB(*ALL)  FRCRCD(10)
```

This command starts database monitoring for all jobs on the system. The performance statistics are added to the member named MEMBER1 in the file named FILE1 in the QGPL library. Ten records will be held before being written to the file.

Example 2: Starting Database Monitoring for a Specific Job

```
STRDBMON  OUTFILE(*LIBL/FILE3)  OUTMBR(MEMBER2)
          JOB(134543/QPGMR/DSP01)  FRCRCD(20)
```

This command starts database monitoring for job number 134543. The job name is DSP01 and was started by the user named QPGMR. The performance statistics are added to the member named MEMBER2 in the file named FILE3. Twenty records will be held before being written to the file. >>

Example 3: Starting Database Monitoring for a Specific Job to a file in a library in an Independent ASP

```
STRDBMON  OUTFILE(LIB41/DBMONFILE)  JOB(134543/QPGMR/DSP01)
```

This command starts database monitoring for job number 134543. The job name is DSP01 and was started by the user named QPGMR. The performance statistics are added to the member name DBMONFILE (since OUTMBR was not specified) in the file named DBMONFILE in the library named LIB41. This library may exist in more than one independent ASP (auxiliary storage pool); the library in the namespace of the originator's job will always be used. <<

Error messages for STRDBMON

*ESCAPE Messages

CPF1321

Job &1 user &2 job number &3 not found.

CPF436B

&1 can not be specified on the OUTFILE parameter.

CPF436C

Job &4 is already being monitored.

CPF436E

Job &1 user &2 job number &3 is not active.

STRDBRDR (Start Database Reader) Command Description

STRDBRDR Command syntax diagram

Purpose

The Start Database Reader (STRDBRDR) command starts a spooled reader to a database file. The reader reads a batch input stream from the database file and places the jobs onto one or more job queues. This command specifies the name of the database file and member from which the input stream is read, the name of the reader, and the names of the job queue and message queue that are used.

More than one reader can be active at the same time (as determined by the spooled subsystem description). Each database reader must have a unique reader name, and the specified file or member must be available. The reader can also be held or canceled if the Hold Reader (HLDRDR) or End Reader (ENDRDR) command is used.

Because each reader runs independent of the job that started it, users can continue doing other work on the system after they have started a reader. The reader is owned by the user who issues the STRDBRDR command.

Restriction: The specified database file either must consist of single-field records and must have an arrival sequence access path, or it must be a standard database source file.

Required Parameter

FILE Specifies the qualified name of the database file from which the input stream is to be read. The file must be available for allocation to the spooled reader before the reader can be started.

The name of the 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.

database-file-name: Specify the name of the database file.

Optional Parameters

MBR Specifies the name of the member in the specified file that contains the input stream that is read by the reader.

***FIRST:** The first member in the database file is used.

file-member-name: Specify the name of the member that contains the input stream that is read by the reader.

RDR Specifies the name of the spooled reader being started. Each reader name must be unique.

***FILE:** The name of the reader is the same as that of the database file specified by the FILE parameter.

reader-name: Specify the name by which the reader being started is identified.

JOBQ Specifies the qualified name of the job queue on which the spooled reader places jobs. An entry is placed on this job queue for each job in the input stream that specifies JOBQ(*RDR) on its Batch Job (BCHJOB) command. If *RDR is not specified on the BCHJOB command, the job queue specified in the BCHJOB command or in the job description is used. Each job in the input stream can have a different job queue.

Restriction: If the user starting the reader is not authorized for the job queue used by the reader for a job in the input stream, an error message is sent to the message queue specified by the MSGQ parameter and the job does not run.

The name of the job queue 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.

QBATCH: The job entry is placed on the system-supplied QBATCH job queue. QBATCH is the default if JOBQ(*RDR) is specified on the BCHJOB command.

job-queue-name: Specify the qualified name of the job queue to which each job read by this reader is sent if JOBQ(*RDR) is specified on its BCHJOB command.

MSGQ

Specifies the qualified name of the message queue to which the messages created by the reader are sent.

The name of the message queue 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.

QSYSOPR: The messages are sent to the system operator's message queue, QSYSOPR.

message-queue-name: Specify the name of the message queue to which messages created by the reader are sent.

Other Single Values

***REQUESTER:** The messages are sent to the workstation message queue of the workstation of the user who started the process. If this value is specified for a batch job, *OUTQ is used.

Example for STRDBRDR

```
STRDBRDR FILE(QGPL/BILLING)
```

This command starts a spooled reader that reads its input from the database file named BILLING, which is in the QGPL library. The reader name is also BILLING because the RDR parameter was not specified. The first member in the BILLING file contains the input stream to be processed. The default job queue QBATCH and the system-supplied system operator's message queue QSYSOPR are used by the database reader.

Error messages for STRDBRDR

*ESCAPE Messages

CPF1338

Errors occurred on SBMJOB command.

CPF2207

Not authorized to use object &1 in library &3 type *&2.

CPF3301

Reader &1 already started.

CPF3307

Job queue &1 in &2 not found.

CPF3330

Necessary resource not available.

CPF3362

Objects in QTEMP not valid for parameter values.

CPF3363

Message queue &1 in library &2 not found.

CPF3364

File &1 in library &2 not database file or DDM file.

CPF9812

File &1 in library &2 not found.

CPF9815

Member &5 file &2 in library &3 not found.

STRDBG (Start Debug) Command Description

STRDBG Command syntax diagram

Purpose

The Start Debug (STRDBG) command puts a job into debug mode and, optionally, adds as many as 20 programs and 20 class files and 20 service programs to debug mode. It also specifies certain attributes of the debugging session. For example, it can specify whether database files in production libraries can be updated while in debug mode.

Debug can operate in two environments at the same time. They are OPM (Original Program Model) and ILE (Integrated Language Environment). Some parameters for this command are applicable for both environments. Some parameters for this command are applicable only for either OPM or ILE. There is a statement in each parameter description stating the environment in which the parameter is applicable.

The Change Debug (CHGDBG) command can be used later in the job to change the attributes of the debug mode. Also, OPM programs can be added to or removed from the debugging session if they are specified in the Add Program (ADDPGM) or Remove Program (RMVPGM) commands. OPM programs added with ADDPGM command will be in the OPM debug environment.

When one job is servicing another job, and the STRDBG command is entered, all debug commands are valid for the job being serviced. If that job is held on a job queue, no further debug commands may be entered until that job is allowed to run. When the job starts, an initial breakpoint screen is displayed. From this screen, additional debug commands may be entered. To service another job, see the STRSRVJOB (Start Service Job) command. More information about debugging one job from another job is in the CL

Programming  book.

Restrictions:

1. This command is not valid in debug mode. To end debug mode, refer to the ENDDBG (End Debug) command.
2. This command cannot be used if the user is servicing another job, and that job is held, suspended, or ending.
3. This command is shipped with public *EXCLUDE authority, and the QPGMR, QSRV, and QSRVBAS user profiles have private authorities to use this command.
4. The user must have either *CHANGE authority to the program or *USE authority to the program and *SERVICE special authority.

Note: Do not precede an entry with an asterisk unless that entry is a special value that is shown (on the display itself or in the help information) with an asterisk.

Optional Parameters

PGM Specifies the qualified names of up to 20 programs to debug in the job. Before a program can be debugged, its name must be specified on this parameter, in the Add Program (ADDPGM) command or the Display Module Source (DSPMODSRC) command.

You can enter multiple values for this parameter. If you are on an entry display and you need additional entry fields to enter these multiple values, type a plus sign (+) in the entry field opposite the phrase "+ for more" and press the Enter key.

This parameter is applicable in both OPM and ILE environment.

***NONE:** No program names are specified at the start of the debugging session. The Add Program (ADDPGM) command or the Display Module Source (DSPMODSRC) command can be used to add programs later.

The name of the program 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.

program-name: Specify the names of up to 20 programs to debug. The user cannot debug two programs that have the same name at the same time in the OPM debug environment. In the ILE debug environment, 2 programs with the same name can be debugged if they are in different libraries.

DFTPGM

Specifies the name of the OPM 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 that specify *DFTPGM on their PGM parameter. That is, if a default program was previously specified, this parameter can change it.

This parameter is applicable only in the OPM environment.

***PGM:** The program named in the PGM parameter of *this* command is the default program for the job's debugging session. If PGM has more than one program name specified, the first OPM program named in the list that is added to the OPM debug environment is the default program. If PGM(*NONE) is specified or is the default, DFTPGM(*NONE) is also assumed when *PGM is specified here.

***NONE:** No program is specified as the default program. If the job has no default program, *DFTPGM cannot be specified on the PGM parameter of any other debug commands.

program-name: Specify the name of the program to use as the default program during debug mode. The same qualified name must also be specified in the PGM parameter of this command. A bound program cannot be specified on this parameter.

Note: If OPMSRC(*YES) is specified, the OPM program may be added to either the OPM debug environment or the ILE debug environment. If added to the ILE debug environment, then it can not be a default program for the OPM debug environment.

MAXTRC

Specifies the maximum number of trace statements that the system puts into the job's trace file before either stopping tracing, or wrapping around (overlying) on the trace file. When the trace file contains the maximum number specified, the system performs the actions specified in the TRCFULL parameter.

Note: Instruction stepping can be performed on a program being debugged in an interactive environment by setting the maximum number of trace statements to 1 and specifying *STOPTRC on the TRCFULL parameter.

This parameter is applicable only in the OPM environment.

200: Up to 200 trace statements can be put into the trace file before tracing is ended or wrapping on the file occurs.

maximum-trace-statements: Specify the maximum number of trace statements that can be in the trace file.

TRCFULL

Specifies what happens when the job's trace file is full; that is, it contains the maximum number of trace statements specified by the MAXTRC parameter.

This parameter is applicable only in the OPM environment.

***STOPTRC:** In a batch environment, tracing stops but the program continues processing. In an interactive environment, control is given to the user when a breakpoint occurs. If the user continues processing, a breakpoint occurs before processing each subsequent statement within the range of statements being traced, and the trace file is extended to contain the new entry.

***WRAP:** The trace file is overlaid with new trace statements as they occur, wrapping from the beginning of the file. The program completes processing without sending a message to indicate that wrapping has occurred. The trace file never has more than the maximum specified statements, and they are the most recently recorded statements.

UPDPROD

Specifies whether or not database files in a production library can be opened for updating records, or for adding new records, while the job is in debug mode. If not, the files must be copied into a test library before trying to run a program that uses the files.

This parameter is applicable in both OPM and ILE environment.

***NO:** Database files in production libraries cannot be updated while the job is in debug mode. Database files can be opened for reading only. This protects database files from unwanted updates while a program is being debugged. The exception to this is starting debug mode after a

production library is already opened. This is possible if the file is opened with SHARE(*YES) in a different program first. To prevent this, start debug mode before any files are opened, or use SHARE(*NO) on the Override Database File (OVRDBF) command for all files used in the programs being debugged. If this value is specified, some commands, which may include function keys, menu options, or pulldowns, may not work properly.

***YES:** Database files in production libraries can be updated while the job is in debug mode.

DSPMODSRC

Specifies whether the first display of the source debug program is shown when this command is processed and no errors occur.

This parameter is applicable only in the ILE environment.

***PGMDEP:** The showing of the source debug program display is dependent on the programs specified on the program prompt (PGM) parameter. If any of the programs are ILE programs, the display is shown. If any of the programs are OPM programs with source debug data, and OPMSRC(*YES) is specified, the display is shown. If all of the programs are OPM programs and OPMSRC(*NO) is specified, the display is not shown.

***NO:** The first display of the source debug program is not shown.

***YES:** The first display of the source debug program is shown.

SRCDBGPGM

Specifies the source debug program to be used.

This parameter is applicable only in the ILE environment.

***SYSDFT:** The system source debug program is used.

The name of the program 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.

program-name: Specify the name of the program to be used to debug programs.

UNMONPGM

Specifies the qualified name of the user-supplied program called when a message that is not monitored occurs in the job being debugged. When the program specified is called, it is passed parameters that identify the program name, the recursion level, the high-level language statement identifier, the machine instruction number at which the breakpoint occurred, the message that was not monitored, the message data, the length of the message data, and the message reference key.

This parameter is applicable only in the OPM environment.

The passed parameters have the following formats:

1. Program name (10 bytes). Specifies the name of the program in which the breakpoint was reached.
2. Recursion level (5 bytes). Specifies the recursion level number of the program in which the breakpoint 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. This is the statement identifier specified on the Add Breakpoint (ADDBKP)

command. If a machine instruction number is used to specify the breakpoint, this 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 breakpoint 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 a machine instruction number is passed on the third parameter, the numbers in the third and fourth parameters are the same.
5. Message ID (7 bytes). Specifies the ID of the message that was not monitored.
6. Message data (256 bytes). Specifies the first 256 bytes of message data sent with the message not monitored.
7. Message data length (5 bytes). Specifies the length of the message data sent with the message not monitored.
8. Message MRK (4 bytes). Specifies the message reference key (MRK) of the message not monitored.

All the parameter values are left-adjusted and padded on the right with blanks. When control returns to the program with the message that was not monitored, processing continues.

***NONE:** No program is called when a message that is not monitored occurs.

The name of the program 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.

program-name: Specify the name of the user-supplied program called when a message that is not monitored occurs in the job being debugged. After the program runs, control is returned to the interrupted program.

OPMSRC

Specifies that OPM programs will be debugged using the system source debug support (same as ILE source debug).

This parameter is valid for OPM CL, OPM RPG and OPM COBOL programs that were created with OPTION(*SRCDBG) with CRTCLPGM, CRTRPGPGM and CRTCLPGLPGM commands. Additionally, this parameter is valid for OPM RPG and OPM COBOL programs that were created with OPTION(*LSTDBG) with the CRTRPGPGM, CRTCLPGLPGM, CRTSQLRPG, CRTSQLCBL, and CRTRPTRPG commands. If the OPM program is not CL, RPG or COBOL and was not compiled with a valid debug option, then this parameter is ignored.

This parameter is applicable in both OPM and ILE environment.

ILE functions will now operate on OPM programs. For example, source level debug.

***NO:** OPM debug functions will be used for OPM programs.

***YES:** ILE debug functions will be used for OPM programs.

SRVPGM

Specifies the qualified names of up to 20 service programs to debug in the job. Before a service program can be debugged, its name must be specified on this parameter, or in the debugger user interface.

You can enter multiple values for this parameter. If you are on an entry display and you need additional entry fields to enter these multiple values, type a plus sign (+) in the entry field opposite the phrase “+ for more” and press the Enter key.

This parameter is applicable only in the ILE environment.

***NONE:** No service program names are specified at the start of the debugging session. The debugger user interface can be used to add service programs later.

The name of the program 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.

service-program-name: Specify the names of up to 20 service programs to debug.

CLASS

Specifies class file name to debug. A maximum of twenty class file names are allowed on the STRDBG command. Each name may be up to 5000 characters long. After debug is started more class files may be added to debug through the debugger user interface.

This parameter is applicable only in the ILE environment.

***NONE:** No class file names are specified at the start of the debugging session. The debugger user interface can be used to add class files later.

class-file-name: Specify the class file names to debug.

Examples for STRDBG

```
STRDBG PGM(TESTLIB/PAYROLL)
```

This command starts debug mode to debug the program PAYROLL, which is in the test library TESTLIB. If tracing is used, up to 200 trace statements can be stored in the trace before tracing stops. If program PAYROLL is a bound program, the Display Module Source display will be shown, giving the source for the module that contains the program entry point. Any database files updated by the PAYROLL program must be in a test library.

```
STRDBG CLASS('financial.payapplet')
```

This command starts debug mode to debug the class **payapplet** in the **financial** package.

Error messages for STRDBG

*ESCAPE Messages

CPF1999

Errors occurred on command.

STRDBGSVR (Start Debug Server) Command Description

STRDBGSVR Command syntax diagram

Purpose

The Start Debug Server (STRDBGSVR) command starts the debug server router function. Only one debug server router can be active at a time. Once started, the debug server router remains active until it is ended using the ENDDBGSVR command.

There are no parameters for this command.

Example for STRDBGSVR

```
STRDBGSVR
```

This command starts the debug server router function.

No error messages.

STRDIRSHD (Start Directory Shadowing) Command Description

STRDIRSHD Command syntax diagram

Purpose

The Start Directory Shadowing (STRDIRSHD) command submits a job to start the directory shadowing environment in the system work subsystem (QSYSWRK). The system administrator can use this command to restart the directory shadowing environment if it is not already active. Only one active directory shadowing environment per system is allowed. If the directory shadowing environment is already active, a warning message is issued.

The system work subsystem (QSYSWRK) automatically starts the directory shadowing environment as a prestart job when the subsystem is started.

To ensure the job submitted with this command is successful, use the Work with Jobs (WRKJOB) command using the job number returned in the message after issuing the STRDIRSHD command.

There are no parameters for this command.

Restriction: You must have job control (*JOBCTL) authority to use this command.

Example for STRDIRSHD

```
STRDIRSHD
```

This command submits a job to start the directory shadowing environment in the system work subsystem.

Error messages for STRDIRSHD

*ESCAPE Messages

CPF89A8

Unable to start job that controls directory shadowing.

STRDSKRGZ (Start Disk Reorganization) Command Description

STRDSKRGZ Command syntax diagram

Purpose

The Start Disk Reorganization (STRDSKRGZ) command allows the user to start the disk reorganization function for one or more auxiliary storage pools (ASP's). The user specifies a time limit that the function is to run for each ASP being reorganized. A message will be sent to the system history (QHST) log when the reorganization function is started for each ASP.

Unused space will be collected together within the ASP. This allows future large disk allocations to be done more efficiently.

Restrictions: You must have *ALLOBJ special authority to use this command.

Optional Parameters

ASP Specifies for which auxiliary storage pools the disk reorganization function is to be started.

***ALL:** Disk reorganization will be started for the system ASP (ASP number 1) and all basic ASPs (ASP numbers 2-32) defined to the system.

auxiliary-storage-pool-number: Specify the ASP for which disk reorganization is to be started. Valid ASP numbers are 1 to 32. Up to 32 ASP numbers may be specified.

ASPDEV

Specifies the name of the auxiliary storage pool (ASP) device for which the reorganization function is to be started.

***ALLAVL:** The reorganization function will be started for all ASP devices that currently have a status of 'Available'.

auxiliary-storage-device-name: Specify the name of the independent ASP device for which the reorganization function is to be started. Up to 32 ASP device names may be specified.

TIMLMT

Specifies the amount of time, in minutes, that the reorganization function will be allowed to run. When the time limit is reached the function will end.

The time limit specified is for each ASP being reorganized. For example, if ASP(*ALL) is specified and the machine has four ASP's configured and TIMLMT(60) is specified, four reorganization functions are started and each can run 60 minutes. If reorganization of any ASP has not completed after 60 minutes, it will be forced to end. This allows you to do disk reorganization incrementally.

***NOMAX:** There is no time limit for the reorganization function. For a large ASP that has many small unused disk areas, the reorganization function can take a long time to complete. If you start the reorganization function with *NOMAX and you want to force the function to end, you can use the End Disk Reorganization (ENDDSKRGZ) command.

time-limit: Specify the time limit that the reorganization function will be allowed to run. Valid values range from 1 to 9999 minutes.

Examples for STRDSKRGZ

Example 1: Starting Disk Reorganization for ASP 1

```
STRDSKRGZ ASP(1) TIMLMT(*NOMAX)
```

This command allows the user to start the disk reorganization function for ASP 1. The reorganization function will run until the ASP has been reorganized or it is ended through the ENDDSKRGZ command.

Example 2: Starting Disk Reorganization with a Time Limit

```
STRDSKRGZ ASP(*ALL) TIMLMT(60)
```

This command allows the user to start the disk reorganization function for each ASP on the system. Each reorganization function will have a time limit of sixty minutes. After sixty minutes, any reorganization functions which have not completed will be ended.

Example 3: Starting Disk Reorganization for an ASP Device

```
STRDSKRGZ ASPDEV(MYASP1) TIMLMT(*NOMAX)
```

This command allows the user to start the disk reorganization function for ASP device MYASP1. The reorganization function will run until the ASP has been reorganized or it is ended through the ENDDSKRGZ command.

Error messages for STRDSKRGZ

CPF1888

Disk reorganization for ASP &2 already started.

CPF1890

*ALLOBJ authority required for requested operation.

STRDKTRDR (Start Diskette Reader) Command Description

STRDKTRDR Command syntax diagram

Purpose

The Start Diskette Reader (STRDKTRDR) command starts a spooled reader to the specified diskette device to read a batch input stream and place it on the appropriate job queue. This command specifies the name of the diskette device from which the input stream is read; the volume and data file that the input stream is on; and the names of the job queue and message queue to be used.

On a system job, more than one reader can be active at the same time (as determined by the spooled subsystem description). Each reader must have a unique reader name and have its own input device assigned to it. A reader that has been started can be actively reading input or waiting for device input. The reader can also be held or ended if the Hold Reader (HLDRDR) or End Reader (ENDRDR) command is used. The reader is ended at end-of-file.

Because each reader runs independent of the job that started it, users can continue doing other work on the system after they have started a reader. The reader is owned by the user who issues the STRDKTRDR command.

Restriction: This command cannot be used to read data files of diskettes that are in the save/restore format.

Required Parameters

DEV Specifies the name of the diskette device that is used to read and enter an input stream into a job.

LABEL

Specifies the data file identifier of the file that is processed as an input stream. More information on this parameter is in Commonly used parameters.

Optional Parameters

RDR Specifies the name of the spooled reader that is started. Each reader name must be unique.

***DEV:** The name of the reader is the same as that of the diskette device specified in the DEV parameter.

reader-name: Specify the name by which the reader being started is identified.

VOL Specifies one or more volume identifiers used by the file. More information on this parameter is in Commonly used parameters.

***NONE:** No volume identifier is specified. Use the name of the current volume.

volume-identifier: Specify the identifiers of one or more volumes in the order that they are put on the device and read. Each identifier can have 6 alphanumeric characters or less. A blank is used as a separator character when listing multiple volumes.

CRTDATE

Specifies the creation date of the object.

Note: The creation date should not be specified if the date is not checked against the diskette date by the system. If the date written on the diskette containing the data file does not match the date specified on this parameter, an error message is sent to the message queue specified on the MSGQ parameter.

***NONE:** The creation date is not specified and no check is made.

creation-date: Specify the creation date of the data file that is read. The date should be in the format specified by the QDATFMT system value.

CODE Specifies the character code used. The code can be either extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII).

***EBCDIC:** The extended binary-coded decimal interchange code (EBCDIC) character set code is used.

***ASCII:** The ASCII character set code is used.

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 (CHGNETJOB) command.

Note: If both the user identified in the job description of the job being read and the user processing the Start Diskette Reader (STRDKTRDR) command are not authorized to the job queue on which the job should be placed, the job ends and a diagnostic message is placed in the job log. The input stream, continues to be processed, starting with the next job. If either user is authorized to the job queue, the job runs without error.

The name of the job queue 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.

QBATCH: The job entry is placed on the QBATCH job queue.

job-queue-name: Specify the name of the job queue to which each job read by this reader is sent if JOBQ(*RDR) is specified on its JOB command.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

***DEV:** The messages are sent to the message queue specified in the device description of the device named in the DEV parameter.

***REQUESTER:** The messages are sent to the workstation message queue of the workstation of the user who started the process. If this value is specified for a batch job, *OUTQ is used.

The name of the message queue 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.

message-queue-name: Specify the name of the message queue to which the diskette reader messages created by the reader are sent.

Example for STRDKTRDR

```
STRDKTRDR DEV(QDKT) LABEL(OCT24) VOL(SALES)
```

This command starts the spooled reader named QDKT, which reads diskette input from the device QDKT. Because *DEV was the default on the unspecified RDR parameter, the device name QDKT is also used as the reader name. The reader reads its input from the data file named OCT24 whose volume identifiers must be SALES. The default job queue QBATCH and the message queue QSYSOPR are used by the diskette reader.

Error messages for STRDKTRDR

*ESCAPE Messages

CPF1338

Errors occurred on SBMJOB command.

CPF2207

Not authorized to use object &1 in library &3 type *&2.

CPF3301

Reader &1 already started.

CPF3307

Job queue &1 in &2 not found.

CPF3330

Necessary resource not available.

CPF3347

Device &1 not found.

CPF3362

Objects in QTEMP not valid for parameter values.

CPF3363

Message queue &1 in library &2 not found.

CPF3367

Device &1 not diskette device.

STRDKTWTR (Start Diskette Writer) Command Description

STRDKTWTR Command syntax diagram

Purpose

The Start Diskette Writer (STRDKTWTR) command starts a spooling writer to the specified diskette device. The writer, which is a system job, takes spooled files from an output queue and produces (writes) the output on the diskette device. This command specifies the names of the diskette device and the diskette writer and the names of the output and message queues to be used.

More than one writer can be active at the same time (as determined by the spooling subsystem description). Each writer must have a unique writer name, its own output queue, and its own device. A writer that has been started can be actively writing output or waiting for a file to be put on the output queue. Optionally, the writer can be automatically stopped when it has processed all the files on the output queue. The writer can also be held or stopped if the Hold Writer (HLDWTR) or End Writer (ENDWTR) command is used.

Because each writer runs independent of the job that started it, users can continue doing other work on the system after they have started a writer.

Required Parameters

DEV Specifies the name of the diskette device to which the spooled file is written.

OUTQ Specifies the qualified name of the output queue.

The name of the output queue 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.

output-queue-name: Specify the name of the output queue.

Optional Parameters

WTR Specifies the name of the writer being started. Each writer name must be unique.

***DEV**: The name of the writer is the same as that of the diskette device specified in the DEV parameter.

writer-name: Specify the name by which the writer being started is identified.

MSGQ

Specifies the qualified name of the message queue to which messages are sent.

***DEV**: The messages are sent to the message queue specified in the device description of the device named in the DEV parameter.

***REQUESTER**: The messages are to be sent to the workstation message queue of the workstation of the user who started the writer. This value is not valid for batch jobs.

The name of the message queue 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.

message-queue-name: Specify the name of the message queue to which the diskette writer messages are sent.

AUTOEND

Specifies whether the writer stops automatically when the output queue has no more available entries for spooled files to be written to the diskette device.

Element 1: Stop Writer Option

***NO**: The writer does not end when the last available entry has been removed from the output queue; it waits for another spooled file entry to be put on the queue.

***YES**: The writer automatically ends after it has reached the state specified by the second part of the parameter (*NORDYF or *FILEEND).

Element 2: Conditions for Stopping Writer

***NORDYF**: The writer automatically ends when there are no ready files (all the available entries have been removed from the output file).

***FILEEND**: The writer stops after it has finished processing one spooled file.

FILE Specifies the name of the first (or only) spooled file processed by the spooling writer and written to diskette. If several files are available on the output queue, the next file produced is the first one available with the highest priority.

***NONE**: No spooled file name is specified; the first spooled file that becomes available on the output queue is processed first.

spooled-file-name: Specify the name of the spooled file that is the first (or only) spooled file to be written to diskette.

JOB Specifies the name of the job that created the spooled file being written to diskette. This parameter is valid only if a spooled file name is specified in the FILE parameter. If no job qualifier is given, all of the jobs currently in the system are searched for the simple job name.

A job identifier is a special value or a qualified name with up to three elements. For example:

```
*  
job-name  
user-name/job-name  
job-number/user-name/job-name
```

More information on this parameter is in Commonly used parameters.

*: The job that issued this STRDKTWTR command is the job that created the spooled file.

job-name: Specify the name of the job that created the spooled file.

user-name: Specify the name of the user of the job that created the spooled file.

job-number: Specify the number of the job that created the spooled file.

SPLNBR

Specifies the number of the spooled file that is processed first. This parameter is valid only if a spooled file name is specified in the FILE parameter. More information on this parameter is in Commonly used parameters.

***ONLY:** One spooled file from the job has the specified file name. The number of the spooled file is not necessary. If *ONLY is specified and more than one spooled file has the specified file name, a message is sent.

***LAST:** The spooled file with the highest number and the specified file name is used.

spooled-file-number: Specify the number of the specified file from the job on the specified output queue that is processed first.

Example for STRDKTWTR

```
STRDKTWTR DEV(QDKT) OUTQ(QDKT) AUTOEND(*YES)
```

This command starts a spooling writer to the diskette drive. The files written on the diskettes are on the IBM-supplied output queue QDKT. When all the files have been written (no more entries are on the QDKT output queue), the writer is automatically ended and the diskette drive is available for other uses.

Error messages for STRDKTWTR

*ESCAPE Messages

CPF0906

A duplicate job named &3/&2/&1 was found.

CPF1338

Errors occurred on SBMJOB command.

CPF1764

Writer already started for device &1.

CPF1842

Cannot access system value &1.

CPF2207

Not authorized to use object &1 in library &3 type *&2.

CPF3303

File &1 not found in job &5/&4/&3.

- CPF3305**
Output queue &1 in library &2 assigned to another writer.
- CPF3309**
No files named &1 are active.
- CPF3310**
Writer &1 already started.
- CPF3330**
Necessary resource not available.
- CPF3340**
More than one file with specified name found in job &5/&4/&3.
- CPF3342**
Job &5/&4/&3 not found.
- CPF3343**
Duplicate job names found.
- CPF3347**
Device &1 not found.
- CPF3357**
Output queue &1 in library &2 not found.
- CPF3362**
Objects in QTEMP not valid for parameter values.
- CPF3363**
Message queue &1 in library &2 not found.
- CPF3367**
Device &1 not diskette device.
- CPF3418**
Duplicate file &1 number &2 found in job.
- CPF3463**
Output queue for device &1 not found.
- CPF3478**
File &1 not found in job &5/&4/&3 on output queue &6 in library &7.

STREDU (Start Education) Command Description

STREDU Command syntax diagram

Purpose

The Start Education (STREDU) command starts the education session.

This command is shipped with public *EXCLUDE authority.

There are no parameters for this command.

Example for STREDU

```
STREDU
```

This command shows the following menus:

The Start Education Administration menu is shown for the Administrator.

The Select Course Option menu is shown for the new student that was enrolled by the Administrator.

The Specify your Name data entry screen is shown for the new student that was not enrolled.

The Select Course Option menu is shown for the enrolled student.

Error messages for STREDU

None

STREXPBRM (Start Media Expiration Processing for BRM) Command Description

Note: To use this command, you must have the 5722-BR1 (Backup Recovery and Media Services for iSeries) licensed program installed. For detailed information on the parameters of this command, see the online help.

STREXPBRM Command syntax diagram

Purpose

The Start Media Expiration Processing for BRM (STREXPBRM) command expires media that is in the BRMS media content information. When media is expired it can be reused in save processing. Volumes are expired when the expiration date or number of days is exceeded or when the volume has been reused. Volumes that are expired are marked with *YES in the *Expired* field on the Work with Media display. The STREXPBRM command can be scheduled to run periodically or can be processed as part of the STRMNTBRM command. The Media Expiration report can be produced by specifying *REPORT in the Action file action prompt (ACTION parameter). The report, if printed, is written to the printer file QP1AEP.

Example for STREXPBRM

Example 1: Running the Start Expiration for BRM Command

```
STREXPBRM ACTILCNT(2) ACTION(*EXPMED)
FILRET(*VERSION)
```

In the example, any volume that has 2 or less active files on it and whose media expiration is controlled using version control will be expired.

Error messages for STREXPBRM

None

STRFMA (Start Font Management Aid) Command Description

STRFMA Command syntax diagram

Purpose

The Start Font Management Aid (STRFMA) command shows the main Font Management Aid (FMA) menu. From this menu you can select the following options:

- Accessing workstation user-font and dictionary files on the FMA work file from the workstation.
- Copying user-defined characters from a workstation font file to the DBCS font table.
- Copying user-defined characters from the DBCS font table to the workstation font file.

There are no parameters for this command.

Example for STRFMA

STRFMA

This command displays the main FMA menu.

STRGRPJS (Start Group Using Job Scheduler) Command Description

Note: To use this command, you must have the 5722-JS1 (Job Scheduler for iSeries) licensed program installed.

STRGRPJS Command syntax diagram

Purpose

The Start Group using Job Scheduler (STRGRPJS) command allows you to start a job group.

Note: When referring to a job in this command, we are referring to an entry in Job Scheduler. An **entry** in Job Scheduler is a user-defined name for commands or programs that you want to process at scheduled times and dates. Job Scheduler jobs (entries) are not OS/400 objects.

The group will start with the first job you specify and submit all jobs that follow the job. For instance, if a group contains 10 jobs (group sequence numbers 1 through 10), and you specify job number 5 in the JOB parameter, then jobs 5 through 10 will be submitted as a result of the STRGRPJS command. If you want to process the entire group, you should specify the name of the job that has sequence number 1. You can specify the time and parameter data associated with the group.

Required Parameter

JOB Specifies the name of the job, the group in which the job resides and the sequence number of the job. This is a required parameter.

Element 1: Job

job-name: Specify the name of the job in the group.

Element 2: Group

group-name: Specify the name of the group to which the job is a member.

Element 3: Group sequence

group-sequence-number: Specify a sequence number ranging from 1 to 99.

Optional Parameters

TIME Specifies the time that you want this group to be submitted. A group can be submitted immediately or at the next scheduled time.

***SCHED:** The group is to be submitted by Job Scheduler at the next scheduled time.

***IMMED:** The group is submitted immediately. This is the default for a sequence number greater than 1.

PARM Specifies the name of the parameter that you want to use for the submitted group. Parameters that you specify must be defined in Job Scheduler.

You can enter multiple values for this parameter. If you are on an entry display and you need additional entry fields to enter these multiple values, type a plus sign (+) in the entry field opposite the phrase "+ for more", and press the Enter key.

***NONE:** The group that you are submitting does not require any parameters entered here.

Element 1: Parameter name

parameter-name: Specify the parameter that you want to use with this group.

Element 2: Parameter data

Specify the parameter data that you want to use for the parameter name that you specified in the PARM parameter.

Any character is valid for entry into this field. Validity will be checked to the extent of the use of the parameter within your programs or at the processing of the Job Scheduler job when submitted. The number of characters in the parameter data should correspond with the length placed in the **Parameter length** field. Data entered in a shorter length than the **Parameter length** field will pass data padded with blanks to the length of the parameter. Leading blanks and embedded blanks will be passed to the parameter in your request data exactly as keyed. Apostrophes (') used in parameter data must appear in pairs to be valid. If only one apostrophe is used, the parameter will be rejected with the error "Quotes (") in Parameter Data Must Appear In Pairs". You must correct this error before continuing.

parameter-data: Specify the parameter for the parameter specified above.

Example for STRGRPJS

Example 1: Starting a job group

```
STRGRPJS JOB(JOB02 APDAILY 20)
TIME(*IMMED)
```

In this example a group called APDAILY is started immediately starting with the job sequence number 20.

Error messages for STRGRPJS

None

STRHOSTSVR (Start Host Server) Command Description

STRHOSTSVR Command syntax diagram

Purpose

The Start Host Server (STRHOSTSVR) command is used to start the optimized host server daemons and the server mapper daemon.

There is one server daemon for each of the host server types. In addition, there is one server mapper daemon for all host servers which provides support for client applications to obtain a particular host server daemon's port number. This port number is then used by the client application to connect to the host server's daemon. The daemon accepts the incoming connection request and routes it to the server job for further processing.

The daemons are batch jobs submitted to either the QSYSWRK or QSERVER subsystem, depending on the value or values specified for the SERVER keyword. All daemon jobs are submitted to the QSYSWRK subsystem with the exception of the *DATABASE and *FILE server daemons which are submitted to the QSERVER subsystem.

In order for the server daemons and the server mapper daemon to start successfully, the QSYSWRK subsystem and, for *DATABASE and *FILE server, the QSERVER subsystem must be active. If the required subsystem is not active, then the submission of the daemon job will fail. Additionally, the

QUSRWRK subsystem or the user-defined subsystem must be active in order to start the associated server jobs. All associated server jobs can run in the QUSRWRK subsystem or a user-defined subsystem, except for the following:

- **»** Server jobs QPWFSERVSO and QPWFSERVSS - these jobs run in the QSERVER subsystem or a user-defined subsystem
- Server job QPWFSERVS2 - this job runs in the QSERVER subsystem **«**
- Server job QIWVPPJT - this job runs in the QSYSWRK subsystem
- Server job QTFPJTCP - this job runs in the **»** QSERVER **«** subsystem

There are no server jobs associated with the server mapper daemon.

The server daemons must be active in order to allow client applications to establish a connection with the host server using sockets communication support. Once started, the server daemons and the server mapper daemon remain active until they are ended explicitly using the End Host Server (ENDHOSTSVR) command or an error occurs.

Restrictions:

- This command is used only for enabling client applications to communicate with any of the host servers using sockets communication support. This command does not start any of the APPC host servers; these are started as a result of a program start request.
- Only one server daemon can be active for a specific server type. Requests to start a server daemon that is already active will result in an informational message to the user issuing this command.

Required Parameter

SERVER

Specifies the host server daemons to be started by this command.

***ALL:** All of the server daemons and the server mapper daemon are started.

***CENTRAL:** The central server daemon is started in the QSYSWRK subsystem. The daemon job is named QZCSRVD. The associated server job is named QZCSRVS.

***DATABASE:** The database server daemon is started in the QSERVER subsystem. The daemon job is named QZDASRVSD. The associated server jobs are named QZDASOINIT, QZDASSINIT, and QTFPJTCP.

***DTAQ:** The Data Queue server daemon is started in the QSYSWRK subsystem. The daemon job is named QZHQRVD. The associated server job is named QZHQRV.

***FILE:** The file server daemon is started in the QSERVER subsystem. The daemon job is named QPWFSERVSD. The associated server jobs are named QPWFSERVSO, QPWFSERVSS, and QPWFSERVS2.

***NETPRT:** The network print server daemon is started in the QSYSWRK subsystem. The daemon job is named QNPSEVRD. The associated server jobs are named QNPSEVRS and QIWVPPJT.

***RMTCMD:** The remote command and distributed program call server daemon is started in the QSYSWRK subsystem. The daemon job is named QZRCSRVD. The associated server job is named QZRCSRVS.

***SIGNON:** The signon server daemon is started in the QSYSWRK subsystem. The daemon job is named QZSOSGND. The associated server job is named QZSOSIGN.

***SVRMAP:** The server mapper daemon is started in the QSYSWRK subsystem. The daemon job is named QZSOSMAPD.

Optional Parameter

RQDPCL

Specifies which communication protocols are required to be active for the host server daemons to start.

Single Values

***ANY:** >> The TCP/IP communication protocol << must be active at the time the STRHOSTSVR command is issued. If >> TCP/IP is not << active, escape message PWS300D will be issued and the host server daemons will not be started. A diagnostic message (PWS3008 >>) << will be issued >> if TCP/IP << is found to be inactive.

***NONE:** No communication protocols need to be active at the time the STRHOSTSVR command is issued for the host server daemons to start. No messages will be issued for protocols which are inactive.

>> << ***TCP:** The TCP/IP communication protocol must be active at the time the STRHOSTSVR command is issued. If TCP/IP is not active, diagnostic message PWS3008 and escape message PWS300D will be issued and the host server daemons will not be started. >> <<

Examples for STRHOSTSVR

Example 1: Starting All Host Server Daemons

```
STRHOSTSVR  SERVER(*ALL)
```

This command starts all of the server daemons and the server mapper daemon, as long as at least one communication protocol is active.

Example 2: Starting Specific Server Daemons

```
STRHOSTSVR  SERVER(*CENTRAL *SVRMAP)
            RQDPCL(*NONE)
```

This command starts the central server daemon and the server mapper daemon in the QSYSWRK subsystem, even if no communication protocols are active.

Example 3: Specifying One Required Protocol

```
STRHOSTSVR  SERVER(*ALL) RQDPCL(*TCP)
```

This command starts all of the host server daemons and the server mapper daemon, as long as the TCP/IP communications protocol is active.

Error messages for STRHOSTSVR

*ESCAPE Messages

PWS300D

Unable to start host server daemon jobs.

PWS3006

Errors occurred starting server daemon jobs.

STRHTTPCRL (Start HTTP Crawl) Command Description

Note: To use this command, you must have the 5722-DG1 (HTTP Server) product installed.

Purpose

The Start HTTP (STRHTTPCRL) starts a crawling session.

Restriction: You must have *IOSYSCFG special authority to use this command.

Required Parameters

OPTION

Specifies the function to perform.

The possible values are:

*CRTDOCL

Specifies that a document list is created.

*UPDDOCL

Specifies that a document list is updated.

DOCLIST

Specifies the document list file to store the list of path names of each document found.

OBJECTS

Specifies the set of URL and options objects to use to create or update a document list. Specify this parameter to create or update a document list instead of individual values.

URLOBJ

Specifies a URL object name.

OPTOBJ

Specifies an options object name.

DOCDIR

Specifies the document directory where the downloaded documents are stored.

LANG Specifies the language associated with the URL.

The possible values are:

*ARABIC

*BALTIC

*CENTEUROPE

*CYRILLIC

*ESTONIAN

*GREEK

*HEBREW

*JAPANESE

*KOREAN

*SIMPCHINESE

*TRADCHINESE

*THAI

***TURKISH**

***WESTERN**

URLLIST

Specifies the URL, URL filter, and crawl depth. Used only for OPTION (*CRTURLOBJ) or OPTION (*UPDURLOBJ).

URL Specifies the URL to start the crawl.

URLFTR

Specifies the URL filter or domain name restriction.

MAXDEPTH

Specifies the maximum crawl depth. The top level is the URL and is known as level 0. Links on level 0 go to level 1 pages. Links on level 1 pages go to level 2 pages, and so on.

MAXSIZE

Specifies the maximum size file to download. Only files that are within this limit are downloaded.

MAXSTGSIZE

Specifies the maximum storage size for downloaded files.

MAXTHD

Specifies the maximum number of threads.

MAXRUNTIME

Specifies the maximum run time. The maximum amount of time in hours and minutes that the crawl should run. Any file in the process of being downloaded when time expires is completely downloaded.

LOGFILE

Specifies the path and name of the crawl activity log file. This file contains the URLs that are found and downloaded as well as any exceptions (such as empty file) that are found during the crawl.

CLRLOG

Specifies if the log file should be cleared before writing to it.

Optional Parameters

PRXSVR

Specifies the proxy server for HTTP.

PRXPORT

Specifies the proxy server port for HTTP.

SECPRXSVR

Specifies the proxy server for HTTPS.

SECPRXPORT

Specifies the proxy server port for HTTPS.

Error messages for STRHTTPCRL

***ESCAPE messages**

HTP160C

Request to create or append to a document list failed. Reason &1.

HTP166E

Request to print the status of a document list failed. Reason &1.

STRINFSKR (Start InfoSeeker) Command Description

STRINFSKR Command syntax diagram

Purpose

The Start InfoSeeker (STRINFSKR) command starts InfoSeeker, which is the iSeries 400 version of BookManager READ. InfoSeeker allows you to read and search online books. When you enter the command without parameters, your default list of bookshelves and books is displayed.

Restrictions:

1. Do not use the STRINFSKR command if another job is running commands such as the SAVSHF, RSTSHF, RCLDLO, SAVDLO, or RSTDLO commands. These commands may have obtained exclusive use of internal objects which can prevent you from accessing books.
2. You must be enrolled in the system distribution directory to use the STRINFSKR command. Use the Add Directory Entry (ADDDIRE) command to update the directory.

Optional Parameters

OPTION

Specifies what should be opened or displayed.

***LIST:** Specifies your default list of books and bookshelves to be displayed. Initially, those are all the bookshelves you can access.

***BOOK:** Specifies a book should be opened.

***SHELF:** Specifies a bookshelf should be opened.

SHELF

Specifies which bookshelf or list of bookshelves to open.

Element 1: Bookshelf Name

bookshelf-name: Specify the name of a bookshelf, such as QBKADA01, to open.

Element 2: Bookshelf Search Path

***BOOKPATH:** Specifies the book search path as specified in the QBOOKPATH system value.

directory: Specify the directory that contains the bookshelf you want to open. A folder directory begins with a slash (/), followed by QDLS, another slash, and the folder name. For example: /QDLS/QBKBOOKS/BOOKS.

BOOK Specifies a specific book to open.

Element 1: Book Name

book-name: Specify the name of a book, such as QBKA9100, to open.

Element 2: Book Search Path

***BOOKPATH:** Specifies the book search path as specified in the QBOOKPATH system value.

directory: Specify the directory that contains the book you want to open. A folder directory begins with a slash (/), followed by QDLS, another slash, and the folder name. For example: /QDLS/QBKBOOKS/BOOKS.

TOPIC

Specifies a specific topic to open a book to.

***NONE:** No topic is specified.

topic-identifier: Specify a topic identifier, such as '1.2.3' or CONTENTS to open a book to. Up to 64 characters can be entered. If the topic identifier has periods (.), enclose the identifier in apostrophes (').

SEARCH

Specifies the book or bookshelf you want to search. This parameter also displays the results of the search.

***NONE:** No book or bookshelf is searched.

'text': Specify up to 128 characters of text enclosed in apostrophes that you want to search for. For example, 'stopping batch jobs'.

You can use an asterisk (*) as a wildcard character to specify the character to be used as a substitute for unknown characters or words in a search. For example, if you search for the phrase 'printer 1*', your search results include every phrase that begins with 'printer 1' (such as, 'printer 12' or 'printer 153').

You can use a comma (,) as a search-phrase separator character to split apart search phrases. For example, the search request 'conduct, behavior' uses a comma as the separator to tell the search to look for either the word 'conduct' or the word 'behavior'.

Examples for STRINFSKR

Example 1: Listing Your Default Bookshelves

```
STRINFSKR
```

This command displays a list of all your default bookshelves.

Example 2: Opening a Specific Bookshelf

```
STRINFSKR OPTION(*SHELF) SHELF(MYSHELF)
```

This command displays the MYSHELF bookshelf.

No error messages.

STRIDD (Start Interactive Data Definition Utility) Command Description

STRIDD Command syntax diagram

Purpose

The Start Interactive Data Definition Utility (STRIDD) command displays the main IDDU menu. From this menu, you can select an option that allows you to work with data definitions, data dictionaries, files, and libraries, or use related commands and office tasks.

There are no parameters for this command.

Example for STRIDD

```
STRIDD
```

This command displays the main IDDU menu.

No error messages.

STRITF (Start Interactive Terminal Facility) Command Description

STRITF Command syntax diagram

Purpose

The Start Interactive Terminal Facility (STRITF) command allows the user to send and receive data and file members for asynchronous display stations. The system user can also send documents. The user must vary on asynchronous communications descriptions before the Interactive Terminal Facility (ITF) can be used.

Required Parameter

RMTLOCNAME

Specifies the remote location name of the system with which this object communicates.

Example for STRITF

```
STRITF CHICAGO
```

This command allows the user to communicate with the remote location CHICAGO.

No error messages.

STRIPSIFC (Start IP over SNA interface) Command Description

STRIPSIFC Command syntax diagram

Purpose

The Start IP over SNA Interface (STRIPSIFC) command is used to start an AF_INET sockets over SNA interface (an IP address by which this local host is known on the SNA transport).

Restriction:

Only eight (8) AF_INET sockets over SNA interfaces can be active on a single host. If the maximum number of interfaces is active and you want to start another interface, you must first end one or more interfaces using the End IP over SNA Interfaces (ENDIPSIFC) CL command.

Required Parameter

INTNETADR

Specifies the internet address of an inactive interface that had previously been added to the IP over SNA configuration with the ADDIPSIFC CL command. The internet address is specified in the form *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.

Example for STRIPSIFC

```
STRIPSIFC INTNETADR('9.5.1.248')
```

This command activates (starts) the interface with IP address 9.5.1.248.

Error messages for STRIPSIFC

*ESCAPE Messages

CPFA10F

IP over SNA interface &1 not started.

STRJS (Start Job Scheduler) Command Description

Note: To use this command, you must have the 5722-JS1 (Job Scheduler for iSeries) licensed program installed.

STRJS Command syntax diagram

Purpose

The Start Job Scheduler (STRJS) command allows you to start the Job Scheduler job monitor or to capture job information for application software. If you start the job monitor and there are jobs with scheduled dates and times less than the current date and time, you are transferred to the Start Monitor display where you can choose whether you want to start the Job Scheduler monitor without reviewing scheduled jobs or delay starting the monitor until after you have reviewed the pending jobs.

Optional Parameter

OPTION

Specifies whether you want to start the Job Scheduler monitor or you want to begin capturing job information.

***MONITOR:** Start the Job Scheduler monitor.

***CAPTURE:** Begin capturing job information from application software.

Example for STRJS

Example 1: Starting the Job Scheduler Monitor

```
STRJS OPTION(*MONITOR)
```

In this example the Job Scheduler monitor is started. The monitor stays active until you process the End Job Scheduler (ENDJS) command.

Error messages for STRJS

None

STRJOBTRC (Start Job Trace) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

STRJOBTRC Command syntax diagram

Purpose

The Start Job Trace (STRJOBTRC) command starts the job tracing function to collect performance statistics about the specified job. It issues the following command:

```
TRCJOB MAXSTG(maxstg) TRCFULL(*STOPTRC)
```

The MAXSTG value used is either the STRJOBTRC default or a specified value. The STRJOBTRC command issues a Service Job (SRVJOB) command if a job other than the current job is specified.

Once job tracing is started, a trace record is created for every external (program) call and return, exception, message, and work station wait in the job. At least two, and usually more, trace records are created for every I/O statement (open, close, read, or write) in a high-level language program.

After the target programs have been run, the End Job Trace (ENDJOBTRC) command is used to turn tracing off, record the collected information in a database file, and optionally produce reports used to analyze the data. The Print Job Trace (PRTJOBTRC) command may also be used to print the same report at any time thereafter. Alternatively, the command

```
TRCJOB SET(*CNL)
```

may be used to cancel the job trace (turn it off without recording any collected data).

Tracing has a significant impact on the performance of the current job. It also affects the performance of the system in general, but to a lesser extent.

Optional Parameters

MAXSTG

Specifies the maximum amount of storage to allocate to the internal buffer where every trace record that is created is stored. If the buffer is filled, tracing is automatically turned off.

1024: A 1024KB (1 megabyte) trace buffer is used. This size can handle about 14,000 trace records, which is sufficient in most cases.

maximum-K-bytes: Specify the size of the trace buffer in kilobytes. Each kilobyte can hold almost 14 trace records.

JOB Specifies the job to trace.

A job identifier is a special value or a qualified name with up to three elements. For example:

```
*  
job-name  
user-name/job-name  
job-number/user-name/job-name
```

*****: The current job is traced.

job-name: Specify the name of the job that is traced. User name and job number qualifiers may be specified.

user-name: Specify the name of the user of the job that is traced.

job-number: Specify the number of the job that is traced.

Example for STRJOBTRC

```
STRJOBTRC MAXSTG(512)
```

This command starts collecting trace data for the current job. It uses a 512KB trace buffer, which is large enough to hold about 7000 trace records.

Error messages for STRJOBTRC

*ESCAPE Messages

None.

STRJRN (Start Journal) Command Description

STRJRN Command syntax diagram

Purpose

The Start Journal (STRJRN) command is used to start journaling changes (made to an object or list of objects) to a specific journal. The object types which are supported through this interface are Data Areas (*DTAARA), Data Queues (*DTAQ), Byte Stream Files (*STMF), Directories (*DIR), and Symbolic Links (*SYMLNK). Only objects of type *STMF, *DIR or *SYMLNK that are in the Root ('/'), QOpensys, and User-defined file systems are supported. For more information about the possible journal entries which can be sent, see the [Journal management](#) article in the Information Center.

The user can specify that only the after image or both the before and the after images of an object of type *DTAARA be journaled. Before images are necessary to remove journaled changes using the Remove Journaled Changes (RMVJRNCHG) command.

After journaling begins for the object, the user should save the journaled object to preserve its journal attribute information. Also, the object must be saved because, for example, journaled changes cannot be applied to a version of the object that was saved before journaling was in effect.

Note: For other ways to start journaling see the following commands:

- Access Paths - Start Journal Access Path (STRJRNAP)
- Physical Files - Start Journal Physical File (STRJRNPF)
- Other Objects - Start Journal Object (STRJRNOBJ)

Restrictions:

1. The object must not be journaling changes to another journal.
2. The maximum number of objects that can be associated with one journal is 250,000. This maximum includes objects whose changes are currently being journaled, objects for which journaling was ended while the current receiver was attached, and journal receivers that are or were associated with the journal while the current journal receiver is attached. If the number of objects is larger than this maximum, journaling does not start.
3. The specified journal cannot be a remote journal.
4. The specified journal and object must reside in the same Auxiliary Storage Pool (ASP).
5. For data areas, only local external data area objects may be journaled. The special data areas (*LDA, *GDA, *PDA) and DDM data areas cannot be journaled.
6. For data queues, only local data queues are supported. DDM data queues cannot be journaled.
7. At least one of parameter OBJ or OBJFID must be specified.

Required Parameters

OBJ Specifies a maximum of 300 object path names for which changes are to be journaled. Only objects whose path name identifies an object of type *STMF, *DIR, *SYMLNK, *DTAARA or *DTAQ are supported.

Element 1: Object Name

'object-path-name': Specify the path name of the object for which changes are to be journaled.

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. Symbolic links within the path name will not be followed. If the path name begins with the tilde (~) character, then the path is assumed to be relative to the appropriate home directory.

Additional information about path name patterns is in the Integrated file system topic in the File systems and management category of the Information Center.

Element 2: Include or Omit

The second element specifies whether names that match the pattern should be included or omitted from the operation. Note that in determining whether a name matches a pattern, relative name patterns are always treated as relative to the current working directory.

***INCLUDE:** The objects that match the object name pattern are to be journaled, unless overridden by an ***OMIT** specification.

***OMIT:** The objects that match the object name pattern are not to be journaled. This overrides an ***INCLUDE** specification and is intended to be used to omit a subset of a previously selected path.

OBJFID

Specifies a maximum of 300 file identifiers (FID) for which changes are to be journaled. FIDs are a unique identifier associated with integrated file system related objects. This field is input in hexadecimal format. Only objects whose FID identifies an object of type ***STMF**, ***DIR**, ***SYMLNK**, ***DTAARA** or ***DTAQ** are supported.

file-identifier: Objects identified with the FID are journaled.

JRN Specifies the path name of the journal that receives the journaled changes.

journal-path-name: Specify the name of the journal that receives the journaled changes.

Optional Parameters

SUBTREE

Specifies whether the directory subtrees are included in the start journal operation.

Note:

This parameter is ignored if the OBJ parameter is not specified.

***NONE:** Only the objects that match the selection criteria are processed. The objects within selected directories are not implicitly processed.

***ALL:** All objects that meet the selection criteria are processed in addition to the entire subtree of each directory that matches the selection criteria. The subtree includes all subdirectories and the objects within those subdirectories.

PATTERN

Specifies a maximum of 20 patterns to be used to include or omit objects for the start journal operation.

Note:

This parameter is ignored if the OBJ parameter is not specified.

Element 1: Name Pattern

'*': All objects that match the input OBJ parameter are to be included into the start journal operation or omitted from the start journal operation.

name-pattern: Specify the pattern to either include or omit objects for the start journal operation.

Only the last part of the path name will be considered for the name pattern match. Path name delimiters are not allowed in the name pattern. 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.

If the Name Pattern parameter is not specified the default will be to match all patterns.

Additional information about path name patterns is in the Integrated file system topic in the File systems and management category of the Information Center.

Element 2: Include or Omit

The second element specifies whether names that match the pattern should be included or omitted from the operation. Note that in determining whether a name matches a pattern, relative name patterns are always treated as relative to the current working directory.

Note: The SUBTREE parameter specifies whether the subtrees are included or omitted.

***INCLUDE:** The objects that match the object name pattern are included into the start journal operation unless overridden by an *OMIT specification.

***OMIT:** The objects that match the object name pattern are not to be included into the start journal operation. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

INHERIT

Specifies whether new objects created within a journaled directory should inherit the journal options and the journal state of its parent directory.

***NO:** New objects created within the directory will not inherit the journal options and journal state of the parent directory.

***YES:** New objects created within the directory will inherit the journal options and journal state of the parent directory.

IMAGES

Specifies the kinds of images that are written to the journal receiver for changes to objects.

***AFTER:** Only *after* images are generated for changes to objects.

***BOTH:** The system generates both *before* and *after* images for changes to objects.

Note: The value *BOTH is only valid for *DTAARA objects.

OMTJRNE

Specifies the journal entries that are omitted.

***NONE:** No entries are omitted.

***OPNCLOSYN:** Open, close and force entries are omitted. Open, close and force operations on the specified objects do not generate open, close and force journal entries. This prevents the use of TOJOB0 and TOJOB1 entries on the Apply Journal Changes (APYJRNCHG) command, but it saves some storage space in the journal receivers.

Note:

The value *OPNCLOSYN is only valid for *DIR and *STMF objects.

Examples for STRJRN**Example 1: Start Journaling with Omit of Directory**

```
STRJRN OBJ('/mypath' *INCLUDE) ('/mypath/myobject' *OMIT))
      JRN('/QSYS.LIB/MYLIB.LIB/JRNLA.JRN')
```

This command journals all changes to all objects supported by this command within the first-level of directory '/mypath' except '/mypath/myobject' to journal '/QSYS.LIB/MYLIB.LIB/JRNA.JRN'. None of the objects within the subdirectories of '/mypath' will be journaled.

Only the *after* images of updated records are written to the journal.

Example 2: Start Journaling with Pattern Matching

```
STRJRN OBJ('/mypath' *INCLUDE) ('/mypath/myobject.txt' *OMIT))
      JRN('/QSYS.LIB/MYLIB.LIB/JRNLA.JRN') SUBTREE(*ALL)
      PATTERN('*.*.TXT' *INCLUDE) OMTJRNE(*OPNCLOSYN)
```

This command journals changes to all objects that match pattern '*.txt' in directory '/mypath' except object '/mypath/myobject.txt'. The open, close and force entries are not journaled.

Only the *after* images of updated records are written to the journal.

Example 3: Start Journaling with Omit by Pattern

```
STRJRN OBJ('/mypath/my*' *INCLUDE))
      JRN('/QSYS.LIB/MYLIB.LIB/JRNLA.JRN')
      PATTERN('*.*.DTA*' *OMIT))
```

This command journals changes to all objects within the first-level directories that match the pattern for path '/mypath/my*' and will omit all objects that match pattern '*.DTA*' (objects of type *DTAARA and *DTAQ).

Only the *after* images of updated records are written to the journal.

Example 4: Start Journaling using File Identifiers

```
STRJRN OBJFID(000000000000000007E09BDB000000009
             000000000000000009E09BDB00000000A)
      JRN('/QSYS.LIB/MYLIB.LIB/JRNLA.JRN')
```

This command journals all changes to the objects represented by the specified file identifiers to journal '/QSYS.LIB/MYLIB.LIB/JRNLA.JRN'.

Only the *after* images of updated records are written to the journal.

Example 5: Start Journaling on a set of Data Queues

```
STRJRN OBJ('/QSYS.LIB/MYLIB.LIB/MYDATA*.DTAQ'))
      JRN('/QSYS.LIB/MYLIB.LIB/MYJRN.JRN')
```

This command starts the journaling of all changes to the objects of type *DTAQ in library MYLIB that begin with the characters 'MYDATA'.

Error messages for STRJRN

***ESCAPE Messages**

CPFA0D4

File system error occurred.

CPF700A

&1 of &2 objects have started journaling.

CPF705A

Operation failed due to remote journal.

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.

» CPF9825

Not authorized to device &1. «

CPF9830

Cannot assign library &1.

» CPF9873

ASP status is preventing access to object.

CPF9875

Resources exceeded on ASP &1. «

STRJRNAP (Start Journal Access Path) Command Description

STRJRNAP Command syntax diagram

Purpose

The Start Journal Access Path (STRJRNAP) command is used to start the journaling of the access path, or access paths, for all members of a database file to a specific journal. The access paths of new members are also journaled.

If a physical file is specified, journaling can be started for its access paths. When access path journaling is started for a physical file, only the access paths for the physical file members are journaled. Journaling for any logical file access paths is started only when access path journaling is started for the logical file.

The journal entries created after running this command cannot be used in any apply or remove journaled changes operation. These entries are used only to recover the access path without rebuilding it after an abnormal system operation ending.

If you start journaling your access paths, consider specifying RCVSIZOPT(*RMVINTENT) on either the Create Journal (CRTJRN) or the Change Journal (CHGJRN) command for this journal. This will reduce the additional storage required to do access path journaling.

If you do not want the overhead of managing the access path journaling yourself, consider taking advantage of the system-managed access-path protection support. For more information, see the [Journal management article in the Information Center](#), and the [Edit Recovery for Access Paths \(EDTRCYAP\)](#) or the [Change Recovery for Access Paths \(CHGRCYAP\)](#) command.

Note: For other ways to start journaling see the following commands:

- Integrated File System objects - Start Journal (STRJRN)
- Physical Files - Start Journal Physical File (STRJRNPf)
- Other Objects - Start Journal Object (STRJRNOBJ)

Restrictions:

1. Before journaling an access path, all physical files over which the access path is built must first be journaled to the same journal that is used to journal the access path. Even if all physical file members for a particular physical file are removed from the access path of a logical file, all physical files must still be journaled to the same journal before journaling the access path.
2. All access paths to be journaled must specify MAINT(*IMMED) or MAINT(*DLY) and FRCACCPH(*NO).
3. If only after images are being journaled for the physical file members, the system automatically starts journaling the before and after images for the physical file once journaling is started for any access path built over the physical file. When journaling ends for the access paths, the system automatically stops journaling the before images for the physical file and again only journals the after images.
4. Overrides are not applied to files specified on the FILE parameter.
5. This command cannot be used on or with a remote journal.

Required Parameters

FILE Specifies a maximum of 50 qualified names of the database files for which access paths are to be journaled.

The name of the 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.

file-name: Specify the name of the file for which access paths are to be journaled.

JRN Specifies the qualified name of the journal that is to receive the journaled access path changes.

The name of the journal 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.

journal-name: Specify the name of the journal that is to receive the journaled access path changes.

Example for STRJRNAP

```
STRJRNAP FILE(MYFILE) JRN(MYLIB/JRNLA)
```

This command journals all access paths for all members in file MYFILE (found using the library search list) to journal JRNLA in library MYLIB.

Error messages for STRJRNAP

***ESCAPE Messages**

CPF6971

Damage prevents object &1 from being journaled.

CPF6972

Cannot allocate access path for file &1 in &2.

CPF7003

Entry not journaled to journal &1. Reason code &3.

CPF7004

Maximum number of objects journaled to journal &1.

CPF7008

Cannot start or end access path journaling for file &1.

CPF7009

Not all based-on files being journaled to &3.

CPF7011

Not enough storage.

CPF703C

DDL transaction prevents journaling operation.

CPF703D

DDL transaction prevents journaling operation.

CPF703E

DDL transaction prevents journaling operation.

CPF7030

Object of type *&3 already being journaled.

CPF7031

Cannot allocate member &3 file &1 in &2.

CPF7033

Start or end journaling failed for member &3.

CPF7034

Logical damage of file &1 in &2.

CPF7035

Object &1 in &2 already known to journal.

CPF705A

Operation failed due to remote journal.

CPF7079

Access path journaling for file &1 not started.

CPF708D

Journal receiver found logically damaged.

CPF7084

Object of type *&6 could not be journaled.

CPF709D

Cannot start journaling object of type *&7.

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9803

Cannot allocate object &2 in library &3.

CPF9812

File &1 in library &2 not found.

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

» CPF9825

Not authorized to device &1.◀

CPF9830

Cannot assign library &1.

» CPF9873

ASP status is preventing access to object.

CPF9875

Resources exceeded on ASP &1.◀

STRJRNOBJ (Start Journal Object) Command Description

STRJRNOBJ Command syntax diagram

Purpose

The Start Journal Object (STRJRNOBJ) command is used to start journaling changes (made to an object or list of objects) to a specific journal. The object types which are supported through this interface are Data Areas (*DTAARA) and Data Queues (*DTAQ). For more information about the possible journal entries which can be sent, see the » Journal management article in the Information Center. ◀

Additionally, the user can specify that only the after image or both the before and the after images of an object of type *DTAARA be journaled. Before images are necessary to remove journaled changes using the Remove Journaled Changes (RMVJRNCHG) command.

After journaling begins for the object, the user should save the journaled object to preserve its journal attribute information. Also, the object must be saved because, for example, journaled changes cannot be applied to a version of the object that was saved before journaling was in effect.

Note: For other ways to start journaling see the following commands:

- Access Paths - Start Journal Access Path (STRJRNAP)

- Integrated File System objects - Start Journal (STRJRN)
- Physical Files - Start Journal Physical File (STRJRNPF)

Restrictions:

1. The object must not be journaling changes to another journal.
2. The maximum number of objects that can be associated with one journal is 250,000. This maximum includes objects whose changes are currently being journaled, objects for which journaling was ended while the current receiver was attached, and journal receivers that are or were associated with the journal while the current journal receiver is attached. If the number of objects is larger than this maximum, journaling does not start.
3. The specified journal cannot be a remote journal.
4. The specified journal and object must reside in the same Auxiliary Storage Pool (ASP).
5. For data areas, only local external data area objects may be journaled. The special data areas (*LDA, *GDA, and *PDA) and DDM data areas cannot be journaled.
6. For data queues, only local data queues are supported. DDM data queues cannot be journaled.

Required Parameters

OBJ Specifies a maximum of 300 qualified object names for which changes are to be journaled. The name of the object 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.

object-name: Specify the name of the object for which changes are to be journaled.

OBJTYPE

Specifies the object type for which changes are to be journaled.

***DTAARA:** Changes for data area objects are to be journaled.

***DTAQ:** Changes for data queue objects are to be journaled.

JRN Specifies the qualified name of the journal that receives the journaled changes.

The name of the journal 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.

journal-name: Specify the name of the journal that receives the journaled changes.

Optional Parameter

IMAGES

Specifies the kinds of images that are written to the journal receiver for changes to the object.

***AFTER**: Only *after* images are generated for changes to the object.

***BOTH**: The system generates both *before* and *after* images for changes to the object.

Note: The value ***BOTH** is only valid for ***DTAARA** objects.

Example for STRJRNOBJ

```
STRJRNOBJ OBJ(DTALIB/MYDTAARA) OBJTYPE(*DTAARA) JRN(MYLIB/JRNLA)
```

This command journals all changes to data area MYDTAARA in library DTALIB to journal JRNLA in library MYLIB. Only the *after* images of updates are written to the journal.

Error messages for STRJRNOBJ

*ESCAPE Messages

CPF6979

Journal &1 in library &2 is unusable.

CPF700A

&1 of &2 objects have started journaling.

CPF705A

Operation failed due to remote journal.

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.

» CPF9825

Not authorized to device &1.«

CPF9830

Cannot assign library &1.

» CPF9873

ASP status is preventing access to object.

CPF9875

Resources exceeded on ASP &1.«

STRJRNPf (Start Journal Physical File) Command Description

STRJRNPf Command syntax diagram

Purpose

The Start Journal Physical File (STRJRNPf) command is used to start journaling changes (made to all members of a specific physical file) to a specific journal. Changes in new members added to the file are also journaled to that journal.

The user can specify that only the after image or both the before and the after images of records in the journaled physical file be journaled. Before images are necessary to remove journaled changes using the Remove Journaled Changes (RMVJRNCHG) command. In addition, the system will automatically capture the before images for a database file if the file is opened under commitment control. For more information about commitment control, see the Commitment control article in the Information Center.

After journaling begins for the file, and after any new members are added to the file, the user should run the Save Changed Object (SAVCHGOBJ) command with OBJTYPE(*FILE) and OBJJRN(*YES) specified. The file must be saved because journaled changes cannot be applied to a version of the file that was saved before journaling was in effect.

When the file being journaled is a distributed file, the STRJRNPf command is also distributed if journaling was successfully started locally. Even if the distribution request fails, the local file remains journaled.

Note:

For other ways to start journaling see the following commands:

- Access Paths - Start Journal Access Path (STRJRNAP)
- Integrated File System objects - Start Journal (STRJRN)
- Other Objects - Start Journal Object (STRJRNOBJ)

Restrictions:

1. The file must not be journaling changes to another journal.
2. Overrides are not applied to files specified on the FILE parameter.
3. The maximum number of objects that can be associated with one journal is 250,000. This maximum includes physical file members whose changes are currently being journaled, members for which journaling was ended while the current receiver was attached, and journal receivers that are or were associated with the journal while the current journal receiver is attached. If the number of objects is larger than this maximum, journaling does not start.
4. This command cannot be used on or with a remote journal.
5. In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA.
6. If the file has Large Object (LOB) columns, and the total size of the LOB columns plus the record size is greater than 15,761,440 bytes, then the file can only be journaled to a journal with RCVSIZOPT(*MAXOPT2) specified.

Required Parameters

FILE Specifies a maximum of 50 qualified names of the physical files for which changes are to be journaled.

The name of the 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.

file-name: Specify the name of the file for which changes are to be journaled.

JRN Specifies the qualified name of the journal that receives the journaled changes.

The name of the journal 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.

journal-name: Specify the name of the journal that receives the journaled changes.

Optional Parameters

IMAGES

Specifies the kinds of record images that are written to the journal receiver for changes to records in the file.

***AFTER:** Only *after* images are generated for changes to records in this file.

***BOTH:** The system generates both *before* and *after* images for changes to records in this file.

OMTJRNE

Specifies the journal entries that are omitted.

***NONE:** No entries are omitted.

***OPNCLO:** Open and close entries are omitted. Open and close operations on the specified file members do not generate open and close journal entries. This prevents the use of TOJOB0 and TOJOB0C entries on the Apply Journal Changes (APYJRNCHG) and Remove Journal Changes (RMVJRNCHG) commands, but it saves some storage space in the attached journal receivers.

Example for STRJRNPF

```
STRJRNPF FILE(MYFILE) JRN(MYLIB/JRNLA)
```

This command journals all changes to all members of file MYFILE (as found using the library search list) to journal JRNLA in library MYLIB. Only the *after* images of updated records are written to the journal.

Error messages for STRJRNPF

*ESCAPE Messages

CPF6971

Damage prevents object &1 from being journaled.

CPF6979
Journal &1 in library &2 is unusable.

CPF700D
File &1 in library &2 not journaled.

CPF7002
File &1 in library &2 not a physical file.

CPF7003
Entry not journaled to journal &1. Reason code &3.

CPF7004
Maximum number of objects journaled to journal &1.

CPF7011
Not enough storage.

CPF703C
DDL transaction prevents journaling operation.

CPF703D
DDL transaction prevents journaling operation.

CPF703E
DDL transaction prevents journaling operation.

CPF7030
Object of type *&3 already being journaled.

CPF7031
Cannot allocate member &3 file &1 in &2.

CPF7033
Start or end journaling failed for member &3.

CPF7034
Logical damage of file &1 in &2.

CPF7035
Object &1 in &2 already known to journal.

CPF704B
Journaling started locally but distributed requests failed.

CPF705A
Operation failed due to remote journal.

CPF708D
Journal receiver found logically damaged.

CPF7084
Object of type *&6 could not be journaled.

CPF709D
Cannot start journaling object of type *&7.

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.

CPF9812

File &1 in library &2 not found.

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

>> CPF9825

Not authorized to device &1.<<

CPF9830

Cannot assign library &1.

>> CPF9873

ASP status is preventing access to object.

CPF9875

Resources exceeded on ASP &1.<<

STRMSF (Start Mail Server Framework) Command Description

STRMSF Command syntax diagram

Purpose

The Start Mail Server Framework (STRMSF) command starts the mail server framework jobs in the system work subsystem (QSYSWRK).

Optional Parameters

MSGOPT

Specifies how the mail server framework processes existing mail server framework messages.

***RESUME:** All existing mail server framework messages continue processing from the point the mail server framework previously ended.

***RESET:** All existing mail server framework messages are processed as if they were just created.

***CLEAR:** All existing mail server framework messages are deleted. This option should only be used when a software error is reported with the mail server framework or its associated exit point programs.

NBRMSFJOB

Specifies the number of mail server framework jobs to start. This option allows concurrent processing of several mail server framework messages.

3: Three jobs are started.

number-of-jobs: Specify the number of jobs you want handling mail server framework messages. The valid values range from 1 through 99.

Examples for STRMSF

Example 1: Starting One Mail Server Framework Job

STRMSF NBRMSFJOB(1)

This command starts one mail server framework job in a normal manner, processing any mail server framework messages at the point at which processing was interrupted.

Example 2: Restarting Mail Server Framework Jobs

```
STRMSF MSGOPT(*RESET)
```

This command starts three mail server framework jobs and any mail server framework messages which were partially handled by previous mail server framework jobs are processed again from the beginning.

Error messages for STRMSF

*ESCAPE Messages

CPFADFAA

STRMSF did not complete successfully.

CPFADFAD

Mail Server Framework currently active.

CPFADFA0

Errors detected on MSF internal message index.

CPFADFA1

Errors detected on MSF internal message queue.

CPFADFFF

Internal system error in program &1.

STRMNTBRM (Start Maintenance for BRM) Command Description

Note: To use this command, you must have the 5722-BR1 (Backup Recovery and Media Services for iSeries) licensed program installed. For detailed information on the parameters of this command, see the online help.

STRMNTBRM Command syntax diagram

Purpose

The Start Maintenance for BRM (STRMNTBRM) command performs various BRMS maintenance and cleanup functions. The STRMNTBRM command can be scheduled to process periodically either by manual submission or inclusion in an automatic job scheduler.

The STRMNTBRM command processes several other commands that can be run separately. These include:

- RMVMEDIBRM - Remove media information from BRM
- MOVMEDEBRM - Move media using BRM
- STREXPBRM - Expire media from BRM
- RMVLOGEBRM - Remove BRM log entries
- WRKMEDIBRM - Display media information
- STRRCYBRM - Recovery analysis

Note:

Depending on the size of your media inventory, this command can take a long time to process.

» Notes:

1. Depending on the size of your media inventory, this command can take a long time to process.

2. The STRMNTBRM command should be run as an independent job. No other BRMS functions should be active when the STRMNTBRM command is processing.
3. Reorganizing the BRMS database files can be a long running process requiring additional disk storage. Reorganization will occur on all BRMS data base files. Reorganization of files may be skipped if sufficient disk storage is not available, or if the file has no members or records.



The STRMNTBRM produces multiple reports depending on the print options you select as well as the current conditions that it detects. For instance, the Media Audit report will not print if you are not using a clustered network environment in BRMS. Listed below are the printer files and associated reports that can be produced when the STRMNTBRM command processes:

- QP1AVER - Version Control
- QP1AEP - Media Expiration
- QP1AMM - Media Report by Expiration Date
- QP1AHS - Media Information
- QP1ALE - Save Strategy Exceptions
- QP1ARCY - Recovering Your Entire System
- QP1AASP - Display ASP Information
- QP1ARW - Recovery Activities
- QP1A2SL - Location Analysis
- QP1ASYN - Media Audit
- QP1A2RCY - Recovery Volume Summary

Note:

This command should not be used by control group *EXIT item processing as results will be unpredictable.

Review the information on daily maintenance in Backup, Recovery and Media Services for iSeries book for additional information on how to automate the use of this command as part of your regular backup processing.



Example for STRMNTBRM

Example 1: Running the BRMS Maintenance Command

```
STRMNTBRM MOVMED(*YES) RMVLOGE(*ARC)
```

In the example media movement will be processed and archive entries that are older than 90 days old will be removed from the BRMS/400 log.

Error messages for STRMNTBRM

None >>

STRMGDSYS (Start Managed System Services) Command Description

Note: To use this command, you must have the 5722-MG1 (Managed System Services for iSeries) licensed program installed.

STRMGDSYS Command syntax diagram

Purpose

The Start Managed System Services (STRMGDSYS) command starts jobs to perform managed system functions such as receiving objects, running programs, running commands, and applying program temporary fixes (PTFs).

Restrictions:

1. The user must have *JOBCTL authority to use the start command.
2. The QSNADS subsystem must be started for Managed System Services to be fully operational.

There are no parameters for this command.

Example for STRMGDSYS

STRMGDSYS

This command starts the managed system functions. Requests received from a central site system can then be processed.

Error messages for STRMGDSYS

***ESCAPE Messages**

CPF90FF

*JOBCTL special authority required to do requested operation.

MSS0059

Internal object not found or damaged.

MSS005B

Storage limit exceeded.

MSS0064

Internal object not found or damaged.

MSS0066

Internal processing error occurred.

MSS0067

Not able to allocate internal object.

MSS0312

Managed System Services/400 already active.

MSS0313

Managed System Services/400 not started.



STRMGRSRV (Start Manager Services) Command Description

Note: To use this command, you must have the 5722-MG1 (Managed System Services for iSeries) licensed program installed.

STRMGRSRV Command syntax diagram

Purpose

The Start Manager Services (STRMGRSRV) command starts the services needed at the central site system.

Restrictions:

1. You must have *JOBCTL authority to use the start command.
2. Public authority for this command is *EXCLUDE.

Optional Parameter

SERVICE

Specifies the service to be started. One or more values may be entered.

***ALL:** Specify to start all the manager services.

***TOPOLOGY:** Specify to gather the topology information from nodes and clients in the network.

***RMTCMD:** Specify to enable the sending of remote commands to managed systems.

Example for STRMGRSRV

```
STRMGRSRV SERVICE(*ALL)
```

This command starts all the central site system services.

Error messages for STRMGRSRV

*ESCAPE Messages

MSS0601

*JOBCTL special authority required for requested operation.

MSS0604

Subsystem QSYSWRK not active.

MSS0730

Error found on &1 command.



STRMGRBRM (Start Migration Using BRM) Command Description

Note: To use this command, you must have the 5722-BR1 (Backup Recovery and Media Services for iSeries) licensed program installed. For detailed information on the parameters of this command, see the online help.

STRMGRBRM Command syntax diagram

Purpose

The Start Migration using BRM (STRMGRBRM) command allows you to specify a migration control group to process. You can start the migration immediately or you can schedule it using the system job scheduler. Processing can be batch or interactive.

The default OPTION parameter is *REPORT, which produces the Migration Item Candidate report. This allows you to process a report and review what will be migrated prior to performing the migration operation. To perform a migration for the items in the Migration Item Candidate report created by the *REPORT option or to restart a migration that was interrupted for some reason, you can specify the special value*RESUME.

Each time you process the STRMGRBRM command using either the *REPORT or *MIGRATE special values, BRMS creates a migration candidate file for the items in the control group. This file contains information about the items in the control group at the time that you processed the STRMGRBRM command, as well as information about ASP sizes and utilization. The migration candidate file that you created is used for producing candidate reports and performing migration operations for the control group.

If you specify *RESUME for a control group, the migration candidate file that was created for the control group when you processed the STRMGRBRM command is used. A new migration candidate file is not created when you specify *RESUME, even though items in the control group or ASP characteristics may have changed.

Restriction: The Advanced Functions feature is required to use this command.

Note: If you use the special value *EXT in a migration control group and the resulting processing modifies the characteristics of either the control group items or ASP characteristics, BRMS does not record the changes in the resulting migration candidate file. You should consider not using the *EXIT special value in migration control groups if the resulting processing alters item or ASP characteristics.

Note: This command should not be used by control group *EXIT item processing as results will be unpredictable.

Example for STRMGRBRM

Example 1: Producing a Report of Migration Candidates

```
STRMGRBRM CTLGRP(MGR01) SCDTIME(2000 0400)
```

In this example migration operations for items identified in control group MGR01 are initiated between the hours of 10:00 PM and 4:00 AM.

Error messages for STRMGRBRM

No error messages. <<

STRMOD (Start Mode) Command Description

STRMOD Command syntax diagram

Purpose

The Start Mode (STRMOD) command starts one or all modes currently in use for an advanced program-to-program communications (APPC) remote location. The user can use STRMOD in either the reset or ended state; it is required only after an End Mode (ENDMOD) command has ended a mode. More information is in the APPC, APPN, and HPR topic in the Information Center.

Restriction: The user must have operational authority for the APPC device to use this command.

Required Parameter

RMTLOCNAME

Specifies the remote location name for which one or more modes is being started. Specify the name of the remote location.

Optional Parameters

DEV Specifies the device description used with the remote location.

***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.

device-name: Specify the name of the device being used.

MODE Specifies the mode that is started.

***NETATR**: The mode name specified in the network attributes is used.

***ALL**: Specifies all modes currently in use by the remote location are being started.

BLANK: The mode name consisting of 8 blank characters is used.

mode-name: Specify the mode name being started.

LCLLOCNAME

Specifies the local location name.

***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.

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

local-location-name: Specify the name of the local location. This name is specified if the user wants to indicate a specific local location name to be associated with the remote location.

RMTNETID

Specifies the remote network ID used with the remote location.

***LOC**: The remote network identifier (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 LCLNETID value specified in the system network attributes is used.

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

remote-network-ID: Specify the name of the remote network ID used.

Example for STRMOD

```
STRMOD RMTLOCNAME(APPCRLOC) DEV(APPCDEV)
MODE(APPCMODE) RMTNETID(CHICAGO)
```

This command starts a mode named APPCMODE for a remote location named APPCRLOC, a device named APPCDEV, and a remote network ID of CHICAGO.

Error messages for STRMOD

*ESCAPE Messages

CPF598B

The &1 command failed for one or more modes.

STRNFSSVR (Start Network File System Server) Command Description

STRNFSSVR Command syntax diagram

Purpose

The Start Network File System Server (STRNFSSVR) command starts one or all of the Network File System (NFS) server daemons. For more information about these daemon jobs, see the OS/400 Network

File System Support  book.

You should use SERVER(*ALL), which will start the daemons in the following order. (This order is the recommended order for starting the Network File System daemons.)

- The Remote Procedure Call (RPC) binder daemon
- The block I/O (BIO) daemon
- The server (SVR) daemon
- The mount (MNT) daemon
- The network status monitor (NSM) daemon
- The network lock manager (NLM) daemon

If you are choosing to start just one daemon, be sure you understand the appropriate order for starting NFS daemons and the possible consequences of starting daemons in an order other than that specified

above. For more information about starting NFS daemons, see OS/400 Network File System Support  book.

If you attempt to start a daemon or daemons that are already running, they will not cause the command to fail, and it will continue to start other daemons you have requested to start. The command will issue diagnostic message CPDA1BA if the daemon is already running. For best results, end NFS daemons before attempting the STRNFSSVR command.

To determine if an NFS daemon is running, use the Work with Active Jobs (WRKACTJOB) command and look in the subsystem QSYSWRK for existence of the following jobs:

QNFSRPCD	The RPC binder daemon
QNFSBIOD	The block I/O (BIO) daemon
QNFSNFSD	The NFS server (SVR) daemon
QNFSMNTD	The mount (MNT) daemon
QNFSNSMD	The network status monitor (NSM) daemon
QNFSNLM	The network lock manager (NLM) daemon

Restrictions:

1. You must have *IOSYSCFG special authority to use this command.
2. The user must be enrolled in the system distribution directory. Use the Add Directory Entry (ADDIRE) command to enroll the user.

Required Parameter

SERVER

Specifies the Network File System (NFS) daemon jobs to be started by this command. The specified daemon should not already be running.

***ALL:** All NFS daemons are started.

***RPC:** The NFS RPC binder daemon are started.

***BIO:** Starts NFS block I/O daemons. Additional daemons are started if the number you specify on the NBRBIO parameter is greater than the number of block I/O daemons already running on the system.

***SVR:** Starts NFS server daemons. Additional daemons are started if the number you specify on the NBRSVR parameter is greater than the number of server daemons already running on the system.

***MNT:** The NFS mount daemon is started.

***NSM:** The NFS network status monitor daemon is started.

***NLM:** The NFS network lock manager daemon is started.

Optional Parameters

STRJOBTIMO

Specifies the number of seconds to wait for each daemon to successfully start. If a daemon has not started within the timeout value, the command will fail.

30: Default seconds before timeout.

***NOMAX:** Wait forever for daemons to start; do not timeout.

timeout-value: Specify the number of seconds to wait for daemons to start before timing out and failing the command. Valid values range from 1 to 3600 seconds.

NBRSVR

Specifies the number of NFS server (*SVR) daemon jobs you want to have running. Additional daemons are started if the number you specify on this parameter is greater than the number of server daemons already running on the system. This parameter can only be used if SERVER(*SVR) is specified.

1: One NFS server daemon job should be started if there are not already any NFS server daemons running.

number-server-daemons: Specify the number of NFS server daemon jobs you want to have running. Valid values range from 1 to 20.

NBRBIO

Specifies the number of NFS block I/O (*BIO) daemon jobs you want to have running. Additional daemons are started if the number you specify on this parameter is greater than the number of block I/O daemons already running on the system. This parameter can only be used if SERVER(*BIO) is specified.

1: One NFS block I/O daemon job should be started if there are not already any NFS block I/O daemons running.

number-block-I/O-daemons: Specify the number of NFS block I/O daemon jobs you want to have running. Valid values range from 1 to 20.

RTVRPCREG

Specifies whether to retrieve previously recorded registration information when the RPC daemon is started.

If registration information is retrieved, servers do not have to re-register with the RPC server. This parameter can only be used if SERVER(*RPC) is specified.

***NO:** Do not retrieve registration information.

***YES:** Retrieve registration information.

Examples for STRNFSSVR

Example 1: Start All NFS Daemons

```
STRNFSSVR SERVER(*ALL) STRJOBTIMO(*NOMAX)
```

This command starts all NFS daemons, and waits forever for them to start. No daemons should be previously running.

Example 2: Start Only One Daemon

```
STRNFSSVR SERVER(*MNT)
```

This command starts the NFS mount daemon, and waits up to the default of 30 seconds for it to start. The mount daemon should not be already running, and other daemons have been started in the appropriate order.

Error messages for STRNFSSVR

None

STRNWSCSL (Start Network Server Console) Command Description

STRNWSCSL Command syntax diagram

Purpose

The Start Network Server Console (STRNWSCSL) command is used to start a console session on a network server with a type of *AIX. Upon entering this command, the server console output will be displayed on the user's display. The user can also respond to prompts and enter commands. To end the console session, press the attention (ATTN) key and select option 99 on the menu that appears.

Restrictions:

- The user of this command must have IOSYSCFG authority.
- The network server must have *AIX specified for the TYPE parameter and must not be varied off.

Required Parameter

NWSD

Specifies the name of the network server description.

network-server-description-name: Specifies the network server name.

Optional Parameters

CCSID

Specifies the ASCII coded character set identifier (CCSID) in use on the server. The ASCII CCSID value is used to map between the EBCDIC CCSID of the user's job and the console output from the server.

***NWS**: The CCSID of the primary language installed on the network server is used.

CCSID-value: The requested CCSID value is used. This value is validated to ensure that a valid ASCII CCSID is requested. Valid values range from 2 through 65533.

DSPCHRATTR

Specifies whether or not character attributes are displayed. Character attributes (highlighting and reverse image for example) take up a position on 5250 displays. This means that the user may observe a loss of data when a display contains screen attributes. The user may prefer to specify *NO in this case in order to see all screen data without character attributes. This parameter is not applicable when using a display that supports extended attributes (such as a 3477). Character attributes can be displayed without data loss on such displays.

***YES**: Character attributes are displayed.

***NO**: Character attributes are not displayed.

PAGESCROLL

Specifies whether or not page scrolling is in effect for the console session. Page scrolling allows the user to control the pace of the output from AIX.

***NO**: Page scrolling is not in effect. AIX output will be displayed as fast as it is received.

***YES**: Page scrolling is in effect. AIX output will pause after each screen of data that is received from AIX. The user presses the ROLL DOWN key to continue with AIX console output.

Examples for STRNWSCSL

Example 1: Start console session

```
STRNWSCSL NWSD(AIXIOP)
```

The above command starts a console session with the network server named AIXIOP. All of the default values for the console session are used.

Example 2: Start session with no character attributes

```
STRNWCSL NWS(AIXIOP) DSPCHRATTR(*NO)
```

The above command starts a console session with the network server named AIXIOP. Character attributes will not be displayed on the screen.

STROBJCVN (Start Object Conversion) Command Description

STROBJCVN Command syntax diagram

Purpose

The Start Object Conversion (STROBJCVN) command converts user objects from the format used in a previous version, release, and modification level of an IBM-supported operating system to the format required for use in the current operating system. The following iSeries 400 objects are converted with this command:

1. Programs (original program model (OPM) and Integrated Language Environment (ILE))
2. Service programs
3. Modules
4. Database files (physical and logical)

The user objects that are not converted with this command are automatically converted when they are first specified for use.

Note: Using objects that are not yet converted can increase the response time of an operation.

Restrictions:

1. System libraries cannot be specified for object conversion.
2. Programs, service programs, and modules must have all observable creation data to be converted. Unobservable creation data cannot be used by this command.

Required Parameter

LIB Specifies the user library whose objects are to be converted to the format of the current operating system.

» ***ALLUSR:** User libraries are all libraries with names that do not begin with the letter Q except for the following:«

#CGULIB	#DSULIB	#SEULIB
#COBLIB	#RPGLIB	
#DFULIB	#SDALIB	

» Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:«

QDSNX	» QSYS2xxxx«	QUSROND
QGPL	QS36F	QUSRPOSGS

QGPL38	QUSER38	QUSRPOSSA
QMPGDATA	QUSRADSM	QUSRPYMSVR
QMQMDATA	QUSRBRM	QUSRRDARS
QMQMPROC	QUSRDIRCL	QUSRSYS
QPFRDATA	QUSRDIRDB	QUSRVI
QRCL	QUSRIJS	QUSRVxRxMx
» QRCLxxxx«	QUSRINFSKR	
» QSYS2«	QUSRNOTES	

Notes:

1. » 'xxxx' is the number of a primary auxiliary storage pool.«
2. A different library name, of the form QUSRVxRxMx, can be created by the user for each release that IBM supports. VxRxMx is the version, release, and modification level of the library.

***PTY:** Refer to the PTY parameter to determine the libraries to be converted.

library-name: Specifies the name of the user library whose objects are converted.

Note: Although only one library can be specified on this command, repeated calls of this command in a CL program can be used to convert a set of user libraries.

OBJTYPE

Specifies the type of objects to be converted to the format of the current operating system.

***ALL:** All *PGM, *SRVPGM, *MODULE, and database files in the specified library will be converted to the format of the current operating system.

***ALLPGM:** Only *PGM and *SRVPGM objects in the specified library will be converted to the format of the current operating system.

***FILE:** Only database file member objects in the specified library will be converted to the format of the current operating system.

PTY Specifies the priority of libraries to be converted to the format of the current operating system. The priority groups are set on the source system during upgrade preparation. This parameter is valid only when LIB(*PTY) is specified.

***ALL:** All user libraries will be converted to the format of the current operating system. Specifying this with LIB(*PTY) is the the same as specifying LIB(*ALLUSR).

priority-number: The libraries with the specified priority will have objects converted to the format of the current operating system.

Example for STROBJCVN

```
STROBJCVN LIB(LIB1)
```

This command converts all OPM and ILE programs, service programs, modules, and physical and logical database files contained in the LIB1 user library to the format of the current operating system.

Error messages for STROBJCVN

***ESCAPE Messages**

CPDA972

Errors occurred while converting object &1 in library &2 type *&3.

CPFA96D

No objects converted in library &1.

CPFA960

Library &1 is not a user library.

CPFA972

Not all eligible objects in library &1 converted.

CPFB0C1

Not all eligible objects in priority &1 converted.

CPFB0C2

No objects converted in priority &1.

CPF9810

Library &1 not found.

STRPASTHR (Start Pass-Through) Command Description

STRPASTHR Command syntax diagram

Purpose

The Start Pass-Through (STRPASTHR) command allows pass-through (and sign-on) to a remote system as if attachment is local. This command uses the iSeries 400 advanced program-to-program communications (APPC) and advanced peer-to-peer networking (APPN) support to connect the local display station to the remote system. More information on configuring or operating the pass-through

operation is in the Remote Work Station Support  book.

Restriction:

1. This command cannot be entered at a work station with a display that has 12 lines by 80 characters.

Required Parameter

RMTLOCNAME

Specifies, if CNNDEV(*LOC) is specified, the name of the remote location that is the target of the pass-through session, or if devices are specified on the CNNDEV parameter, it specifies the first system that does intermediate pass-through routing.

***CNNDEV:** The APPC devices specified on the CNNDEV parameter are used.

remote-location-name: Specify the name of the remote location that is the target of the pass-through session or the name of the first system that does intermediate pass-through routing. The advanced peer-to-peer networking (APPN) support function determines the route to this location.

Optional Parameters

CNNDEV

Specifies the names of the device descriptions that connect the first system that does pass-through routing from the source system to the target system. If RMTLOCNAME(*CNNDEV) is specified, the first device specified on this parameter is on the source system. If the RMTLOCNAME parameter is a location, the first device specified is on the system containing that location. If the target system is more than one system away and pass-through must establish the intermediate sessions, a list of APPC device descriptions must be specified. The APPC device names must be listed in the order that the systems are passed through to reach the target system.

***LOC:** The RMTLOCNAME parameter is used to identify the target of the pass-through session and the path used to reach the target.

APPC-device-name(s): Specify the names of the device descriptions that complete the route from the source system to the target system. Up to 16 names can be specified.

VRTCTL

Specifies the name of the virtual controller on the remote system that is used to do a pass-through session. If a virtual controller is specified, one of the virtual display devices attached to it is selected to do the pass-through session. The system tries to compare the device type and model of the physical display device with an available virtual device. If the same type or a similar type is not available, a comparison is made with a 5251 Model 11, if available. Graphics cannot be done with a 5251 Model 11 device; a 5292 Model 2 device is required. If no virtual devices are available, another virtual controller must be specified, or the user must try again later.

***NONE**: No controller is specified. VRTDEV(*NONE) requests that the target system automatically configures a virtual device.

virtual-controller-name: Specify the name of the virtual controller description on the remote system.

VRTDEV

Specifies one or more devices on the remote system that are connected to a virtual controller used for the pass-through session. A device from the list on the remote system is selected based on a comparison of device type and model. If more than one device is in the list, the first available device that closely matches the type and model of the device on which the command is entered is used. If the same type or a similar type is not available, a comparison is made with a 5251 Model 11, if available. Graphics cannot be done with a 5251 Model 11 device; a 5292 Model 2 device is required.

***NONE**: No device names are specified. VRTCTL(*NONE) requests that the target system automatically configures a virtual device.

virtual-display-device-name: Specify the names of the virtual display device descriptions on the remote system used for the pass-through session. Up to 32 names can be specified.

MODE Specifies the mode name used for the pass-through session. This parameter is not valid if RMTLOCNAME(*CNNDEV) is specified. In this case, the system uses the first mode name specified in the APPC device description.

***NETATR**: The pass-through mode in the network attributes is used for the pass-through session.

mode-name: Specify a mode name to use for the pass-through session. Specify BLANK for a mode name consisting of eight blank characters.

LCLLOCNAME

Specifies the local location name.

***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.

***NETATR**: The default location name defined in the Change Network Attributes (CHGNETA) command is used.

local-location-name: Specify the local location name for the source system.

RMTNETID

Specifies the network identifier (ID) of the network in which the remote location resides. This parameter is not valid if RMTLOCNAME(*CNNDEV) is specified.

***LOC**: The remote network identifier (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 identifier (ID) is used.

remote-network-ID: Specify the identifier of the network in which the target system (specified by the RMTLOCNAME parameter) is located.

SRQ10PGM

Specifies that the System Request menu or a user-written program starts when SYSREQ option 10 is selected. The user program displays a menu that allows selection of the system to access, and then transfers to a group job that sends the STRPASTHR command to the desired system.

More information is in the Remote Work Station Support  book.

***SRQMNU**: The System Request menu is displayed.

The name of the program 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.

program-name: Specify the name of the program that starts when the SYSREQ option 10 is selected.

RMTUSER

Specifies the user profile for automatic sign-on to the target system. If a profile is specified for this parameter and password security is active on the target system, RMTPWD(*NONE) is not valid.

***NONE**: No user profile name is sent, and automatic sign-on does not occur.

***CURRENT**: The user profile that is currently running is used.

user-profile-name: Specify a user profile name to use that exists on the target system. If the target system allows it, and the user profile exists on the target system, the user is automatically signed on. Otherwise, the user is presented with a sign-on display on the target system or a failure message on the source system, depending upon the configuration of the target system. If a profile is specified and password security is active on the target system, a password must be specified, even if the profile specified is the same as the current profile.

RMTPWD

Specifies the password sent to the target system.

***NONE**: The system does not pass a password. If a profile is specified on the RMTUSER parameter and password security is active on the target system, this value is not allowed.

password: Specify a password sent to the target system to verify the sign-on of the user specified in the RMTUSER parameter. This password is encrypted before being sent across the communication line.

RMTINLPGM

Specifies the program called immediately after sign-on to the system.

***RMTUSRPRF**: The initial program specified in the remote user profile is run immediately after the job (which was automatically signed-on) starts.

***NONE**: No program is run before the first menu is displayed, even if the first program is specified in the remote user profile.

program-name: Specify the name of a program that is run immediately after automatic sign-on.

RMTINLMNU

Specifies the first menu shown when automatically signed on the target system after the first program is run.

***RMTUSRPRF:** The initial menu specified in the remote user profile is shown immediately after the first program is run.

***SIGNOFF:** No menu is displayed after the first program is run, even if an initial menu is specified in the remote user profile. After the program runs, the user is signed off, and the program runs.

menu-name: Specify the menu shown immediately after the initial program is run.

RMTCURLIB

Specifies the name of the library that becomes the current library in the library list of the job after automatic sign-on to the system.

***RMTUSRPRF:** The current library specified in the remote user profile becomes the current library in the library list after automatic sign-on.

library-name: Specify the name of the library that becomes the current library in the library list before the pass-through session is established.

PASTHRSCN

Specifies whether the pass-through display and associated status messages appear before the pass-through session is established.

***YES:** The pass-through display and informal messages are shown before the pass-through session is established.

***NO:** The pass-through display and information messages are not shown before the pass-through session is established.

Examples for STRPASTHR

Example 1: Pass-Through to Toronto

```
STRPASTHR RMTLOCNAME(*CNNDEV) CNNDEV(DET CHI TOR)
          VRTCTL(VWSC)
```

This command specifies starting a pass-through to the Toronto system by going through Detroit and Chicago. More information is in the Remote Work Station Support  book.

Example 2: Pass-Through to Detroit

```
STRPASTHR RMTLOCNAME(DETROIT) VRTCTL(VWSC)
```

This command specifies a pass-through to the Detroit system. APPN establishes the route to Detroit.

Example 3: Pass-Through to Toronto

```
STRPASTHR RMTLOCNAME(DETROIT) CNNDEV(CHI TOR)
          VRTCTL(VWSC)
```

This command specifies another way to pass-through to the Toronto system by going through Chicago and Detroit. APPN establishes the route to Detroit.

Example 4: Pass-Through to Detroit

```
STRPASTHR RMTLOCNAME(DETROIT) RMTUSER(*CURRENT)
```

This command specifies a pass-through to the DETROIT system and an automatic sign-on using the user profile with the same name as the one currently used on the source system. It also specifies that the DETROIT system automatically configures a virtual device for the pass-through session, since a virtual controller or virtual device was not specified.

Error messages for STRPASTHR

***ESCAPE Messages**

CPF2702

Device description &1 not found.

CPF2703

Controller description &1 not found.

CPF5383

Mode &7 specified for device &4 not valid.

CPF5536

System cannot automatically select an APPC device description for the remote location.

CPF5546

Class-of-service for device &4 not valid.

CPF8901

Virtual device &1 not varied on.

CPF8902

Virtual device &1 not available.

CPF8903

Device &1 not valid for pass-through.

CPF8904

Pass-through request not accepted.

CPF8905

Pass-through not allowed on this system.

CPF8906

Error during session initialization. Reason code &1.

CPF8907

Communications failure for device &1.

CPF8908

Controller &1 not varied on.

CPF8909

Old software release. Pass-through ended.

CPF8910

Controller &1 not valid for pass-through.

CPF8911

Communications failure. Session was not started.

CPF8912

Pass-through session ended. Reason code &1.

CPF8913

Pass-through ended abnormally.

CPF8916

Cannot select virtual device &1 at system &2.

- CPF8917**
Not authorized to &1.
- CPF8918**
Job canceled at system &1.
- CPF8919**
Device &1 not accessed by system &2.
- CPF8920**
Pass-through failed. &1 must be varied off and on.
- CPF8921**
APPC failure. Failure code is &3.
- CPF8922**
Negative response from device &1 at system &2.
- CPF8923**
Data stream received at system &1 not valid.
- CPF8924**
No available virtual controller.
- CPF8925**
Device &1 not created.
- CPF8928**
Device &1 could not be changed.
- CPF8929**
Device &1 could not be varied on.
- CPF8931**
Location &1 not an APPC location.
- CPF8932**
Device &1 must be non-networking APPC device.
- CPF8933**
Route to specified location not found.
- CPF8935**
Pass-through not allowed to system &1.
- CPF8936**
Pass-through failed for security reasons.
- CPF8937**
Automatic sign on not allowed.
- CPF8938**
Error in QRMTSIGN program. Pass-through failed.
- CPF8939**
Trying to send too much data.
- CPF8940**
Cannot automatically select virtual device.
- CPF8941**
Incorrect internal use of pass-through.
- CPF8943**
Pass-through not allowed from server TELNET session.

***STATUS Messages**

CPI8901

No matching device on remote system. Function limited.

CPI8902

Pass-through started at system &1.

CPI8903

Virtual device &1 selected at system &2.

CPI8906

Automatic sign-on not allowed.

STRPEX (Start Performance Explorer) Command Description

STRPEX Command syntax diagram

Purpose

The Start Performance Explorer (STRPEX) command starts a new performance explorer session or resumes a suspended performance explorer session.

» 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 Operating System/400 through iSeries Navigator's Application Administration support. The Change Function Usage Information (QSYCHFUI) API, 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. The following user profiles have private authorities to use the command:
 - QPGMR
 - QSRV



Required Parameters

SSNID

Specifies an identifier for this performance explorer session. This name must be unique within the active sessions of the performance explorer tool.

session-identifier: Specify the session identifier for a new or suspended performance explorer session.

OPTION

Specifies whether a new session should be started or a session which is currently suspended should be resumed.

***NEW:** A new session of the performance explorer tool is started. Performance data collection begins as soon as the session has been established.

***INZONLY:** A new performance explorer session is started, but once the session is established, it is suspended. This option allows the user to perform the setup for a particular session before the scenario to be monitored is started. To begin data collection, the user must invoke this command again specifying the same session identifier and OPTION(*RESUME).

***RESUME:** A suspended performance explorer session is resumed. The session was suspended either by specifying OPTION(*SUSPEND) on the ENDPEX command or by specifying OPTION(*INZONLY) on a previous STRPEX command.

Optional Parameters

DFN Specifies the name of the performance explorer definition to use when starting a new performance explorer session. The definition identifies the performance data to be collected. This parameter is required in order to start a new performance explorer session. This parameter cannot be specified when resuming a suspended session, since the suspended session already has an associated performance explorer definition.

» ***SELECT:** A list of existing definitions will be displayed for user selection. This value is valid only if the command is run in an interactive job. «

definition-name: Specify the name of the performance explorer definition.

FTR » Specifies the name of the performance explorer filter to use when starting a new performance explorer session. The filter determines which events are collected based on the filter values which are compared to the event's actual data. If a filter is not specified, then all events in the definition are collected. This parameter cannot be specified when resuming a suspended session.

***SELECT:** A list of existing filters will be displayed for user selection. This value is valid only if the command is run in an interactive job.

filter-name: Specify the name of the performance explorer filter. «

Examples for STRPEX

Example 1: Start a New Session »

```
STRPEX SSNID(TESTRUN2) DFN(NEWDESC) OPTION(*NEW)
FTR(MYFILTER)
```

This command starts a new session of the performance explorer using the criteria identified in a definition named NEWDESC and a filter named MYFILTER. The new session name is TESTRUN2. «

Example 2: Resume a Suspended Session

```
STRPEX SSNID(TESTRUN1) OPTION(*RESUME)
```

This is a resume operation for an already existing session of the performance explorer named TESTRUN1.

No error messages.

STRPFRG (Start Performance Graphics) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

STRPFRG Command syntax diagram

Purpose

The Start Performance Graphics (STRPFRG) command calls up the performance graphics menu giving the user access to all available performance graphics. This command is valid only in an interactive environment.

Optional Parameters

FMTLIB

Specifies the library in which the graph formats and graph packages are kept. The graph formats and graph packages are used to create performance and historical data graphs.

QPFRDATA: The graph formats and graph packages are located in the IBM-supplied performance data library, QPFRDATA.

library-name: Specify the name of the library where the graph formats and graph packages are located.

PFRLIB

Specifies the library in which the database files are stored. The database files contain the performance and historical information used on the graphs.

QPFRDATA: The database files are located in the IBM-supplied performance data library, QPFRDATA.

library-name: Specify the name of the library where the database files are located.

JOB Specifies the job name used for submitting a display job for batch processing.

Note: If *NONE is specified on the JOBD parameter, this parameter is ignored; job processing is performed interactively.

***CMD:** The command name that corresponds to the selected menu option is used. Jobs from the same option have the same name.

***MBR:** The name specified for the performance data member on the MBR parameter of the corresponding command is used.

job-name: Specify the name used for batch jobs.

JOBD Specifies the job description used to submit jobs for batch processing.

The name of the job description 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.

QPFRJOB: The IBM-supplied Performance Tools job description is used.

job-description-name: Specify the name of an alternate job description.

Other Single Values

***NONE:** A batch job is not submitted; processing continues interactively while the user waits. The user's work station cannot be used during this time, which may be significant for long jobs.

Example for STRPFRG

STRPFRG

This command displays the menu interface for Performance Tools graphics. Library QPFRDATA is used for storing and retrieving graph formats, graph packages, and performance data for this session. Any requests that are submitted to run in batch use job description QPFRJOB found in the library list.

Error messages for STRPFRG

***ESCAPE Messages**

PFR7023

Interval data cannot be displayed for IDLC lines.

PFR9026

&1 cannot be changed or deleted.

PFR9027

Option chosen for graph format &1 not valid.

PFR9028

Option chosen for graph package &1 not valid.

PFR9032

Too many formats selected.

PFR9033

Too many items selected to be deleted.

PFR9039

Historical data cannot be created for member &2.

PFR9040

Specify *AUTO for range with *TIME for variable.

PFR9044

Graph format selected for historical graph not valid.

PFR9045

*OUTFILE not valid for graph package.

PFR9095

&2 cannot be changed or deleted.

STRPFRT (Start Performance Tools) Command Description

Note: To use this command, you must have the 5722-PT1 (Performance Tools for iSeries) licensed program installed.

STRPFRT Command syntax diagram

Purpose

The Start Performance Tools (STRPFRT) command calls the Performance Tools menu interface. This shows the Performance Tools main menu (PERFORM), which provides access to all of the performance tools.

Optional Parameters

LIB Specifies the library where the performance data is located.

QPFRDATA: The performance data files are located in the IBM-supplied performance data library, QPFRDATA.

library-name: Specify the name of the library where the performance database files are located.

JOB Specifies the job name used for submitting report and utility jobs for batch processing.

Note: If *NONE is specified on the JOBD parameter, this parameter is ignored; job processing is performed interactively.

***CMD:** The command name that corresponds to the selected menu option is used. Jobs started from the same option have the same name.

***MBR:** The name specified for the performance data member on the MBR parameter of the corresponding command is used. If the command does not have a MBR parameter and *MBR is specified for this parameter, the command name is used.

job-name: Specify the name used for batch jobs.

JOB Specifies the job description used to submit jobs for batch processing.

The possible library values are:

- ***LIBL:** All libraries in the job's library list are searched until the first match is found.
- ***CURLIB:** The current library is used to locate the job description. 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 where the job description is located.

QPFRJOB: The IBM-supplied Performance Tools job description is used.

job-description-name: Specify the name of an alternate job description.

***NONE:** A batch job is not submitted; processing continues interactively while the user waits. The user's work station is not available for other use during this time, which could be significant for long jobs.

Example for STRPFRT

STRPFRT

This command calls the menu interface to Performance Tools. The library, QPFRDATA, is used in storing and getting the performance data for this session. Requests that are submitted to run in batch use the job description QPFRJOB in the library list.

Error messages for STRPFRT

*ESCAPE Messages

None.

STRPFRTRC (Start Performance Trace) Command Description

STRPFRTRC Command syntax diagram

Purpose

The Start Performance Trace (STRPFRTRC) command is a simplified interface to the TRCINT command for collecting Multiprogramming level (MPL) and Transaction trace data. This command collects the same performance trace data as was collected in previous releases by the Start Performance Monitor (STRPFRMON) command.

The trace started by this command creates and uses trace table QPM_STRPFRTRC. If the trace table exists, any existing data will be deleted before this trace begins.

The trace can be stopped and the data can be written to a data base file by using the End Performance Trace (ENDPFRTRC) command.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.

2. The following user profiles have private authorities to use the command:

- QSRV

SIZE Specifies the size of the trace table.

Note:

The storage indicated on this parameter is immediately allocated from the system auxiliary storage pool (ASP 1). Refer to the Trace Internal (TRCINT) command SIZE parameter for additional information regarding the setting of trace table size.

Specifying a size of less than 16 megabytes is not recommended.

» **Element 1: Size** «

***CALC:** The minimum trace table size is determined based on the processor group of your system.

***MAX:** The trace table is set to the maximum size of » 258048 megabytes «.

table-size: Specify the size of the trace table in » kilobytes or megabytes «. Valid values range from » 1 « through 998000.

» **Element 2: Unit of Measure**

Specify whether the table-size value specified for the Size element should be treated as number of kilobytes or number of megabytes.

***KB:** The trace table size is specified in kilobytes. The valid range is 128 through 998000.

***MB:** The trace table size is specified in megabytes. The valid range is 1 through 258048. «

OMTTRCPNT

Specifies trace points whose trace records are to be excluded.

***NONE:** No trace points are to be excluded.

***RSCMGT:** Resource management trace points (seize/lock conflict data) will be excluded from the trace.

JOBTYPE

Specifies the types of jobs for which trace data is being collected for use in the batch job trace report. A maximum of 11 of the following job types can be traced, or one of the single values *NONE or *ALL can be specified.

Note:

The value *DFT includes the values *ASJ, *BCH, *EVK, *MRT, *PDJ, *PJ and *BCI. The value *BCH includes the values *EVK, *MRT, *PDJ, *PJ, and *BCI.

***DFT:** Batch and autostart job types are traced.

- ***ASJ**: Autostart job types are traced.
- ***BCH**: Batch job types are traced.
- ***EVK**: Jobs started (evoked) by a procedure start request are traced.
- ***INT**: Interactive jobs are traced.
- ***MRT**: Multiple requester terminal job types are traced.
- ***RDR**: Reader job types are traced.
- ***SBS**: Subsystem monitor job types are traced.
- ***SYS**: System job types are traced.
- ***WTR**: Writer job types are traced.
- ***PDJ**: Print driver job types are traced.
- ***PJ**: Prestart job types are traced.
- ***BCI**: Batch Immediate job types are traced.
- ***NONE**: No job types are traced.
- ***ALL**: All of the job types listed above are traced.

JOBTRCIV

Specifies the time (in CPU seconds) between each collection of the job trace data.

0.5: A time slice interval value of 0.5 CPU seconds is used.

number-of-seconds: Specify a time slice interval value ranging from 0.1 through 9.9 CPU seconds.

Example for STRPFRTRC

Example 1: Starting Performance Trace

```
STRPFRTRC SIZE(*CALC)
```

In this example, the trace table size may be adjusted to the calculated minimum and performance trace data is collected. This example will result in the same trace table size and data as would STRPFRMON TRACE(*ALL) DMPTRC(*NO).

Error messages for STRPFRTRC

***ESCAPE Messages**

CPF0A2A

Performance trace already started

STRTCPPTP (Start Point-to-Point TCP/IP) Command Description

STRTCPPTP Command syntax diagram

Purpose

The Start Point-to-Point TCP/IP (STRTCPPTP) command is used to start a point-to-point TCP/IP session job. A session job operates in one of two possible modes. Answer mode sessions (*ANS) allow a remote system to contact this iSeries 400 and establish a point-to-point TCP/IP session. Dial mode sessions (*DIAL) are used to have this iSeries 400 contact a remote system and establish a point-to-point TCP/IP session.

The TCP/IP point-to-point session jobs run in the QSYSWRK subsystem.

Required Parameter

CFGPRF

Specifies the name of the point-to-point configuration profile to use. The profile specifies all the attributes and values necessary to define this session. The mode specified in the profile determines whether this iSeries 400 will be contacting a remote system (*DIAL session), or if a remote system will be contacting this iSeries 400 (*ANS session). All other information about the characteristics of the session is also contained in the point-to-point configuration profile.

configuration-profile-name: Specify the name of a valid, inactive point-to-point configuration profile.

Optional Parameters

RESTART

Specifies whether to restart the point-to-point TCP/IP session job running the profile defined by the CFGPRF parameter. This parameter is only valid when used for a running point-to-point profile. If RESTART(*YES) is used with a point-to-point profile that is not running then an error will be posted.

***NO:** Do not restart the TCP/IP point-to-point session job.

***YES:** Restart the TCP/IP point-to-point session job. How the TCP/IP point-to-point session job is restarted depends on the type of point-to-point session that is running. By type, the session will be restarted as follows:

- Switched Line Dial (Dial remote systems)
 - Reset the session to start from Re-Dialing the remote system.
- Switched Line Answer (Answering incoming calls)
 - Reset the session to Ring Wait state (wait for remote system to dial)
- Leased Line (Initiator or Terminator) for PPP
 - Reset back to LCP Configure Request
- Leased Line (Initiator or Terminator) for SLIP
 - Reset back to an 'Active' state
- Dial-on-Demand
 - Reset the session to wait for the next Dial-on-Demand request

The OUTPUT, SNDINQMSG and AUTODLTCFG parameters will be ignored when RESTART(*YES) is specified.

OUTPUT

Specifies whether or not to print a copy of the script dialog between this iSeries 400 and the remote system. You can use this dialog to diagnose problems that prevent a point-to-point TCP/IP session from being successfully established.

***ERROR:** Only print the script dialog if one or more errors occur while establishing the TCP/IP point-to-point session job.

***NONE:** Do not print the script dialog.

***PRINT:** Print the script dialog with the spooled output of the job that issued the STRTCPPTP command, regardless of whether or not any errors occurred.

SNDINQMSG

Specifies whether or not to send an inquiry message after any initialization has been completed. The inquiry message will be sent to the QTCP message queue.

Until you respond to the inquiry message, the point-to-point session job is held. While the job is held, you can set up trace tools for diagnosing problems related to this TCP/IP point-to-point session job.

***NO:** Do not send an inquiry message after initialization has been completed.

***YES:** The point-to-point session job will send an inquiry message after it has completed its initialization. If the configuration profile has the attribute for automatic creation of the controller and device, the inquiry message will be sent after the controller description and device description have been created.

AUTODLTCFG

Specifies whether or not any controller description and device description configuration objects that were automatically created during STRTCTPTP initialization should be automatically deleted when the point-to-point session job ends.

***NO:** Do not delete any controller description and device description configuration objects that were automatically created by STRTCPPTP. This allows the controller description and device description to be reused the next time a TCP/IP point-to-point session job is started using the specified configuration profile.

***YES:** Delete any controller description and device description configuration objects that were automatically created during STRTCPPTP initialization.

Examples for STRTCPPTP

Example 1: Start a point-to-point TCP/IP session job.

```
STRTCPPTP CFGPRF(DIALPRF)
```

This command starts a point-to-point TCP/IP session job. The point-to-point configuration profile "DIALPRF" contains the information that will determine whether this iSeries 400 will be contacting a remote system (*DIAL session), or if a remote system will be contacting this iSeries 400 (*ANS session).

Example 2: Start a session and capture the script dialog.

```
STRTCPPTP CFGPRF(DIALPRF) OUTPUT(*PRINT)
```

This command starts a point-to-point TCP/IP session job. The point-to-point configuration profile "DIALPRF" contains the information that the iSeries 400 will use to contact the remote system and establish a point-to-point TCP/IP session with it. A copy of the dialog exchanged between the two systems prior to establishing a point-to-point TCP/IP session is written to the default output queue for the job that issued the STRTCPPTP command.

Example 3: Send an inquiry message after initializing the TCP/IP session job.

```
STRTCPPTP CFGPRF(ANSWERPRF) SNDINQMSG(*YES)
```

This example uses the SNDINQMSG parameter. This parameter is normally used only if it is necessary to collect detailed internal trace information when the controller and device description used by the point-to-point session job are automatically created.

This command starts a point-to-point TCP/IP session job. The point-to-point configuration profile "ANSWERPRF" contains the information that will be used to establish a point-to-point TCP/IP session when a remote system contacts this iSeries 400.

After automatically creating the controller and device description it will use, the point-to-point session job will send an inquiry message to the QTCP message queue. Depending on the response, the job will either continue running or will be cancelled.

Example 4: Restarting a running point-to-point profile.

```
STRTCPPTP CFGPRF(ANSWERPRF) RESTART(*YES)
```

This example will restart the running point-to-point profile ANSWERPRF. How the session is reset (restarted) depends on the connection type of the line (switched or unswitched) and the mode (dial or answer). For more information on how the session would be reset, see the details for RESTART(*YES) above.

Error messages for STRTCPPTP

***ESCAPE Messages**

TCP1A1F

Cannot process request while &3/&2/&1 using &6.

TCP8205

Required object &2/&1 type *&3 not found.

TCP8207

STRTCPPTP &1 not performed. Job &6/&5/&4 using &1.

TCP8208

STRTCPPTP &1 not performed. See previous messages.

STRPJ (Start Prestart Jobs) Command Description

STRPJ Command syntax diagram

Purpose

The Start Prestart Jobs (STRPJ) command starts jobs for a prestart job entry in an active subsystem when there are no currently active prestart jobs for the prestart job entry. This command is valid after an ENDPJ command is complete or when all prestart jobs have been ended by the system due to an error. This command is also valid after a STRSBS (Start Subsystem) command where the prestart job entry indicates the prestart jobs should not be started when the subsystem is started. The number of jobs started is determined by the INLJOBS value on the prestart job entry and is limited by the MAXJOBS value for the subsystem.

Restriction: This command is restricted to a user with job control special authority. The user must also have *USE authority to the subsystem description.

Required Parameters

SBS Specifies the name of the active subsystem that contains the prestart job entry. Specify the name of the active subsystem.

PGM Specifies the qualified name of the program for the prestart job entry. This is also the name of the program that the prestart job runs. This program name is used to match an incoming request with an available prestart job.

The name of the program 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.

program-name: Specify the name of the program for the prestart job entry.

Example for STRPJ

```
STRPJ SBS(SBS1) PGM(PJLIB/PJPGM)
```

This command starts prestart jobs for prestart job entry PJPGM in subsystem SBS1. Subsystem SBS1 must be active when this command is issued. The number of jobs started is the number specified in the INLJOBS value of prestart job entry PJPGM. The subsystem starts program PJPGM in library PJLIB.

Error messages for STRPJ

*ESCAPE Messages

CPF0921

Start Prestart Jobs command not allowed now.

CPF1226

Start prestart jobs failed.

CPF1227

No authority has been granted to use command.

CPF1317

No response from subsystem for job &3/&2/&1.

CPF1351

Function check occurred in subsystem for job &3/&2/&1.

CPF1834

Prestart job entry for program &1 in &2 does not exist.

CPF1835

Not authorized to subsystem description.

STRPRTWTR (Start Printer Writer) Command Description

STRPRTWTR Command syntax diagram

Purpose

The Start Printer Writer (STRPRTWTR) command starts a spooling writer that moves spooled files from an output queue to a specified printer. The writer, which is a system job, takes spooled files from an output queue and produces (writes) the output on the printer device. This command specifies the name of the printer, the names of the output and message queues used, and the name of the writer.

More than one writer can be active at the same time (as determined by the spooling subsystem description), and up to 10 writers can be active to the same output queue. Each writer must have a unique writer name, its own device, and only one type of writer (print, remote, or diskette) can be active to a single output queue. A writer that has been started can be actively writing output or waiting for a file to be put on the output queue. Optionally, the writer can end automatically when it has processed all the files on the output queue. The writer can also be changed, held, or canceled if the Change Writer (CHGWTR), Hold Writer (HLDWTR), or End Writer (ENDWTR) command is used.

Because each writer runs independent of the job that started it, users can continue doing other work on the system after they have started a writer. The writer is owned by the user who issues the STRPRTWTR command.

Required Parameter

DEV Specifies the name of the printer device used to print the spooled file. The device must be available for allocation before the writer can be started.

***ALL:** A printer writer is started for every printer configured on the system.

***SYSVAL:** A printer writer is started for the system default printer.

printer-device-name: Specify the name by which the printer device being started is identified.

Optional Parameters

OUTQ Specifies the qualified name of the output queue.

***DEV:** The default output queue associated with the printer specified on the DEV parameter is used.

The name of the output queue 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.

output-queue-name: Specify the name of the output queue from which the writer processes spooled files.

WTR Specifies the name of the spooling writer being started. Each writer name must be unique.

***DEV:** The name of the writer is the same as that of the printer device specified on the DEV parameter.

writer-name: Specify the name by which the writer being started is identified.

MSGQ

Specifies the qualified name of the message queue to which messages created by the writer are sent.

***DEV:** Messages are sent to the message queue specified in the device description of the device named on the DEV parameter.

***REQUESTER:** The messages are sent to the workstation message queue of the workstation of the user who started the process. If this value is specified for a batch job, *DEV is used.

The name of the message queue 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.

message-queue-name: Specify the name of the message queue to which messages created by the writer are sent.

AUTOEND

Specifies whether the writer ends automatically.

***NO**: The writer does not end when the last available file has been removed from the output queue. It waits for another spooled file entry to be put on the queue.

Element 1: Ending a Writer Automatically

***YES**: The writer automatically ends after it has reached the state specified on Element 2 of this parameter (*NORDYF or *FILEEND).

Element 2: Ending a Writer with a Ready File

***NORDYF**: The writer automatically ends when there are no ready files (all the available files have been removed from the output queue).

***FILEEND**: The writer ends after it finishes processing one spooled file.

ALWDRTPRT

Specifies whether the printer writer allows files to be printed directly to the printer. A file printed directly to the printer is created by specifying SPOOL(*NO) for a printer file. When direct printing is allowed, the non-spooled printer file is printed immediately if the printer is available or, if the printer is busy, the non-spooled printer file waits until the printer is available. The maximum wait is the length of time specified on the WAITFILE parameter on the printer file, after which the job is automatically canceled. The user can cancel a non-spooled printer file only with an End Job (ENDJOB) command.

***NO**: The printer does not allow non-spooled printer files to be printed to the device.

***YES**: The printer can be used to print spooled and non-spooled output. See the Create, Change, or Override Printer File (CRTPRTF, CHGPRTF, or OVRPRTF) command to set the value of the WAITFILE parameter.

Note:

Nonspooled files wait up to 30 seconds regardless of whether the value specified on the WAITFILE parameter is less than 30 seconds.

FORMTYPE

Specifies the form type selection values that govern which spooled files are produced by the writer. A file's form type is specified in the device file that produced the spooled file. This parameter specifies that only the files with this form type are processed now. Files with other form types are left on the output queue and are available when their forms types are specified.

When a writer is sent to a printer device, a form load message is shown on the user's display if:

- The printer device has not been configured since it was last varied on.
- The form type mounted on the printer device does not match the form type of the first file spooled to the printer.

If the previous conditions occur, the form load message instructs the user to load the appropriate form type in the printer device.

To change the type of form for which spooled files are being produced, use the CHGWTR command.

Note:

The form load message is issued when the spooled file to be printed has a form type different from the form type of the last spooled file that was printed on the device. The value for the last form type printed is kept from the last STRPRTWTR, CHGWTR, or VRYCFG command issued.

Consider the following example:

1. The last spooled file printed on printer PRT01 had the form type *STD.
2. The user changes the form type on PRT01 to XYZ using the following command:

```
CHGWTR PRT01 FORMTYPE(XYZ)
```
3. No spooled file with the form type XYZ is printed on PRT01.
4. The user then sends a spooled file with the form type *STD to PRT01. The form load message is not issued, despite the intervening CHGWTR command, because the last spooled file printed on PRT01 had the same form type as the spooled file being printed.
The form load message would be issued if a spooled file with the form type XYZ were actually printed on PRT01.

Element 1: Type of Form Designation

***ALL:** All form types are processed by the writer.

***FORMS:** Available files on the output queue with the same form type are processed as a group before the writer moves on to the next form type group. The writer first chooses the first available file on the queue. After the first file is complete, all files with the same form type are processed. The writer again chooses the first available file on the queue and repeats the process for that form type.

***STD:** Only files that specify the standard form type are selected.

form-type: Specify the form type of the spooled files being produced.

Element 2: Message Sending Options

***INQMSG:** An inquiry message is sent to the message queue when a spooled file has a form type that is different than the form type in the printer.

***INFOMSG:** An informational message is sent to the message queue when no spooled files requiring this form type remain in the output queue.

***MSG:** An inquiry message is sent to the message queue when a spooled file has a form type that is different than the form type in the printer and an informational message is sent when no spooled files requiring this form type remain in the output queue.

***NOMSG:** Neither an inquiry message nor an informational message is sent to the message queue.

FILESEP

Specifies how to control the number of file separator pages that are printed preceding each file.

***FILE:** The number of separators specified for each individual file is used.

number-of-file-separators: Specify the number of separator pages to print. Valid values range from 0 through 9. Whenever the user responds to the change form type message indicating that a new form type has been put on the printer, the writer issues a message inquiring how many file separator pages are printed with the new form type.

SEPDRAWER

Specifies which drawer is selected for printing separators.

***DEV D:** The value stored in the device description for the printer is used.

***FILE:** The separator pages are printed on paper from the same drawer as the rest of the spooled file.

separator-drawer: Specify a value ranging from 1 through 255 to indicate the drawer from which the separator pages are printed.

Note: For some printers, SEPDRAWER(3) implies an envelope drawer.

ALIGN

Specifies how to control the forms alignment.

***WTR:** The writer keeps track of the output that is printed and issues a forms alignment message whenever it determines that alignment is needed.

***FILE:** The forms alignment message is issued for every file for which ALIGN(*YES) is specified. This option must be selected whenever the automatic forms alignment control provided by the writer does not provide the desired results.

***FIRST:** The forms alignment message is issued only for the first file printed. No alignment messages are issued when subsequent errors occur on the printer.

INIT Specifies how often to initialize the printer device.

Note: This parameter is ignored if TRANSFORM(*YES) or a user data transform program was specified on the printer device description.

***WTR:** The writer initializes the printer device when necessary.

***FIRST:** The writer initializes the printer device only before the first file is printed, or after a device error occurs.

***ALL:** The writer initializes the printer device before each file and each copy of the file is printed.

FILE Specifies the name of the first (or only) spooled file being processed by the spooling writer and printed. If several files are available on the output queue, the next file produced is the first one available that has the highest priority.

***NONE:** No spooled file name is specified. The first spooled file that becomes available on the output queue is processed first.

***LAST:** The spooled file which was being printed when the writer ended will start to print when the writer is restarted.

spooled-file-name: Specify the name of the spooled output file that is the first (or only) output file written to the printer.

JOB Specifies the name of the job that created the spooled file. This parameter is valid only if a spooled file name is specified on the FILE parameter. If no job qualifier is given, all of the jobs currently in the system are searched for the job name.

A job identifier is a special value or a qualified name with up to three elements. For example:

*
job-name
user-name/job-name
job-number/user-name/job-name

More information on this parameter is in Commonly used parameters.

***:** The job from which this STRPRTWTR command was entered is the job that created the spooled file.

job-name: Specify the name of the job that created the spooled file.

user-name: Specify the name of the user of the job that created the spooled file.

job-number: Specify the number of the job that created the spooled file.

SPLNBR

Specifies the number of the spooled file being processed first. This parameter is valid only if a spooled file name is specified in the FILE parameter. More information on this parameter is in Commonly used parameters.

***ONLY:** One spooled file from the job has the specified file name. The number of the spooled file is not necessary. If *ONLY is specified and more than one spooled file has the specified file name, a message is sent.

***LAST:** The spooled file with the highest number and the specified file name is used.

➤ ***ANY:** The spooled file number is not used to determine which spooled file is used. Use this value when the job system name parameter or the spooled file creation date and time parameter is to take precedence over the spooled file number when selecting a spooled file. ◀

spooled-file-number: Specify the number of the job's spooled file that is on the specified output queue and that is being processed first.

➤ JOBSYSNAME

Specifies the name of the system where the job that created the spooled file to be processed first (JOB parameter) ran. This parameter is considered after the job name, user name, job number, spooled file name, and spooled file number parameter requirements have been met. This parameter is valid only if a spooled file name is specified in the FILE parameter.

***ONLY:** There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and spooled file creation date and time.

***CURRENT:** The spooled file created on the current system with the specified job name, user name, job number, spooled file name, spooled file number, and creation date and time is used.

***ANY:** The job system name is not used to determine which spooled file is used. Use this value when the spooled file creation date and time parameter is to take precedence over the job system name when selecting a spooled file.

system name: Specify the name of the system where the job that created the spooled file ran.

CRTDATE

Specifies the date and time the spooled file to be processed first was created. This parameter is considered after the job name, user name, job number, spooled file name, spooled file number, and job system name parameter requirements have been met. This parameter is valid only if a spooled file name is specified in the FILE parameter.

***ONLY:** There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.

***LAST:** The spooled file with the latest creation date and time of the specified job name, user name, job number, spooled file name, spooled file number, and job system name is used.

Element 1: Date spooled file was created

date: Specify the date the spooled file was created.

Element 2: Time spooled file was created

***ONLY**: There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file creation date.

***LAST**: The spooled file with the latest creation time of the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file creation date is used.

time: Specify the time the spooled file was created. <<

PAGE Specifies the number of the first page to print of the first file specified on the FILE parameter.

***BEGIN**: Printing begins on the restart page of the spooled file. If the RESTART parameter value is *STRPAGE on the Change Spooled File Attributes (CHGSPLFA) command, then printing begins on the page specified on the PAGERANGE parameter, which is also on the CHGSPLFA command.

page-number: Specify the number of the page on which printing begins. The page number must be within the page range of the file that is to be printed. This value overrides the value specified on the RESTART parameter on the CHGSPLFA command.

Example for STRPRTWTR

```
STRPRTWTR DEV(QSYSPRT) OUTQ(QPRINTS) WTR(TOM)
```

This command starts a spooling writer named TOM. This writer takes the output from the output queue named QPRINTS and prints the output on the printer named QSYSPRT. Writer messages are sent to the system operator's message queue, and the writer waits for more output when the queue is emptied.

Error messages for STRPRTWTR

*ESCAPE Messages

CPF0906

A duplicate job named &3/&2/&1 was found.

CPF1338

Errors occurred on SBMJOB command.

CPF1764

Writer already started for device &1.

CPF1842

Cannot access system value &1.

CPF2115

Object &1 in &2 type *&3 damaged.

CPF2207

Not authorized to use object &1 in library &3 type *&2.

CPF3303

File &1 not found in job &5/&4/&3.

CPF3305

Output queue &1 in library &2 assigned to another writer.

CPF3309

No files named &1 are active.

CPF3310

Writer &1 already started.

CPF3330

Necessary resource not available.

CPF3340

More than one file with specified name found in job &5/&4/&3.

CPF3342

Job &5/&4/&3 not found.

CPF3343

Duplicate job names found.

CPF3347

Device &1 not found.

CPF3357

Output queue &1 in library &2 not found.

CPF3362

Objects in QTEMP not valid for parameter values.

CPF3363

Message queue &1 in library &2 not found.

CPF3369

Device &1 not printer device.

CPF336B

Not found or not authorized to driver exit program &1 in library &2.

CPF3418

Duplicate file &1 number &2 found in job.

CPF346A

Transform exit program &1 in library &2 not found or user is not authorized.

CPF346B

Driver exit program &1 in library &2 not found or user is not authorized.

CPF3463

Output queue for device &1 not found.

CPF3464

Not authorized to output queue &1 in library &2.

CPF3478

File &1 not found in job &5/&4/&3 on output queue &6 in library &7.

STRPGMPRF (Start Program Profiling) Command Description

STRPGMPRF Command syntax diagram

Purpose

The Start Program Profiling (STRPGMPRF) command starts collection of profiling information of ILE programs or service programs that have been compiled or changed with the program profiling option of PRFDTA(*COL). All programs compiled or changed with this option that are active will have profiling information updated until an ENDPGMPRF command is issued.

The profiling information will be added to existing profiling information. If this is not desired, the profiling data can be cleared by specifying PRFDTA(*CLR) on the CHGPGM (Change Program) or CHGSRVPGM (Change Service Program) CL command.

There are no parameters for this command.

Restriction: The command requires *ALLOBJ authority.

Example for STRPGMPRF

```
STRPGMPRF
```

Program profiling is started.

Error messages for STRPGMPRF

***ESCAPE Messages**

CPF5CAA

Unexpected error occurred during program profiling.

STRPGMMNU (Start Programmer Menu) Command Description

STRPGMMNU Command syntax diagram

Purpose

The Start Programmer Menu (STRPGMMNU) command shows the Programmer Menu. This command can be used instead of the CALL QPGMMENU function, and allows the user to pass parameters to specify and control the data which appears in the associated fields on the programmer menu.

Notes:

1. A user exit program can be called instead of submitting a job when option 3 is selected.
2. The first four parameters control the defaults that appear when the menu is first displayed.

More information about using the Programmer Menu is in the CL Programming  book.

Optional Parameters

SRCFILE

Specifies the name of an existing source file that contains source file members being updated or to which new members are being added.

***DFT:** This is the default for the TYPE of source file being specified on the menu. This field is blank when shown on the display.

source-file-name: Specify the name of the source file being updated.

SRCLIB

Specifies the qualified name of the library that is searched for the source file.

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.

OBJLIB

Specifies the name of the library, which was created with option 3, that receives objects.

***DFT:** The library used is dependent on the menu option selected. If this value is specified, the object library field will be blank.

***CURLIB:** The current library for the job contains or receives the object.

library-name: Specify the name of the library that receives the created object.

JOB Specifies the job description used to submit jobs for batch processing.

***USRPRF:** The job description included in the user profile is used for the job.

job-description-name: Specify the name of the job description that is used for the job. (The job description is found through the library list being used by the job.)

ALWUSRCHG

Specifies whether the menu screen fields, specified in the previous parameters in this command, can be changed by the user.

***YES:** The values on the display can be changed.

***NO:** The display fields cannot be changed. This option is intended to minimize errors, and is not considered a security function.

EXITPGM

Specifies the qualified name of a user-written program that is called as an exit program in place of submitting a batch job when menu option 3 is selected. When the exit program is called, it receives parameters that are sent by the Programmer Menu. (Refer to the examples at the end of this command description.) More information about the EXITPGM parameter is in the CL

Programming CL Programming  book.

***NONE:** No user-written program is called; a batch job is submitted. When *NONE is specified, DLTOPT(*DLT) must be specified.

The name of the program 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.

program-name: Specify the name of the program called when option 3 is selected, instead of submitting the create command as a batch job. When a value is specified for this parameter, the text that appears on the menu for option 3 shows the name and library of the exit program.

DLTOPT

Specifies the action that is taken when (1) a program name is specified for the EXITPGM parameter, (2) option 3 of the Programmer Menu is selected, and (3) an object of the name and type being created exists in the library specified by the OBJLIB field of the menu. Regardless of

the value specified, the system passes a parameter (from among the parameters passed from the Programmer Menu) to the exit program that specifies whether the object exists.

***DLT:** This value must be specified if *NONE is specified for the EXITPGM parameter. If an EXITPGM is specified, and the object specified to be created with option 3 exists, and the Enter key is pressed, a message is shown; press the F11 key to proceed. When the F11 key is pressed, the system deletes or replaces the object before calling the program specified by the exit program. This is the normal Programmer Menu function when an EXITPGM is not specified.

Notes:

1. When the *DLT value is specified, the object is deleted or replaced before the job is submitted or the user exit program is called.
2. If the source type is one of the following, the object is replaced rather than deleted (an exit program must be used to delete instead of replace):

BAS	C	CBL
CBL36	CLP	DSPF
DSPF36	FTN	ICFF
MNU36	MSGF36	PAS
PLI	PRTF	RPG
RPG36	RPT36	

***PROMPT:** The system does not delete or replace the object, but the user is prompted for approval to delete the object. If the object exists, and the Enter key is pressed, a message is displayed. Press the F11 key to proceed; the system does not delete the object. The user confirms whether the object is deleted or replaced, yet the deletion is still controlled by the exit program.

***NODLT:** The user exit program is called regardless of the presence of the object.

Table 1 shows the actions taken by the system when the exit program is called. The value passed to the exit program is a character variable of length 1 that contains the character value shown. Note that a value other than 0 indicates that the object does not exist.

Table 1. Actions Taken by the System When an Exit Program Is Called

DLTOBJ Specified	F11 Required	Object Deleted	Value Passed to Exit Program
If object existed when option 3 was selected			
*DLT	Yes	Yes	1
*PROMPT	Yes	No	0
*NODLT	No	No	0
If object did not exist when option 3 was selected			
*DLT	No	—	2
*PROMPT	No	—	2
*NODLT	No	—	2

Examples for STRPGMMNU

Example 1: Displaying Programmer Menu

STRPGMMNU

This command displays the Programmer Menu with defaults for all parameters. This has the same result as entering CALL QPGMMENU.

Example 2: Preventing Values from Being Changed

STRPGMMNU SRCFILE(YOURFILE) SRCLIB(YOURLIB)
OBJLIB(YOURLIB) JOB(YOURJOB) ALWUSRCHG(*NO)

This command prevents the values on the menu from being changed from those specified on the command.

Example 3: Calling an Exit Program

```
STRPGMMNU EXITPGM(OPT3PGM) DLTOPT(*PROMPT)
```

This command calls user exit program OPT3PGM instead of submitting a batch job when option 3 is specified. If the object already exists, DLTOPT(*PROMPT) requires the user to press the F11 key; however, the object is not deleted.

Example 4: Receiving Parameters

The following portion of a CL program is an example of how these parameters would be received by a user exit program. If the specified type is one of those listed, the object is not deleted. The create command with REPLACE(*YES) specified is passed to the exit program. The value passed to the exit program is 0.

```
PGM PARM(&OPTION &PARM &TYPE &PARM2
  &SRCFILE &SRCLIB
  &OBJLIB &JOBID &RQSLEN &RQSDTA512
  &F4 &F11 &EXIST)

/* The following values are passed in exactly as */
/* they appear on the Programmer Menu.          */
DCL VAR(&OPTION) TYPE(*CHAR) LEN(2)
DCL VAR(&PARM) TYPE(*CHAR) LEN(10)
DCL VAR(&TYPE) TYPE(*CHAR) LEN(10)
DCL VAR(&PARM2) TYPE(*CHAR) LEN(21)
DCL VAR(&SRCFILE) TYPE(*CHAR) LEN(10)
DCL VAR(&SRCLIB) TYPE(*CHAR) LEN(10)
DCL VAR(&OBJLIB) TYPE(*CHAR) LEN(10)
DCL VAR(&JOBID) TYPE(*CHAR) LEN(10)

/* The following values are derived by QPGMMENU */
/* from the information entered to the above fields */
/* and the F keys.                               */

/* NUMBER OF BYTES OF REQUEST DATA */
DCL VAR(&RQSLEN) TYPE(*DEC) LEN(3 0)

/* DATA FOR RRQSDTA PARAMETER OF SBMJOB COMMAND. */
DCL VAR(&RQSDTA512) TYPE(*CHAR) LEN(512)

/* F4 WAS PRESSED, '1', OTHERWISE '0'. */
DCL VAR(&F4) TYPE(*CHAR) LEN(1)

/* F11 WAS PRESSED, '1', OTHERWISE '0'. */
DCL VAR(&F11) TYPE(*CHAR) LEN(1)

/* OBJECT EXISTS- '0' OBJECT WAS DELETED- '1'
OR OBJECT DID NOT EXIST - '2'*/
DCL VAR(&EXIST) TYPE(*CHAR) LEN(1)
```

Additional information, along with examples of the STRPGMMNU command with the EXITPGM parameter, is in the CL Programming  book.

No error messages.

STRQSH (Start QSH) Command Description

STRQSH Command syntax diagram

Purpose

The Start QSH (STRQSH) command starts the **qsh** command shell interpreter.

QSH is an alias for this CL command.

Optional Parameters

CMD Specifies the shell command to be run.

***NONE:** No command is provided. If run in an interactive job, STRQSH starts an interactive shell session. If a shell session is not already active in the job, then:

1. A new shell session is started and a terminal window is displayed.
2. **qsh** runs the commands from the file */etc/profile* if it exists.
3. **qsh** runs the commands from the file specified by the expansion of *{\$HOME/.profile}* if it exists.
4. **qsh** runs the commands from the file specified by the expansion of the ENV variable if it exists.

If a shell session is already active in an interactive job, you are reconnected to the existing session. From the terminal window, you can enter shell commands and view output from the commands.

Using the Terminal Window

The terminal window has two parts:

- an input line for entering commands, and
- an output area that contains an echo of the commands you entered and any output generated by the commands.

The terminal window supports the following function keys:

F3 (Exit)

Close the terminal window and end the qsh session.

F6 (Print)

Print the output area to a spool file.

F7 (Up)

Roll output area up one page.

F8 (Down)

Roll output area down one page.

F9 (Retrieve)

Retrieve a previous command. You can press this key multiple times to retrieve any previous command. For example, to retrieve the second to last command you entered, press this key two times. You can also select a specific command to be run again by placing the cursor on that command and pressing this key. When the interactive job is running in a double-byte CCSID, this key is not available.

F11 (Toggle line wrap)

Toggle the line wrap/truncate mode in the output area. In line wrap mode, lines longer than

the width of the terminal window are wrapped to the next line. In truncate mode, the portion of a line beyond the width of the terminal window is not shown.

F12 (Disconnect)

Disconnect from the **qsh** session. This key only closes the terminal window and does not end the **qsh** session. You can redisplay the disconnected **qsh** session by running STRQSH again.

F13 (Clear)

Clear the output area.

F14 (Move)

Move the terminal window on your screen.

F15 (Resize)

Resize the terminal window on your screen.

F17 (Top)

Display top of output area.

F18 (Bottom)

Display bottom of output area.

F21 (CL command line)

Display a command entry window where you can enter CL commands.

If CMD(*NONE) specified and STRQSH is run in a batch job, STRQSH does nothing.

'command': Specify a character string of up to 1024 characters that represents a command that is run by **qsh**.

The command must be enclosed in apostrophes if it contains embedded blanks or special characters.

STRQSH starts a shell interpreter, runs the specified command, prints output generated by the command, and ends the shell interpreter.

Examples for STRQSH

Example 1: Run a Shell Command

```
STRQSH CMD('javac mypgm')
```

This starts a command shell interpreter, runs the *javac* command, and ends the shell interpreter. Any output generated to *stdout* by the *javac* command will be printed.

Example 2: Start the Shell Interpreter

```
STRQSH
```

If the command is run in an interactive job, a shell session is started and a terminal window is displayed. If the command is run in a batch job, nothing is done.

Error messages for STRQSH

*ESCAPE Messages

QSH0002

Error found with QSH session, reason code &1, errno &2.

STRQRY (Start Query) Command Description

STRQRY Command syntax diagram

Purpose

The Start Query (STRQRY) command shows the Query/400 main menu.

There are no parameters for this command.

Example for STRQRY

```
STRQRY
```

This command shows the main Query menu.

No error messages.

STRQMPCR (Start Query Management Procedure) Command Description

STRQMPCR Command syntax diagram

Purpose

The Start Query Management Procedure (STRQMPCR) command allows the user to run a query management procedure that was saved in a source file member.

Required Parameter

SRCMBR

Specifies the name of the member in the source file that contains the query management procedure to be run.

Optional Parameters

SRCFILE

Specifies the qualified name of the source file that contains the query management procedure to run.

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.

QQMPCRSRC: The file having the IBM-supplied source file name, QQMPCRSRC, is used.

source-file-name: Specify the name of the source file.

RDB Specifies the name of the relational database that is accessed during the processing of this command.

***NONE:** The local database is accessed. If the user is connected to a remote database, the connection is reset to local and remains local until completion of this command.

***CURRENT:** The relational database to which the user is currently connected is accessed.

relational-database-name: Specify the name of the relational database that is accessed. The database must have an entry in the relational database directory.

RDBCNNMTH

Specifies the connection method to use.

***DUW:** Connections to several relational databases are allowed. Consecutive CONNECT statements to additional relational databases do not result in disconnection of previous connections.

***RUW:** Only one connection to a relational database is allowed. Consecutive CONNECT statements result in the previous connections being disconnected before a new connection is established.

USER Specifies the user name sent to the remote system when starting the conversation.

***CURRENT:** The user name associated with the current job is used.

user-name: Specify the user name being used for the application requester job.

PASSWORD

Specifies the password to be used on the remote system.

***NONE:** No password is sent. The user name specified on the USER parameter is not valid if this value is specified.

password: Specify the password of the user name specified on the USER parameter.

NAMING

Specifies the naming convention used for naming objects.

***SYS:** The system naming convention is used (library-name/object-name).

***SAA:** The System Application Architecture (SAA) naming convention is used (data-base-name.object-name). If NAMING(*SAA) is specified, CMDSRCFILE(*LIBL) cannot be specified or allowed as a default value for locating any objects specified on other parameters on this command.

ALWQRYDFN

Specifies whether query or form information is taken from a QRYDFN object when no QMQRY or QMFORM object can be found using the specified object name. Any information that has to be derived in this way is discarded when the common programming interface (CPI) command in the procedure is completed. No query management objects are created.

***NO:** The information is not taken from a QRYDFN object.

***YES:** The information is taken from a QRYDFN object when the specified QMQRY or QMFORM object is not found.

***ONLY:** The information is taken only from a QRYDFN object. Query management objects are ignored.

CMDSRCFILE

Specifies the qualified name of the command source file that query management uses to run a command procedure. A command procedure can only contain query management set commands which can set application variables as well as query management variables that start with the 'DSQ' value. The supported DSQ variables are:

- DSQCMTLV
- DSQCONFIRM
- DSQOAUTH

- DSQSDBNM
- DSQSRUN

***NONE:** A command source file is not used. The CMDSRCMBR parameter is ignored.

The name of the command 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.

command-source-file-name: Specify the name of the command source file.

CMDSRCMBR

Specifies the name of the command source member that query management uses to run a command procedure. A command procedure can only contain query management set commands which set variables that start with the 'DSQ' value.

***FIRST:** The first member is used.

command-source-member-name: Specify the name of the command source member.

ALWDSPLAY

Specifies the display mode used. The query management session is set to interactive mode if ALWDSPLAY(*YES) is specified. If ALWDSPLAY(*NO) is specified, then the session is set to batch mode. The mode is automatically set to batch when running this command in a batch environment.

***YES:** Displays are shown when used in an interactive session. This mode allows the user to interact with the query management commands in the procedure.

***NO:** No displays are shown.

Examples for STRQMPC

Example 1: Running a Query Management Procedure

```
STRQMPC SRCMBR(MYPROC) SRCFILE(RPTLIB/PROCFIL)
```

This command starts the query management procedure stored as the member named MYPROC in the source file named PROCFILE in the RPTLIB library.

Example 2: Taking Information From QRYDFN Objects

```
STRQMPC SRCMBR(MYPROC) SRCFILE(PROCFIL)
  ALWQRYDFN(*YES) ALWDSPLAY(*NO)
```

This command starts the query management procedure stored as the member named MYPROC in the first file named PROCFILE in the library list for the job. Query and form information is allowed to be taken, as needed, from QRYDFN objects when the procedure statements are processed. No reports are shown but they can be printed if the user specifies a print request. Objects are replaced without confirmation if confirmation is not requested by the user. The procedure ends with some errors if processing locates a global variable that is not set or if confirmation was requested before replacing objects that already exist.

Error messages for STRQMPC

***ESCAPE Messages**

QWM2701

&1 command failed.

QWM2703

&1 command ended.

QWM2707

*LIBL not allowed when SAA naming applied.

QWM2709

User or password not valid with relational database value.

QWM2710

Password value *NONE only valid with user value *CURRENT.

QWM2712

Character in user name not valid.

STRQMQRYP (Start Query Management Query) Command Description

STRQMQRYP Command syntax diagram

Purpose

The Start Query Management Query (STRQMQRYP) command is used to run a query.

The user of this command must first identify the query that is to be processed. The query is any single Structured Query Language (SQL) statement in a QMQRYP object. The SQL statement can also be taken from a query definition (QRYDFN) object when a QMQRYP object does not exist.

The user can show the output on the display, print it, or store it in a database file.

If the SQL statement inside the query does not create an answer-set, then no report or output file is created. This happens if the SQL statement inside a query is not valid or the SQL statement is not a SELECT clause.

If the query contains substitution variables, the SETVAR parameter can be used to set the variables for the query. If prompting is enabled, query management asks the user to provide a value for each variable that was not set.

Required Parameter

QMQRYP

Specifies the name of the query management query to be run. This parameter is required when running a query.

The name of the query 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.

query-name: Specify the name of the query to be run.

Optional Parameters

OUTPUT

Specifies whether the output from the command is shown at the requesting work station, printed with the job's spooled output, or directed to a database file.

***:** The output produced by the query is formatted with the specified report form and, in interactive mode, sent to the work station that runs the command. If the command is run in batch mode, the output is sent to the default printer used by query management.

***PRINT:** The output produced by the query is formatted with the specified query management form, then sent to the default printer used by query management.

***OUTFILE:** The output produced by the query is written to a database file (table), which is inserted into a collection.

QMFORM

Specifies which query management report form is to be applied to the answer-set to format the printed or displayed output.

***SYSDFT:** A default report form is created and used for the report that is printed or displayed.

***QMQR:** The value specified on the QMQR parameter is used to locate the report form.

The name of the report form 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.

report-form-name: Specify the name of the report form.

OUTFILE

Specifies the database file that receives the query output. If the file specified does not exist, the system creates it in the specified library as a table in a collection. If the file is created by this function, the authority for users without specific authority is *EXCLUDE.

The name of the database 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.

database-file-name: Specify the name of the database file that receives the output of the command.

OUTMBR

Specifies the name of the database file member to which the output is directed.

Element 1: Member to Receive Output

***FIRST:** The first member in the file receives the output.

Element 2: Operation to Perform on Member

***REPLACE:** The file is cleared before new records are inserted.

***ADD:** New records are added after any existing records.

DATETIME

Specifies whether the system date and time are printed on the bottom of each page.

***YES:** The system date and time are printed on the bottom of each page.

***NO:** The system date and time are not printed on the bottom of each page.

PAGNBR

Specifies whether page numbers are printed on the bottom of each page.

***YES:** The page numbers are printed on the bottom of each page.

***NO:** The page numbers are not printed.

RDB

Specifies the name of the relational database that is accessed during the processing of this command.

***NONE:** The local database is accessed. If the user is connected to a remote database, the connection is reset to local and remains local until completion of this command. If the connection management method is *DUW, the remote connection is left in a dormant state.

***CURRENT:** The relational database to which the user is currently connected is accessed.

With *RUW connection management, if the user is connected to a remote database, *OUTFILE cannot be specified on the OUTPUT parameter.

With *DUW connection management, if the user is connected to a remote database and OUTPUT(*OUTFILE) is specified, the connection is set to local for the *OUTFILE processing and then the remote connection is restored upon completion of the STRQMQRV command.

relational-database-name: Specify the name of the relational database that is accessed. The database must have an entry in the relational database directory.

With *RUW connection management, if the relational database specified is a remote database and OUTPUT(*OUTFILE) is specified, the connection is reset to local for the *OUTFILE processing and remains local upon completion of the STRQMQRV command.

With *DUW connection management, if the relational database specified is a remote database and OUTPUT(*OUTFILE) is specified, the connection is set to local for the *OUTFILE processing and then the remote connection is restored upon completion of the STRQMQRV command.

RDBCNNMTH

Specifies the connection method to use.

***DUW:** Connections to several relational databases are allowed. Consecutive CONNECT statements to additional relational databases do not result in disconnection of previous connections.

***RUW:** Only one connection to a relational database is allowed. Consecutive CONNECT statements result in the previous connections being disconnected before a new connection is established.

USER

Specifies the user name sent to the remote system when starting the conversation.

***CURRENT:** The user name associated with the current job is used.

user-name: Specify the user name being used for the application requester job.

PASSWORD

Specifies the password to be used on the remote system.

***NONE**: No password is sent. The user name specified on the USER parameter is not valid if this value is specified.

password: Specify the password of the user name specified on the USER parameter.

NAMING

Specifies the naming convention used for naming objects.

***SYS**: The system naming convention is used.

***SAA**: The System Application Architecture (SAA) naming convention is used. If NAMING(*SAA) is specified, the *LIBL value cannot be specified or allowed to be a default value for locating any objects specified on other parameters on this command.

ALWQRYDFN

Specifies whether query or form information is taken from a QRYDFN object when a QMQRY or QMFORM object cannot be found using the object name specified by the user. Any information that has to be derived in this way is discarded when the command has completed processing. No query management object is created.

***NO**: Information is not taken from a QRYDFN object.

***YES**: Information is taken from a QRYDFN object when the specified QMQRY or QMFORM object name is not found.

***ONLY**: Information is taken only from a QRYDFN object. Query management objects are ignored.

SETVAR

Specifies the variables that are set by query management before the query is run. Up to 50 variables can be set.

Element 1: Variable Name

variable-name: Specify a variable name. Valid values range from 1 to 30 characters. Because lower-case characters in variable names are changed to upper-case characters when passed to the command processing program, the user cannot use this parameter to set values for variables with mixed case names.

Element 2: Variable Value

variable-value: Specify a variable value. Valid values range from 0 to 55 characters. If the user encloses a value in apostrophes, the apostrophes are removed and double apostrophes within the value are condensed to single apostrophes when the value is passed to the command processing program.

Examples for STRQMQR

Example 1: Displaying Query Output

```
STRQMQR QMQRY(MYLIB/MYQRY) QMFORM(FORM1)
```

This command runs query management query MYQRY located in library MYLIB. The library list is searched for form FORM1, which is used for the output sent to the display.

Example 2: Taking Information From Either QMQRY or QRYDFN

```
STRQMQR QMQRY(MYLIB/MYQRY) QMFORM(FORM1) ALWQRYDFN(*YES)
```

This command runs query management query (QMQRY) MYQRY located in library MYLIB. If QMQRY object MYQRY is not found in library MYLIB, the information is taken from query definition (QRYDFN)

MYQRY located in library MYLIB. The library list is searched for query management form FORM1 whose information is used to format the output. If QMFORM object FORM1 is not found in the library list, the library list is searched for QRYDFN FORM1, and that information is used to format the output shown on the display.

Example 3: Printing Query Output

```
STRQMQR Y QMQR Y(MYLIB/QUERY1) OUTTYPE(*PRINTER)
```

This command runs query QUERY1 located in library MYLIB. The report is formatted and printed on the printer specified in the printer file associated with the query session.

Example 4: Sending Output to an Existing File

```
STRQMQR Y QMQR Y(*CURLIB/MYQRY) OUTPUT(*OUTFILE)
  OUTFILE(MYTAB) OUTMBR(*FIRST *ADD)
```

This command runs the query named MYQRY located in the current library for the user's job. The selected data records are added to the previously created table named MYTAB in collection MYCOL.

Example 5: Running a Query Containing Substitution Variables

```
STRQMQR Y QMQR Y(MYQUERY)
  SETVAR((VAR1 'select * from mytable')
  (VAR2 'where salary > 15000'))
```

This command runs query MYQUERY, which contains only substitution variables, &VAR1 and &VAR2. These two variables contain the entire structured query language (SQL) statement.

Example 6: Changing a Variable

```
STRQMQR Y QMQR Y(QRYNAME) SETVAR((LASTNAME ''Smith''))
```

This command runs query QRYNAME, setting the variable LASTNAME to the value, 'Smith'.

Error messages for STRQMQR Y

*ESCAPE Messages

QWM2701

&1 command failed.

QWM2703

&1 command ended.

QWM2707

*LIBL not allowed when SAA naming applied.

QWM2709

User or password not valid with relational database value.

QWM2710

Password value *NONE only valid with user value *CURRENT.

QWM2712

Character in user name not valid.

STRQST (Start Question and Answer) Command Description

STRQST Command syntax diagram

Purpose

The Start Question and Answer (STRQST) command shows the user the main Question & Answer (Q & A) menu. More information is available in the Basic System Operations topic in the Information Center.

Optional Parameters

QSTDB

Specifies the Q & A database with which to work.

***SELECT:** The user is 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 work.

LIB Specifies the name of the library that contains the Q & A database.

***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 for which the user is 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 for which the user is authorized can be selected.

Example for STRQST

```
STRQST
```

This command shows the Question and Answer (Q & A) main menu.

No error messages.

STRRCYBRM (Start Recovery Using BRM) Command Description

Note: To use this command, you must have the 5722-BR1 (Backup Recovery and Media Services for iSeries) licensed program installed. For detailed information on the parameters of this command, see the online help.

STRRCYBRM Command syntax diagram

Purpose

The Start Recovery using BRM (STRRCYBRM) command recovers items ranging from a single library to the entire system. Any library or group of libraries that were saved by a BRMS save command can be restored by this command. Recovery is based on entries in the BRMS media content information.

Recovery using the STRRCYBRM uses the most recent times and dates of entries in the BRMS media content information. This reflects the order in which libraries and objects were saved. For instance, a recovery request for a library will restore the most recent full backup of a library, the latest incremental backup of that library, and all object level saves that have dates after the last full save.

Note: Prior to processing a restore, you should use the report option (*REPORT) to produce a report of the steps required to restore a portion of the system or the entire system. The only time the STRRCYBRM command can be run in batch is when *REPORT is specified in the ACTION parameter. The report that is produced is the Recovery Analysis report. The report, if printed, is written to printer file QP1ARCY.

Note: This command should not be used by control group *EXIT item processing as results will be unpredictable.

Example for STRRCYBRM

Example 1: Reporting ASP02 Selected To Restore

STRRCYBRM OPTION(*ASP) ASP(2)

In this example a report is produced to assist you in restoring saved items to ASP02.

Error messages for STRRCYBRM

None

STRRMTSPT (Start Remote Support) Command Description

STRRMTSPT Command syntax diagram

Purpose

The Start Remote Support (STRRMTSPT) command creates and varies on all configuration objects needed for remote support. Remote support allows the IBM service organization to access your system. Remote support options that are available include:

- remote work station
- virtual device
- virtual device over a systems network architecture (SNA) connection
- virtual device over a point-to-point protocol (PPP) connection internet protocol (IP)
-  virtual device over a virtual private network (VPN) connection using internet protocol (IP) 

Each remote support option has different command parameters that are required for connectivity. If any existing remote support configuration objects are found, they are deleted and then re-created. After the configuration objects have been created, they are varied on. You must provide a user identifier and password before the support person can sign on your system.

Restrictions:

- When using DEVCLS(*RMT) the remote work station used by a support organization must be one of those listed in the DSPTYPE and DSPMODEL parameters. If your support organization has a configuration that does not match, you must work with the support person to create the correct configuration objects on your system.
- When using DEVCLS(*VRT) or DEVCLS(*IPS), the QUSER user profile must not be disabled.

Required Parameter

DEVCLS

Specifies the device class for this display station. Different configuration objects are created, depending on the value specified. The possible values are:

***RMT:** This device class is for a device connected to a remote work station. The configuration objects created for this option include a line description, a controller description, a display device description, and a printer device description.

***VRT:** This device class is for a virtual device. The configuration objects created for this option include a line description, a controller description, a device description, a virtual control description, and a virtual display device description.

***IPS:** This device class is for a virtual device and also supports internet protocol (IP) over SNA sessions on this connection. The configuration objects created for this option include a line description, a controller description, a device description, a virtual control description, and a virtual display device description.

***PPP:** This device class supports IP over a point-to-point protocol(PPP) connection. The configuration objects created for this option include a line description, a controller description, a device description, and a PPP profile. DEVCLS(*PPP) requires that the QRETSVRSEC system

value be set to '1' to retain server security data. This allows for additional authentication to be performed when the service organization attempts to connect. DEVCLS(*PPP) requires that the QRETSVRSEC System Value be set to '1' to retain server security data. This allows for additional authentication to be performed when the service organization attempts to connect.

» ***VPN:** This device class supports IP over a virtual private network (VPN) connection. The Universal Connection wizard must be run before running the STRRMTSPT command in order to set up a VPN configuration on the system. Once the STRRMTSPT command is entered, eleven alphanumeric characters are displayed on your screen. These characters need to be told to the Customer Support Representative for a complete connection to be established. No configuration objects are created for this option. «

Optional Parameters

DSPTYPE

Specifies the work station device type which is used for remote support. A value must be specified for this parameter when DEVCLS has a value of *RMT, *VRT, or *IPS. Valid display station type values are:

Device Type
3179
3180
3196
3197
5251
5291
5292

DSPMODEL

Specifies the model number of the device for this description. A value must be specified for this parameter when DEVCLS has a value of *RMT, *VRT or *IPS. The possible values for each device type are:

Device Type	Model
3179	2
3180	2
3196	A1, A2, B1, B2
3197	C1, C2, D1, D2, W1, W2
5251	11
5921	1, 2
5292	1, 2

STNADR

Depending on the value specified for the DEVCLS parameter, this parameter specifies the address that will be used for either the station address or the local IP address qualifier. If the DEVCLS is

not *PPP or *VPN, then this parameter specifies the station address. The station address is the hexadecimal address by which the local system is known to the remote system. The hexadecimal address is the polling address assigned to this system.

If the DEVCLS is *IPS, *PPP, or *VPN, this parameter is used as the local IP address qualifier. This specifies the value of the last byte of the local internet address that will be used.

The possible values are:

FE: The hexadecimal value FE is the local system address.

address: Specify a hexadecimal value ranging from 01 through FE.

USRPRF

Specifies the name of the user profile that the IBM service personnel will use to sign on to your system. This profile is made the owner of the objects created by the system when remote support is started. The user profile must already exist on your system. The possible values are:

QPGMR: The default system-supplied user profile, QPGMR, is used to sign on to your system.

user-profile-name: Specify the name of the existing user profile that will be used to sign on to the remote system.

RSRCNAME

Specifies the name of the resource used to access iSeries electronic customer support. This parameter is not used for DEVCLS(*VPN). The possible values are:

***DFT:** The resource name will be determined as follows:

- For DEVCLS(*PPP):
 - Look for resources being used by the 2771 integrated modem. If only one 2771 is defined, that resource is used for this PPP line. (Note that more than one 2771 could be defined, but a 2771 resource can only be calculated if only one is defined.)
 - If a 2771 modem resource cannot be used, determine if any resources are defined for use by electronic customer support (ECS). If an ECS resource is available, that resource is used 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.
- For all other DEVCLS values (except *VPN):

The resource name associated with the shipped default port for accessing electronic customer support is used. This is the first port on the I/O (input/output) adapter in card position B of the first multifunction IOP (input/output processor) on the bus. If this port does not exist on the system then CMN01 will be used.

resource-name: Specify the name of the resource used to access electronic customer support.

RMTLOCNAME

Specifies the remote location name of the system with which this object communicates. The parameter is used when DEVCLS is *VRT or *IPS. The possible values are:

QREMOTE: The default system-supplied remote support location name.

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

LCLLOCNAME

Specifies the unique location name that identifies the local system to remote devices.

Note:

The name cannot be the same as that specified for the RMTLOCNAME parameter. The combination of the names specified for the LCLLOCNAME parameter and the RMTLOCNAME parameter must be unique for each device attached to the same controller. This parameter is used when the DEVCLS parameter has a value of *VRT or *IPS. The possible values are:

QLOCAL: The default system-supplied local location name is used.

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

local-location-name: Specify the location name to be used to identify the local system to remote devices.

RMTNETID

Specifies the name of the remote network in which the remote system resides. This parameter is used when the DEVCLS parameter has a value of *VRT or *IPS. The possible values are:

***NONE:** No remote network name is used.

remote-network-name: Specify the remote network name.

MAXLENRU

Specifies the maximum request unit (RU) length allowed. This parameter is used when the DEVCLS parameter has a value of *VRT or *IPS. The possible values are:

32767: The maximum request unit length is 32767 bytes.

***CALC:** The system calculates the maximum value to use.

maximum-request-unit-length: Specify a value, ranging from 241 through 32767 bytes, for the maximum length of incoming request units.

Some common values are, based on line type, are:

- SDLC lines: 256, 512, 1024, 2048
- Token-Ring Network lines: 256, 512, 1024, 1985
- X.25 (QLLC) lines: 247, 503, 1015
- X.25 (ELLC) lines: 241, 497, 1009

More information is in the Communications Configuration  book.

DTACPR

Specifies whether data compression is used. The possible values are:

***REQUEST:** Data compression is requested on the session by the local system. However, the request can be refused or changed to lower compression levels by the remote system. Data compression is allowed on the session if requested by the remote system. The requested compression levels for inbound and outbound data are the levels specified on the Inbound data compression and Outbound data compression prompts (INDTACPR and OUTDTACPR parameters).

If data compression is requested by the remote system, the data compression levels used by the session are the lower of the requested levels and the levels specified on the Inbound data compression and Outbound data compression prompts (INDTACPR and OUTDTACPR parameters).

***NONE:** Compression is not allowed on the session.

***ALLOW:** Data compression is allowed on the session by the local system if requested by a remote system. The local system does not request compression.

If data compression is requested by the remote system, the data compression levels used by the session are the lower of the requested levels and the levels specified on the Inbound data compression and Outbound data compression prompts (INDTACPR and OUTDTACPR parameters).

***REQUIRE:** Data compression is required on the session. If the remote system does not accept the local system's exact required levels of compression, the session is not established.

The data compression levels that the local system require are the levels specified on the Inbound data compression and Outbound data compression prompts (INDTACPR and OUTDTACPR parameters).

INDTACPR

Specifies the desired level of compression for inbound data. No data compression occurs if *NONE is specified on the Data compression prompt (DTACPR parameter). Adaptive dictionary-based compression is a dynamic compression algorithm, similar to Lempel-Ziv, that compresses previously seen strings to 9-, 10-, and 12-bit codes. This algorithm is referred to as LZ in the following parameters. The possible values are:

***LZ12:** The default in-bound data-compression. The LZ algorithm with the 12-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. LZ12 requires the most storage and processing time of the LZ algorithms; however, it compresses the data stream the most.

***RLE:** The Run Length Encoding (RLE) algorithm is used. RLE substitutes a 1- or 2-byte sequence in the data stream for each repeated occurrence of the same character. This algorithm requires no storage and less processing time than the other options.

***LZ9:** The LZ algorithm with the 9-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ9 requires the least storage and processing time of the LZ algorithms; however, it compresses the data stream the least.

***LZ10:** The LZ algorithm with the 10-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ10 table algorithm requires more storage and processing time than the LZ9, but less than the LZ12. The LZ10 compresses the data stream more than the LZ9, but less than the LZ12.

OUTDTACPR

Specifies the desired level of compression for outbound data. No data compression occurs if *NONE is specified on the Data compression prompt (DTACPR parameter). The possible values are:

***LZ12:** The default out-bound data-compression. The LZ algorithm with the 12-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require

storage and extra processing time. LZ12 requires the most storage and processing time of the LZ algorithms; however, it compresses the data stream the most.

***RLE:** The Run Length Encoding (RLE) algorithm is used. RLE substitutes a 1- or 2-byte sequence in the data stream for each repeated occurrence of the same character. This algorithm requires no storage and less processing time than the other options.

***LZ9:** The LZ algorithm with the 9-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ9 requires the least storage and processing time of the LZ algorithms; however, it compresses the data stream the least.

***LZ10:** The LZ algorithm with the 10-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ10 table algorithm requires more storage and processing time than the LZ9, but less than the LZ12. The LZ10 compresses the data stream more than the LZ9, but less than the LZ12.

MODEM

Specifies the modem description to use for the point-to-point protocol (PPP) profile. The possible values are:

***RSRCNAME:** The modem description 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 is used.
- If the resource is defined to use the 2772 integrated modem, the '2772 Internal Modem' description is used.
- If the resource is defined to use the 2761 internal modem, the '2761 Internal Modem' description is used.
- If the ECS resource was chosen, the 'IBM 7852-400' modem description is 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 select the modem to use. This option is only valid when running the STRRMTSPT CL 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.

'modem-identification': 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.

'generic-modem-identification':* Specify the 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 with 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 joblog.

Examples for STRRMTSPT

Example 1: Start Remote Support using PPP

```
STRRMTSPT DEVCLS(*PPP) STNADR(FA)
```

This command creates and starts a PPP answer profile and associated configuration objects. The local internet address assigned is 169.254.2.250.

Example 2: Start Remote Support for Virtual Device

```
STRRMTSPT DEVCLS(*VRT) DSPTYPE(5251) DSPMODEL(11)  
STNADR(FE)
```

This command creates and varies on the 5251 Model 11 Display Station located at station address FE.

➤ Example 3: Start Remote Support using VPN

```
STRRMTSPT DEVCLS(*VPN)
```

This command starts remote support over a VPN connection. The Universal Connection wizard needs to be run successfully before this command can be run.

Error messages for STRRMTSPT

*ESCAPE Messages

CPF1394

User profile &1 cannot sign on.

CPF9801

Object &2 in library &3 not found.

CPF9899

Error occurred during processing of command.

TCP83A7

QRETSVRSEC=0. Unable to save remote service password for &2.



STRRMTWTR (Start Remote Writer) Command Description

STRRMTWTR Command syntax diagram

Purpose

The Start Remote Writer (STRRMTWTR) command starts a spooling writer that sends spooled files from an output queue to a remote system. The writer, which is a system job, takes spooled files from an output queue and sends them to a remote system using SNADS or TCP/IP.

After the spooled file is successfully sent to a remote system, the spooled file will be deleted or saved, as specified by the SAVE spooled file attribute.

More than one writer can be active at the same time (as determined by the spooling subsystem description), and up to 10 writers can be active to the same output queue. Each writer must have a unique writer name, and only one type (print, remote, or diskette) of writer can be active to a single output queue. A writer that has been started can be actively sending output or waiting for a file to be put on the output queue. Optionally, the writer can end automatically when it has processed all the files on the output queue. You can also change, hold, or cancel the writer.

You can continue with other work after starting a writer because each job runs independently. The writer is owned by the user who issues the STRRMTWTR command.

Required Parameter

OUTQ Specifies the qualified name of the output queue.

***ALL:** Starts remote writers for every output queue on the system which have a remote system specified. No writer starts if an output queue has a different type of writer already started. The number of “writers to autostart” attribute in the output queue determines the number of writers started to a particular queue. This will determine how many remote writer jobs will be sending spooled output to a remote system, from a single output queue.

The name of the output queue 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.

output-queue-name: Specify the name of the output queue from which the writer processes output files.

Optional Parameters

WTR Specifies the name of the spooling writer being started. Each writer name must be unique.

***OUTQ:** The name of the writer is the same as that of the output queue specified on the OUTQ parameter. If OUTQ(*ALL) is specified, and you have more than one writer to start, the name for each additional writer is taken from the first nine characters of the output queue followed by a digit. This last digit is sequentially assigned, starting with 2 for the second writer, 3 for the third writer, and continuing through 0 for the tenth writer.

***SYSGEN:** The writer name is generated by the system, starting with ‘RMTW000001’ and incrementing the numeric part for each successive writer.

writer-name: Specify the name by which the writer being started is identified.

MSGQ

Specifies the qualified name of the message queue to which messages created by the writer are sent.

***OUTQ:** Messages are sent to the message queue of the output queue specified on the OUTQ parameter.

***REQUESTER:** The messages are sent to the workstation message queue of the workstation of the user who started the process. If this value is specified for a batch job, *OUTQ is used.

The name of the message queue 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.

message-queue-name: Specify the name of the message queue to which messages created by the writer are sent.

AUTOEND

Specifies whether the writer ends automatically.

Element 1: Stop Writer Option

***NO**: The writer does not end when the last available file has been removed from the output queue. It waits for another spooled file entry to be put on the queue. This is a single value (Element 2 is not specified).

***YES**: The writer automatically ends after it has reached the condition specified on Element 2 of this parameter.

Element 2: Conditions for Stopping Writer

***NORDYF**: The writer automatically ends when there are no ready files (all the available files have been removed from the output queue).

***FILEEND**: The writer ends after it finishes processing one spooled file.

FORMTYPE

Specifies the name of the form type. The writer uses this value to select spooled files from the specified output queue for sending to a remote system.

Note:

A spooled file's form type is specified in the device file that produced the spooled file.

Element 1: Type of Form Designation

***ALL**: All form types are processed by the writer.

***FORMS**: The writer first chooses the first available spooled file on the output queue. After the first spooled file is complete, all spooled files with the same form type are processed. The writer then chooses the first available spooled file on the output queue and repeats the process for that form type.

***STD**: Only spooled files that specify the standard form type are selected.

form-type: The form type of the spooled files being produced.

Element 2: Message Sending Options

***NOMSG**: Neither an inquiry message nor an informational message is sent to the message queue.

***INQMSG**: An inquiry message is sent to the message queue when a spooled file has a form type that is different than the form type last sent.

***INFOMSG**: An informational message is sent to the message queue when no spooled files requiring this form type remain in the output queue.

***MSG**: An inquiry message is sent to the message queue when a spooled file has a form type that is different than the form type in the remote and an informational message is sent when no spooled files requiring this form type remain in the output queue.

Example for STRRMTWTR

STRRTWTR OUTQ(RMTOUTQ) WTR(TOM)

This command starts a spooling writer named TOM. This writer takes the output from the output queue named RMTOUTQ and sends the output to the remote system and printer queue specified in the output queue RMTOUTQ. Writer messages are sent to the system operator's message queue, and the writer waits for more output when the queue is emptied.

No error messages.

STRREXPRC (Start REXX Procedure) Command Description

STRREXPRC Command syntax diagram

Purpose

The Start REXX Procedure (STRREXPRC) command calls the REXX interpreter to start a REXX procedure that resides in the specified source member.

Required Parameter

SRCMBR

Specifies the name of the source file member that contains the REXX procedure to be started.

Optional Parameters

SRCFILE

Specifies the source file containing the REXX procedure to be started.

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.

QREXSRC: The IBM-supplied source file QREXSRC contains the REXX procedure to be started.

source-file-name: Specify the name of the source file that contains the REXX procedure to be started.

PARM Specifies procedure parameter values passed to the REXX procedure when it is started. These values are accessed through the argument (ARG) instruction within the REXX procedure.

***NONE:** There are no procedure parameters for the REXX procedure. The ARG instruction returns a null string.

procedure-parameter-string: Specify the parameter string to be passed to the REXX procedure. Up to 3000 characters are allowed.

CMDENV

Specifies the initial command environment program name to process commands that are merged within the REXX procedure. The REXX interpreter calls this environment whenever a command is encountered during the REXX procedure.

***COMMAND:** The iSeries 400 command environment is used.

***CPICOMM:** The Common Programming Interface (CPI) for communications command environment is used. CPICOMM is the command environment used for CL commands that are imbedded within a REXX procedure.

***EXECSQL** The Structured Query Language (SQL) Command environment is used. EXECSQL is the command environment used for CL commands that are imbedded within a REXX procedure.

The name of the program 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.

program-name: Specify the name of the program.

EXITPGM

Specifies exit programs to be used when the interpreter is called. A maximum of 8 exit programs can be specified.

***NONE:** There are no exit programs for the interpreter being called.

The name of the program 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.

Element 1: Exit Program

program-name: Specify up to eight program names.

Element 2: Exit Code

exit-code: Specify up to eight exit codes. The valid exit codes that can be specified for this value range from 2 through 5 and 6 through 8.

Exit-code Description

1. The associated program is called whenever an external function or subroutine is called by the REXX program. The exit program is then responsible for locating and calling the requested routine.
2. The associated program is called whenever the interpreter is going to call a command. The exit program is responsible for locating and calling the command given the command string and the current environment name.
3. The associated program is called whenever a REXX instruction or function attempts an operation on the REXX external data queue.

4. The associated program is called when session input or output operations are attempted.
5. The associated program is called after running each clause of the REXX procedure to determine whether it is to be halted.
6. The associated program is called after running each clause of the REXX program to check if tracing is to be turned on or off.
7. The associated program is called before interpretation of the first instruction of a REXX procedure (including REXX procedures called as external functions and subroutines).
8. The associated program is called after interpretations of the last instruction of a REXX procedure (including REXX procedures called as external functions and subroutines).

Example for STRREXPRC

```
STRREXPRC SRCMBR(ABC)
```

This command calls the REXX interpreter instructing it to run the source member named ABC in the first QREXSRC source file in the library list (*LIBL).

Error messages for STRREXPRC

*ESCAPE Messages

CPF7CFB

Error occurred while processing REXX exit programs.

CPF7CFD

Error occurred running REXX procedure &1.

CPF7CFF

REXX procedure &1 ended; return code &4.

CPF7CF2

REXX procedure &1 not found.

CPF7CF3

Not authorized to source file &2.

CPF7CF4

Cannot allocate REXX procedure &1.

CPF7CF6

Cannot allocate REXX source file &2.

CPF7CF7

REXX external data queue is damaged.

RPCBIND (Start RPC Binder Daemon) Command Description

RPCBIND Command syntax diagram

Purpose

The Start RPC Binder Daemon (RPCBIND) command starts the Remote Procedure Call (RPC) binder daemon. The RPC binder daemon job must be running to use and run Network File System (NFS) daemons and commands and some of the TI-RPC APIs.

This command can also be issued using the following alternative command:

- STRNFSSVR SERVER(*RPC)

However, the STRNFSSVR command has the following restrictions:

1. You must have *IOSYSCFG special authority to use this command.

If you attempt to start this daemon and it is already running, it will not cause the command to fail. The command will issue diagnostic message CPCA1B6 if the daemon is already running.

To determine if the RPC server daemon is running, use the Work with Active Jobs (WRKACTJOB) command and look in the subsystem QSYSWRK for existence of the following job:

QNFSRPCD The RPC binder daemon

Restrictions:

1. You must have *IOSYSCFG special authority to use this command.

Optional Parameters

RTVRPCREG

Specifies whether to retrieve previously recorded registration information when the RPC daemon is started.

If registration information is retrieved, servers do not have to re-register with the RPC server.

***NO:** Do not retrieve registration information.

***YES:** Retrieve registration information.

Example for RPCBIND

Example: Start RPC Binder Daemon

RPCBIND RTVRPCREG(*YES)

This command starts the RPC binder daemon job, and retrieves previously recorded registration information.

Error messages for RPCBIND

None

STRSCHIDX (Start Search Index) Command Description

STRSCHIDX Command syntax diagram

Purpose

The Start Search Index (STRSCHIDX) command is used to access a search index without using the Help key or the F11 key.

Restriction: The user of this command must have *USE authority for the search index.

Optional Parameter

SCHIDX

Specifies the qualified name of the search index from which the index entries are displayed. If this parameter is not used, the search index names that the user has saved are used.

The following special value may be used:

***USER:** The search index names that the user has saved are used. If no names were saved, the search index display is shown with an empty list message.

The name of the search index 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.

search-index-name: Specify the name of the search index.

Example for STRSCHIDX

STRSCHIDX

This command accesses the search index names that the user has saved.

Error messages for STRSCHIDX

*ESCAPE Messages

CPF6E66

Requested help information not available.

STRSRVJOB (Start Service Job) Command Description

STRSRVJOB Command syntax diagram

Purpose

The Start Service Job (STRSRVJOB) command starts the remote service operation for a specified job (other than the job issuing the command) so that other service commands can be entered to service the specified job. Dump, debug, and trace commands can be run in that job until service operation ends. Service operation continues until the End Service Job (ENDSRVJOB) command is run.

The following commands can be used in the serviced job while remote service is being done:

Dump Job (DMPJOB)

Dump Job Internal (DMPJOBINT)

Dump Object (DMPOBJ)

Dump System Object (DMPSYSOBJ)

Start Debug (STRDBG)

Trace Internal (TRCINT)

Trace Job (TRCJOB)

Trace ICF (TRCICF)

Restrictions:

- This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.

- The issue of the command must be running under a user profile which is the same as the job user identity of the job being serviced, or which has use (*USE) authority to the job user identity of the job being serviced.

Required Parameters

JOB Specifies the name of the job being serviced. If no job number is given, all of the jobs currently in the system are searched for the simple job name. If duplicates of the specified name are found, messages are sent to the user, and the user name and job number must be specified. The job name entered cannot be the name of the job issuing the command.

A job identifier is a qualified name with up to three elements. For example:

```
job-name
user-name/job-name
job-number/user-name/job-name
```

More information on this parameter is in Commonly used parameters.

job-name: Specify the name of the job being serviced.

user-name: Specify the name of the user of the job being serviced.

job-number: Specify the number of the job being serviced.

DUPJOB OPT

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.

Example for STRSRVJOB

```
STRSRVJOB JOB(ABCD)
```

This command starts the remote service operation so that any trace, debug, or dump commands entered in this job are applied to the job named ABCD.

Error messages for STRSRVJOB

*ESCAPE Messages

CPF3501

Job is already being serviced, traced, or debugged.

CPF3520

Job not found.

CPF3524

More than one job with specified name found.

CPF3531

Job cannot be serviced.

CPF3536

Job completed and cannot be serviced.

CPF3549

Job &1/&2/&3 cannot be serviced.

CPF3676

Not authorized to service specified job.

CPF3909

Service command will not be processed.

CPF3918

Service request canceled.

CPF3938

Already servicing another job.

STRSBS (Start Subsystem) Command Description

STRSBS Command syntax diagram

Purpose

The Start Subsystem (STRSBS) command starts a subsystem based on the subsystem description specified in the command. Once started, the subsystem is known by the unqualified name of the subsystem description (the subsystem description name without the library name). When the subsystem is started, the system allocates the necessary and available resources (storage, work stations, and job queue) that are specified in the subsystem description.

Storage is allocated to the subsystem according to the storage pool definitions specified in the subsystem description, starting with the lower numbered storage pool definitions. If all the pool definitions cannot be allocated, because the maximum number of storage pools on the system is reached or because insufficient storage is available, messages indicating which pools could not be allocated are sent to the system operator. If storage becomes available later, or if the number of active storage pools is reduced, the available resources are automatically allocated to the subsystem to satisfy its unfulfilled requirements. Any jobs that would normally run in a storage pool that is not allocated are run in the shared storage pool (*BASE).

Allocating Work Stations: Work stations are allocated to the subsystem according to the work station entries in the subsystem description. Each work station whose name (or type, if not specified by name) is contained in one of the subsystem description's work station entries, and whose entry specifies AT(*SIGNON), is allocated to this subsystem unless it is currently signed on to another subsystem. The sign-on prompt is displayed on each work station that is allocated. Work stations that are already signed on in another subsystem remain allocated to that subsystem until the subsystem that allowed the sign-on is ended, or until the user transfers the job to this subsystem. (Messages indicating the names of the work stations that could not be allocated are sent to the system operator.)

If multiple subsystems specify the same work station in their work station entries, each subsystem, as it is started, attempts to allocate that work station. Each successive subsystem allocates that work station unless a user signs on while the work station is allocated to one of the previously started subsystems. When a signed-on work station is signed off, it still remains allocated to the same subsystem until another subsystem is started that specifies that work station. If, however, a work station is varied offline and several active subsystems specify that work station, the subsystem to which the work station is allocated when it is varied online is unpredictable.

Allocating Job Queues: If a job queue is specified in the work entries of the subsystem description, the job queue is allocated to the subsystem. If the job queue does not exist or if it is already allocated to an active subsystem, no job queue is allocated to the subsystem and a message is sent to the system operator. If the job queue later becomes available, it is *not* automatically allocated to the subsystem. To allocate the job queue to the subsystem, the subsystem must end, then start again.

Restriction: To start a subsystem, the user must have job control (*JOBCTL) special authority.

Required Parameter

SBSD Specifies the qualified name of the subsystem description that defines the operational environment (subsystem) being started.

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

The name of the subsystem description 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.

subsystem-description-name: Specify the name of the subsystem description that defines the subsystem being started.

The name of the subsystem description cannot be the same as the name of a subsystem that is currently active, even though the subsystem descriptions are in different libraries.

Examples for STRSBS

Example 1: Starting a Subsystem

```
STRSBS SBSD(QBATCH)
```

This command starts the batch subsystem named QBATCH.

Example 2: Starting a Subsystem

```
STRSBS SBSD(QGPL/TELLER)
```

This command starts the subsystem that is associated with the TELLER subsystem description in the QGPL library. The subsystem name is TELLER.

Error messages for STRSBS

*ESCAPE Messages

CPF1001

Wait time expired for system response.

CPF1004

Function check occurred during start subsystem.

CPF1010

Subsystem name &1 active.

CPF1011

Start subsystem failed for SBSD &1 in library &2.

CPF1012

No authority to start subsystem.

- CPF1013**
Subsystem &1 in library &2 not found.
- CPF1014**
Subsystem &1 not started.
- CPF1031**
Not authorized to library &1.
- CPF1038**
No authority to use command.
- CPF1049**
Cannot allocate subsystem &1 in library &2.
- CPF1050**
Not enough storage to start subsystem.
- CPF1057**
Subsystem &1 in library &2 damaged.
- CPF1067**
Cannot allocate library &1.
- CPF1080**
Library &1 not found.
- CPF1086**
Subsystem &1 in &2 allocated to your job.
- CPF1099**
Subsystem not started because system ending.
- CPF3D87**
Attempted to use system program QLPCTLIN in QSYS.

STRSBSBRM (Start Subsystems Using BRM) Command Description

Note: To use this command, you must have the 5722-BR1 (Backup Recovery and Media Services for iSeries) licensed program installed. For detailed information on the parameters of this command, see the online help.

STRSBSBRM Command syntax diagram

Purpose

The Start Subsystems using BRM (STRSBSBRM) command restarts subsystems that have been ended as a result of control group processing. You can specify the control group name and the type of control group that you are restarting subsystems for.

Example for STRSBSBRM

Example 1: Restarting Subsystems After a Control Group Processes

```
STRSBSBRM CTLGRP(ARCHIVE01) TYPE(*ARC)
```

In this example subsystems are restarted after the archive control group ARCHIVE01 completes archive processing.

Error messages for STRSBSBRM

None

STRSPTN (Start Support Network) Command Description

STRSPTN Command syntax diagram

Purpose

The Start Support Network (STRSPTN) command opens a communications path through the remote support network to the specified destination application.

This command is provided for customers who want to write their own programs to interface with one of the remote support systems.

Required Parameters

ACCOUNT

Specifies the user's support network organization account number. The organization account ID must be registered with the support network.

SPTNUSRID

Specifies the network user ID in the account specified in the ACCOUNT parameter. The user ID must be registered with the support network and must be valid for the specified account.

SPTNPWD

Specifies the network password for the specified user. The password must be registered with the support network and must be valid for the specified user.

Note:

A support network password can expire. The user must then change the password interactively by using the Work with Product Information (WRKPRDINF) command.

FEADEV

Specifies the description of the device used as the Front End Application (FEA).

DESTAPP

Specifies the destination application (DESTA) to which the communication path is connected.

Example for STRSPTN

```
STRSPTN ACCOUNT(11420880) SPTNUSRID(ACME)
      SPTNPWD(11111) FEADEV(QTIFEA) DESTAPP(AAAAAA)
```

This command establishes a communication path through the remote support network, for user ID ACME operating under password 11111 at account 11420880. The path allows access to application AAAAAA.

Error messages for STRSPTN

None >>

STRSYSMGR (Start System Manager) Command Description

Note: To use this command, you must have the 5722-SM1 (System Manager for iSeries) licensed program installed.

STRSYSMGR Command syntax diagram

Purpose

The Start System Manager (STRSYSMGR) command starts the jobs in the QSYSWRK subsystem that runs the System Manager licensed program functions.

The Start Managed System (STRMGDSYS) command is used to start the services that are common to both a central site system and a managed system. These services must be running in order for the central site system support to be fully operational.

The jobs running the managed system functions, which are started by the STRMGDSYS command, are checked and a diagnostic message is sent if they are not started. This message is only sent for managed system functions that apply to a central site system.

Restrictions:

1. You must have *JOBCTL authority to use this command.
2. *PUBLIC authority for this command is *EXCLUDE.

Example for STRSYSMGR

STRSYSMGR

This command starts the jobs in the QSYSWRK subsystem that run the System Manager functions.

Error messages for STRSYSMGR

***ESCAPE Messages**

None <<

STRSST (Start System Service Tools) Command Description

STRSST Command syntax diagram

Purpose

The Start System Service Tools (STRSST) command shows the System Service Tools (SST) menu.

The user can:

- Start a service function
- Work with active service tools
- Work with disk units
- Work with diskette data recovery
- Work with system partitions

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The QSRV user profile has private authorities to use the command.
3. You must have *SERVICE special authority to use this command.

There are no parameters for this command.

Example for STRSST

STRSST

This command shows the Start System Service Tools menu.

Error messages for STRSST

*ESCAPE Messages

CPC7210

System Service Tools detected function check. CAUTION: Any device(s) in diagnostic mode not reset.

CPC7211

System Service Tools detected function check. See message CPF5263.

CPF7215

System Service Tools already active in this process.

CPF7242

Not authorized to system service tools.

CPF7243

Previous request not completed.

STRS36 (Start System/36) Command Description

STRS36 Command syntax diagram

Purpose

The Start System/36 (STRS36) command starts a System/36 environment session (if one is not already active). Even if the System/36 environment is active, this command allows the user to show a menu or run a program or procedure before showing a menu.

When this command ends, the System/36 environment returns to the active or inactive state from which this command is run.

- If the System/36 environment is active when this command is issued, the environment remains active when the command ends. You can press the Exit or Previous key at the menu to return to the point at which the command was issued.
- If the System/36 environment is inactive when this command is issued, the environment again becomes inactive when the command ends. You must use the End System/36 (ENDS36) command to exit from the menu.

Restrictions: This command cannot be used if a System/36 procedure is already in process. This command cannot be placed in a procedure or in a program that is started by a procedure.

Optional Parameters

MENU Specifies the first menu that is shown when the System/36 environment is started.

***SAME:** The menu specified in the job does not change. If no menu is specified in the job, the initial menu specified in the user profile is shown.

menu-name: Specify the name of the menu that is shown first when the System/36 environment is started.

CURLIB

Specifies the name of the library to use for the current library in the System/36 environment.

***SAME:** The current library does not change. If the current library is *CRTDFT, the current library is set to #LIBRARY.

library-name: Specify the name of the library to use for the current library in the System/36 environment.

PRC Specifies the name of the procedure or program to run before the menu is shown.

***NONE:** No procedure or program runs.

procedure-name: Specify the name of the procedure or program to run.

FRCMENU

Specifies whether a menu is shown and what this command does in an active System/36 environment.

***NO:** The specified menu is not shown if the System/36 environment is active when this command is run. This command does nothing, and the user is returned to the point at which the command was issued.

***YES:** The specified menu is shown even if the System/36 environment is active when this command is run. The current library is set and the program or procedure is run as specified in this command.

Example for STRS36

```
STRS36 MENU(user) CURLIB(MYLIB) PRC(INITPROC)
```

This command starts a System/36 environment session. This command:

- Displays the user menu
- Changes the current library to MYLIB
- Runs procedure INITPROC before showing the user menu

Error messages for STRS36

***ESCAPE Messages**

CPF3709

Tape devices do not support same densities.

CPF3738

Device &1 used for save or restore is damaged.

CPF3767

Device &1 not found.

STRS36PRC (Start System/36 Procedure) Command Description

STRS36PRC Command syntax diagram

Purpose

The Start System/36 Procedure (STRS36PRC) command starts a System/36 procedure. It is valid whether or not the System/36 Environment is active, but it is not valid if a System/36 procedure is already running. It cannot be placed in a procedure or in a program that is called by a procedure.

Required Parameter

PRC Specifies the name of the System/36 procedure to run.

Optional Parameters

CURLIB

Specifies the name of the library being used as the current library for jobs initiated by this user profile.

***SAME:** The value does not change.

library-name: Specify the name of the library to use for the current library while running the System/36 procedure.

PARM Specifies procedure parameters for the procedure. Procedure parameters allow information to be passed to the procedure. If no parameters are specified, no parameters are passed to the procedure.

Examples for STRS36PRC

Example 1: Changing the Current Library

```
STRS36PRC PRC(PROC1) CURLIB(MYLIB)
```

This command changes the current library to MYLIB and runs procedure PROC1.

Example 2: Listing Files Used by the System

```
STRS36PRC PRC(CATALOG) PARM('ALL,F1')
```

This command lists all files used by the System/36 Environment.

Error messages for STRS36PRC

*ESCAPE Messages

SSP0010

System/36 job ended abnormally.

STRTCP (Start TCP/IP) Command Description

STRTCP Command syntax diagram

Purpose

The Start TCP/IP (STRTCP) command initializes and activates TCP/IP processing, starts the TCP/IP interfaces, and starts the TCP/IP server jobs. A STRTCP command must be issued before any TCP/IP processing can be performed on the iSeries. This includes Simple Network Management Protocol (SNMP) agent processing.

The TCP/IP interfaces that are started are those that have set the AUTOSTART parameter to a value of *YES using the Add TCP/IP Interface (ADDTCPIFC) command, the Change TCP/IP Interface (CHGTCPIFC) commands, or the iSeries Navigator. [»](#)

Note:

If TCP/IP is started when no subsystems are active (i.e. the system is in **restricted state**), no application server jobs or TCP/IP interfaces are started.



Jobs can be started for these TCP/IP application servers:

- Simple Network Management Protocol (SNMP) agent
- Router Daemon (ROUTED)
- Bootstrap Protocol (BOOTP)
- Trivial File Transfer Protocol (TFTP)
- Domain Name Server (DNS)
- Dynamic Host Configuration Protocol (DHCP)

- Distributed Data Management (DDM)
- Virtual terminal support (TELNET)
- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)
- Line Printer Daemon (LPD)
- Hypertext Transfer Protocol (HTTP)
- Post Office Protocol (POP)
- Remote Execution (REXEC)
- Network Station Manager Inventory (NSMI)
- Distributed Computing Environment (DCE)
- Directory Services (DIRSRV)
- Ultimedia System Facilities (USF)
- Network Station Login Daemon (NSLD)
- Internet Daemon (INETD)
- Management Central (MGTC)
- On Demand (ONDMD)
- NetServer (NETSVR)
- DataLink File Manager (DLFM)
- Virtual Private Network (VPN)
- Extended Dynamic Remote SQL (EDRSQL)
- IBM Host On-Demand (HOD)
- On-Demand Platform Authentication (ODPA)
- Simple Network Time Protocol (NTP)
- Quality of Service (QoS)
- Triggered Cache Manager (TCM)
- Domino
- Lotus QuickPlace (LQP)
- 5250 WebFacing (WEBFACING)

The Start TCP/IP command only starts a TCP/IP application job if the AUTOSTART attribute in the application's configuration is set to *YES when the command is issued. No TCP/IP application jobs are started in the QSYSWRK subsystem if any of the following is true:

- The TCP/IP licensed program product is not installed.
- All of the TCP/IP applications have an AUTOSTART configuration attribute value of *NO.
- STRSVR(*NO) is specified for the STRTCP command. See the description of the STRSVR parameter below.

Use the commands listed below to change the configuration for an application so that it will be automatically started when the Start TCP/IP command is issued.

- For the SNMP agent use the Change SNMP Attributes (CHGSNMPA) command: CHGSNMPA AUTOSTART(*YES)
- For the RouteD server use the Change RouteD Attributes (CHGRTDA) command: CHGRTDA AUTOSTART(*YES)
- For the SMTP application use the Change SMTP Attributes (CHGSMTPA) command: CHGSMTPA AUTOSTART(*YES)
- For the Post Office Protocol (POP) version 3 mail servers use the Change POP Mail Server Attributes (CHGPOPA) command: CHGPOPA AUTOSTART(*YES)

- For the FTP application use the Change FTP Attributes (CHGFTP) command: CHGFTP AUTOSTART(*YES)
- For the TELNET application use the Change TELNET Attributes (CHGTELNA) command: CHGTELNA AUTOSTART(*YES)
- For the LPD application use the Change LPD Attributes (CHGLPDA) command: CHGLPDA AUTOSTART(*YES)
- For the World Wide Web HyperText Transfer Protocol (HTTP) server use the Change HTTP Attributes (CHGHTT) command: CHGHTT AUTOSTART(*YES)
- For the NTP application use the Change NTP Attributes (CHGNTP) command: CHGNTP AUTOSTART(*YES)
- For the Distributed Computing Environment (DCE) use the iSeries Operations Navigator to configure DCE and select to autostart upon TCP/IP startup.
- For Directory Services (DIRSRV) use the iSeries Navigator to configure DIRSRV and select to autostart upon TCP/IP startup.

» When the STRTCP command is issued and the iSeries is **not** in restricted state, « the QTCPIP job in the QSYSWRK subsystem is started. The QTCPIP job is used for activating and deactivating TCP/IP interfaces.

Note: The STRTCP command does not need to be issued to use socket applications that run over an SNA network.

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 Information (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.

Optional Parameters

STRSVR

Specifies whether or not TCP/IP application servers are started by the Start TCP/IP (STRTCP) command.

***YES:** The STRTCP command also starts all TCP/IP application servers with a configuration attribute of AUTOSTART(*YES) when TCP/IP is activated.

***NO:** The STRTCP command does not start any of the TCP/IP application servers when TCP/IP is activated. The AUTOSTART attribute in the application configuration is ignored.

STRIFC

Specifies whether or not to activate all TCP/IP interfaces that specify AUTOSTART(*YES) when TCP/IP is activated.

***YES:** When the QTCPIP job is started by the STRTCP command, the job will attempt to activate all TCP/IP interfaces for which AUTOSTART(*YES) is specified.

***NO:** The AUTOSTART(*YES) interface parameter will be ignored. No TCP/IP interfaces will be automatically started.

Examples for STRTCP

Example 1: Starting TCP/IP

```
STRTCP
```

This command initializes and activates TCP/IP processing, starts the TCP/IP interfaces, and starts the TCP/IP server jobs.

Example 2: Starting TCP/IP and TCP/IP Servers

```
STRTCP STRSVR(*YES)
```

Because *YES is the default value for the STRSVR parameter, the result of issuing this command is identical to the Example 1.

Example 3: Starting TCP/IP But Not the TCP/IP Servers

```
STRTCP STRSVR(*NO)
```

This will start TCP/IP processing without starting any of the TCP/IP application server jobs.

➤ Example 4: Starting TCP/IP in Restricted State

```
STRTCP STRSVR(*NO) STRIFC(*NO)
```

This will start TCP/IP processing, even if the system is in restricted state. TCP/IP application servers and IP interfaces will not be started. ⏪

Error messages for STRTCP

*ESCAPE Messages

CPF9848

Cannot open file &1 in library &2 member &3.

CPF9849

Error while processing file &1 in library &2 member &3.

TCP1A04

&1 currently active.

TCP1A12

Error occurred submitting interface job.

TCP1A14

Error occurred starting TCP/IP servers.

TCP1A77

&1 completed successfully; however errors occurred.

TCP1D03

&1 member record length not correct.

TCP1D04

Error occurred processing member &1 of &2/&3.

TCP9999

Internal system error in program &1.

STRTCPIFC (Start TCP/IP Interface) Command Description

STRTCPIFC Command syntax diagram

Purpose

The Start TCP/IP Interface (STRTCPIFC) command starts a Transmission Control Protocol/Internet Protocol (TCP/IP) interface. The line associated with the interface is varied on, if required.

This command can be used to:

- Start interfaces that have been specified with the AUTOSTART(*NO) value on the Add TCP/IP Interface (ADDTCPIFC) and Change TCP/IP Interface (CHGTCPIFC) commands.
- Start an interface that was previously ended by the End TCP/IP Interface (ENDTCPIFC) command.
-  Start an interface while the system is in restricted state. This is supported for Token Ring, Ethernet, or FDDI LAN lines that are **not** attached to an NWI or an NWS, or an ATM line. 

Routes are bound to interfaces using a best match first algorithm. This algorithm is based on the state of the interface and on the type of service (TOS) specified for the route and interface. When starting an interface, routes associated with an inactive interface can move to the interface started provided that interface can be used to reach the next hop gateway of the route for the requested TOS.

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 Information (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.

Required Parameter

INTNETADR

Specifies the interface to be started.

internet-address: Specify the internet address of an interface that had previously been added to the TCP/IP configuration using the Add TCP/IP Interface (ADDTCPIFC) command. 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.

***AUTOSTART**: The TCP/IP interfaces that are started are those that have set the AUTOSTART parameter to a value of *YES using the Add TCP/IP Interface (ADDTCPIFC) or the Change TCP/IP Interface (CHGTCPIFC) commands.

Examples for STRTCPIFC

Example 1: Starting an Interface

```
STRTCPIFC INTNETADR('9.5.11.125')
```

This command causes the TCP/IP protocol stack to activate the interface associated with the internet address 9.5.11.125.

Example 2: Starting another Interface

```
STRTCPIFC INTNETADR('156.93.81.7')
```

This command causes the TCP/IP protocol stack to activate the interface associated with the internet address 156.93.81.7.

Example 3: Starting AUTOSTART(*YES) Interfaces

```
STRTCPIFC INTNETADR(*AUTOSTART)
```

This command causes the TCP/IP protocol stack to activate all interfaces that have the AUTOSTART parameter set to *YES using the Add TCP/IP Interface (ADDTCPIFC) or Change TCP/IP Interface (CHGTCPIFC) commands.

Error messages for STRTCPIFC

***ESCAPE Messages**

TCP1B01

Unable to determine if &1 interface started.

TCP1B02

Cannot determine if &1 interface started.

TCP1B05

&2 interface not started. Reason &1.

TCP1B10

&2 interface not started.

TCP1B11

Interface &1 line &2 not started. Maximum &7 active interfaces allowed.

TCP1B12

&1 interface not started. &1 interface already active.

TCP1B13

&1 interface not started. &1 interface not defined in the TCP/IP configuration.

TCP1B14

&1 interface not started. Line description &2 not found.

TCP1B15

Line description &2 unusable. Internal errors encountered.

TCP1B16

&2 interface not started.

TCP1B25

&1 interface not started.

TCP1B26

&1 interface not started

TCP1B27

&1 interface not started for *TDLC line description &2.

TCP265F

INTNETADR parameter value &2 not valid.

TCP9999

Internal system error in program &1.

STRTCPSVR (Start TCP/IP Server) Command Description

STRTCPSVR Command syntax diagram

Purpose

The Start TCP/IP Server (STRTCPSVR) command is used to start the TCP/IP application servers. This command starts TCP/IP jobs for the application servers specified with the server (SERVER) parameter. The number of server jobs started by this command is specified, where appropriate, in the configuration for each TCP/IP application. ➤ The STRTCPSVR command can only be used when TCP/IP is fully

operational. The interface server job, QTCPIP, must be available. If TCP/IP processing was started by running the STRTCP (Start TCP/IP) command when the system was in restricted state, the QTCPIP server job was not started, and the STRTCPSVR command will fail. <<

Each type of TCP/IP server has an autostart (AUTOSTART) parameter maintained in a control file which indicates if the server should be started when the Start TCP/IP (STRTCP) command is entered. The STRTCPSVR command ignores the value of a server's autostart parameter unless SERVER(*AUTOSTART) is specified. Some of the supported servers can have the autostart value changed through a configuration CL command (for example, Change FTP Attributes (CHGFTP)).

Using iSeries Navigator, the ability to start a server automatically whenever TCP/IP starts can be accomplished by turning on the capability from a property panel.

>> Additional servers can automatically be added to list of servers that STRTCPSVR will support by using the ADDTCPSVR (Add TCP/IP Server) CL command. <<

Optional Parameters

SERVER

Specifies the TCP/IP application servers to be started by this command.

Single Values

***ALL:** All of the TCP/IP application servers and all HTTP and DNS instances are started.

***AUTOSTART:** All of the TCP/IP application servers that are defined as AUTOSTART *YES are started.

TCP/IP Application Servers

>> Additional servers could also be available if they were added with the ADDTCPSVR command. Press F4 when prompting the SERVER parameter for a complete list of supported parameter values. <<

***SNMP:** The Simple Network Management Protocol (SNMP) agent server is started. Subsequent use of the STRTCPSVR SERVER(*SNMP) command results in a diagnostic message if the SNMP server has already been started.

***ROUTED:** The Routed server is started. Subsequent usage of the STRTCPSVR SERVER(*ROUTED) command results in a diagnostic message if the Routed server has already been started.

***BOOTP:** The Bootstrap Protocol (BOOTP) server job is started. Subsequent usage of the STRTCPSVR SERVER(*BOOTP) command results in a diagnostic message if the BOOTP server has already been started.

***TFTP:** The configured minimum number of Trivial File Transfer Protocol (TFTP) servers are started. Subsequent use of the STRTCPSVR SERVER(*TFTP) command results in a diagnostic message if the TFTP servers have already been started.

***DNS:** The Domain Name Server (DNS) server is started. Subsequent use of the STRTCPSVR SERVER(*DNS) command results in a diagnostic message if the DNS server has already been started.

***DHCP:** The Dynamic Host Configuration Protocol (DHCP) server is started. Subsequent use of the STRTCPSVR SERVER(*DHCP) command results in a diagnostic message if the DHCP server has already been started.

***DDM:** The server that listens for distributed data management (DDM) and distributed relational database architecture (DRDA) connect requests and dispatches pre-started jobs to service them is started. Subsequent use of the STRTCPSVR SERVER(*DDM) command results in a diagnostic message if the DDM server has already been started.

***TELNET:** The TELNET server is started. Subsequent use of the STRTCPSVR SERVER(*TELNET) command starts one additional TELNET server.

Note: Having more than one TELNET server running reduces the chances of having connection attempts refused.

***FTP:** The File Transfer Protocol (FTP) servers are started based on the number of servers configured with the Change FTP Attributes (CHGFTP) command. Subsequent use of the STRTCPSVR SERVER(*FTP) command starts one additional FTP server.

Note: Having more than one FTP server running can improve the performance of initiating a session when multiple users attempt to connect to the server in a short period of time.

***SMTP:** The Simple Mail Transfer Protocol (SMTP) client and server jobs are started. Additional SMTP client and server jobs cannot be started. Subsequent use of the STRTCPSVR SERVER(*SMTP) command results in a diagnostic message if the SMTP server jobs have already been started.

***LPD:** The line printer daemon (LPD) servers are started based on the number of servers configured with the Change LPD Attributes (CHGLPDA) command. Subsequent use of the STRTCPSVR SERVER(*LPD) command starts one additional LPD server.

Note: LPD works most efficiently when two or more servers are running. Running only one server works, but no jobs can be received while a current job is running. If a large print job is running, new jobs have to wait before LPD is ready to accept any new line printer requester (LPR) requests.

***HTTP:** The World Wide Web HyperText Transfer Protocol (HTTP) servers are started based on the number of servers specified in the Change HTTP Attributes (CHGHTT) command. Subsequent use of the STRTCPSVR SERVER(*HTTP) command results in a diagnostic message if the HTTP server has already been started.

***POP:** The Post Office Protocol (POP) version 3 servers are started based on the number of servers configured with the Change POP Attributes (CHGPOPA) command. Subsequent use of the STRTCPSVR SERVER(*POP) command starts one additional POP server.

***REXEC:** The Remote EXECution (REXEC) servers are started based on the number of servers configured with the Change REXEC Attributes (CHGRXCA) command. Subsequent use of the STRTCPSVR SERVER(*REXEC) command starts one additional REXEC server.

Note: Having more than one REXEC server running can improve the performance of initiating a session when multiple users attempt to connect to the server in a short period of time.

***NSMI:** The Network Station Manager inventory (NSMI) server is started. Subsequent use of the STRTCPSVR SERVER(*NSMI) command results in a diagnostic message if the NSMI server has already been started.

***DCE:** The Distributed Computing Environment (DCE) servers are started based on the configuration specified using the iSeries Navigator. Subsequent use of the STRTCPSVR SERVER(*DCE) command starts any additional DCE servers that are configured, but are not started.

***DIRSRV:** The Directory Services (DIRSRV) servers are started based on the configuration specified using the iSeries Navigator. Subsequent use is ignored because only one DIRSRV server is allowed.

***USF:** The Ultimedia System Facilities (USF) server is started. Subsequent use of the STRTCPSVR SERVER(*USF) command results in a diagnostic message if the USF server has already been started.

***NSLD:** The Network Station Login Daemon (NSLD) server is started. Subsequent use of the STRTCPSVR SERVER(*NSLD) command results in a diagnostic message if the NSLD server has already been started.

***INETD:** The Internet daemon (INETD) super-server is started. Subsequent use of the STRTCPSVR SERVER(*INETD) command results in a diagnostic message if the INETD server has already been started.

***MGTC:** The Management Central (MGTC) server is started. Subsequent use of the STRTCPSVR SERVER(*MGTC) command results in a diagnostic message if the MGTC server has already been started.

***ONDMD:** The OnDemand Client for iSeries (ONDMD) server is started. Subsequent use of the STRTCPSVR SERVER(*ONDMD) command results in a diagnostic message if the ONDMD server has already been started.

***NETSVR:** The NetServer (NETSVR) server is started. Subsequent use of the STRTCPSVR SERVER(*NETSVR) command results in a diagnostic message if the NETSVR server has already been started.

***DLFM:** The DataLink File Manager (DLFM) server is started. Subsequent use of the STRTCPSVR SERVER(*DLFM) command results in a diagnostic message if the DLFM server has already been started.

***VPN:** The Virtual Private Network (VPN) server is started. Subsequent use of the STRTCPSVR SERVER(*VPN) command results in a diagnostic message if the VPN server has already been started.

***EDRSQL:** The Extended Dynamic Remote SQL (EDRSQL) server is started. Subsequent use of the STRTCPSVR SERVER(*EDRSQL) command results in a diagnostic message if the EDRSQL server has already been started.

***HOD:** The Host On Demand (HOD) server is started. Subsequent use of the STRTCPSVR SERVER(*HOD) command results in a diagnostic message if the HOD server has already been started.

***ODPA:** The On-Demand Platform Authentication (ODPA) server is started. Subsequent use of the STRTCPSVR SERVER(*ODPA) command results in a diagnostic message if the ODPA server has already been started.

***QOS:** The Quality of Service (QoS) server is started. Subsequent use of the STRTCPSVR SERVER(*QOS) command results in a diagnostic message if the QoS server has already been started.

***NTP:** The Simple Network Time Protocol (SNTP) server is started. Subsequent use of the STRTCPSVR SERVER(*NTP) command results in a diagnostic message if the SNTP server has already been started.

***TCM:** The Trigger Cache Manager (TCM) server is started. Subsequent use of the STRTCPSVR SERVER(*TCM) command results in a diagnostic message if the TCM server has already been started.

***DOMINO:** The Domino server is started. Subsequent use of the STRTCPSVR SERVER(*DOMINO) command results in a diagnostic message if the Domino server has already been started.

***LQP:** The Lotus QuickPlace (LQP) server is started. Subsequent use of the STRTCPSVR SERVER(*LQP) command results in a diagnostic message if the LQP server has already been started.

***WEBFACING:** The 5250 WebFacing server is started. Subsequent use of the STRTCPSVR SERVER(*WEBFACING) command results in a diagnostic message if the 5250 WebFacing server has already been started.

➤ ***DBG:** The Debug server is started. Subsequent use of the STRTCPSVR SERVER(*DBG) command results in a diagnostic message if the DBG server has already been started.

***ASFTOMCAT:** The Apache Software Foundation (ASF) Tomcat server is started. Subsequent use of the STRTCPSVR SERVER(*ASFTOMCAT) command results in a diagnostic message if the Apache Software Foundation (ASF) Tomcat server has already been started. ⏪

RESTART

Specifies whether to restart the selected server when the STRTCPSVR command is run. If the selected server is not currently running when the STRTCPSVR command is run, this parameter is ignored and the selected server is started.

The SERVER parameter specified must be *ALL, *HTTP, *DNS, *DHCP, or *QOS, or this parameter is ignored.

***NONE:** Do not restart any server.

***HTTP:** Restart the HTTP server if it is already running. This forces the HTTP server to read in the HTTP configuration and use any configuration values or attributes that changed since it was last started.

***DNS:** Restart the DNS server if it is already running. This forces the DNS server to read in the DNS configuration and use any configuration values or attributes that changed since it was last started.

***DHCP:** Restart the DHCP server if it is already running. This forces the DHCP server to read in the HTTP configuration and use any configuration values or attributes that changed since it was last started.

***QOS:** Restart the Quality of Service (QoS) server if it is already running. This forces the QoS policy rules to be reloaded from the QoS policy file.

HTTPSVR

Specifies the HTTP server instance to be started as well as any additional startup values to be used by the HTTP server to control the server instance.

If multiple HTTP server instances have been defined, you can choose to start all instances, or start one specific instance by specifying the instance name to be started.

Single Value

***ALL:** All defined server instances for the HTTP server will be started.

Element 1: Server Instance Name

server-instance-name: The specified server instance will be started.

***ADMIN:** The administration HTTP server will be started. The administration server is an instance of the HTTP server that allows administration of certain iSeries functions using a web browser.

Element 2: Instance Startup Values

***NONE:** No additional startup values are defined. The previously defined startup values for the HTTP server will be used to start the specified server instance.

instance-startup-values: Specifies additional startup values to be used for this server instance. These values will be used to override previously-defined server startup values for the specified server instance. Supported override startup values can be found in the 'Hypertext Transfer

Protocol (HTTP) Server' chapter in the TCP/IP Configuration and Reference  book.

The user is required to have *IOSYSCFG special authority to specify overrides. If instance startup values are specified and the user does not have *IOSYSCFG authority then the start request will be rejected.

DNSSVR

Specifies the DNS server instance to be started as well as any additional startup values to be used by the DNS server to control the server instance.

If multiple DNS server instances have been defined, you can choose to start all instances, or start one specific instance by specifying the instance name to be started.

Single Value

***ALL:** All defined server instances for the DNS server will be started.

Element 1: Server Instance Name

server-instance-name: The specified server instance will be started.

Element 2: Instance Startup Values

***NONE:** No additional startup values are defined. The previously defined startup values for the DNS server will be used to start the specified server instance.

instance-startup-values: Specifies additional startup values to be used for this server instance. These values will be used to override previously-defined server startup values for the specified server instance.

The user is required to have *IOSYSCFG special authority to specify overrides. If instance startup values are specified and the user does not have *IOSYSCFG authority then the start request will be rejected.

TCMSVR

Specifies the TCM instance name to be started as well as any additional startup values to be used by the TCM server to control the instance.

If multiple TCM instance names have been defined, you can choose to start all instances, or start one specific instance by specifying the instance name to be started.

Single Value

***NONE:** No defined instances for the TCM server will be started.

Element 1: Instance Name

instance-name: The specified instance will be started.

Element 2: Instance Startup Values

***NONE**: No additional startup values are defined. The previously defined startup values for the TCM server will be used to start the specified instance name.

instance-startup-values: Specifies additional startup values to be used for this instance. These values will be used to override previously-defined server startup values for the specified instance name.

The user is required to have *IOSYSCFG special authority to specify overrides. If instance startup values are specified and the user does not have *IOSYSCFG authority then the start request will be rejected.

» TOMCATSVR

Specifies the Tomcat server instance name to be started.

If multiple Tomcat server instance names have been defined, you can choose to start all instances, or start one specific instance by specifying the instance name to be started.

***NONE**: No defined instances for the Tomcat server will be started.

server-instance-name: The specified instance will be started.

***ALL**: All defined server instances for the Tomcat server will be started.

The user is required to have *IOSYSCFG special authority to specify overrides. If instance startup values are specified and the user does not have *IOSYSCFG authority then the start request will be rejected. «

Examples for STRTCPSVR

Example 1: Starting All TCP/IP Servers with AUTOSTART(*YES)

```
STRTCPSVR SERVER(*AUTOSTART)
```

This command starts all of the TCP/IP application servers that have the AUTOSTART attribute in the application configuration set to *YES.

Example 2: Starting All TCP/IP Servers

```
STRTCPSVR SERVER(*ALL)
```

This command starts all of the TCP/IP application servers that have been configured. For example: If the Change FTP Attributes (CHGFTP) command was previously used to configure two FTP servers, both servers are started when STRTCPSVR is issued. This example is also true for other TCP/IP application servers.

Where appropriate, the number of servers to start is based on the number of servers configured for the server being started.

Example 3: Starting the TELNET Server

```
STRTCPSVR SERVER(*TELNET)
```

This command starts the TCP/IP TELNET application server. If the TELNET server was previously started, one additional TELNET server job is started.

Example 4: Restarting the HTTP Server

```
STRTCPSVR SERVER(*HTTP) RESTART(*HTTP)
```

This command restarts the TCP/IP HTTP application server for all instances of the HTTP server. If the HTTP server was not currently running, then all defined instances of the HTTP server would be started.

Example 5: Starting an HTTP Server Instance

```
STRTCPSVR SERVER(*HTTP) HTTPSVR('http1')
```

This command starts the TCP/IP HTTP application server instance named 'http1' using the startup values previously defined for this server instance.

Example 6: Specifying Startup Values for an HTTP Instance

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(HTTP1 '-p 81 -sslport 443')
```

This command starts the TCP/IP HTTP application server instance named HTTP, and specifies that the server instance should listen on port 81 for unsecure requests and on port 443 for secure requests. The ports defined here will override any previously defined ports to be used by this server instance.

Example 7: Starting a DNS Server Instance

```
STRTCPSVR SERVER(*DNS) DNSSVR('dns1')
```

This command starts the TCP/IP DNS application server instance named 'dns1' using the startup values previously defined for this server instance.

Example 8: Specifying Startup Values for a DNS Instance

```
STRTCPSVR SERVER(*DNS) DNSSVR(DNS1 '-d 5')
```

This command starts the TCP/IP DNS application server instance named DNS1, and specifies that the server instance should turn on debug level 5.

Error messages for STRTCPSVR

***ESCAPE Messages**

CPF3E30

Errors occurred starting DDM TCP/IP server.

CPF3E31

Unable to start DRDA/DDM TCP/IP server.

TCP1A0C

TCP/IP LPP must be installed or at correct version level.

TCP1A11

&1 failed.

TCP1A21

STRTCPSVR not allowed. TCP/IP is not active.

STRTISSN (Start Technical Information Exchange Session) Command Description

STRTISSN Command syntax diagram

Purpose

The Start Technical Information Exchange Session (STRTISSN) command establishes the communications link for a TIE batch session. This command must precede other TIE batch commands.

Required Parameters

SPTUSRID

Specifies the user ID used to sign on the remote support network.

SPTPWD

Specifies the password used to sign on the remote support network.

Optional Parameter

ACCOUNT

Specifies the network account number used to sign on the remote support network. If the account number is not specified, the account number from the contact database is used.

***RTV:** The account number from the contact database is used.

account-number: Specify the account number that is used to sign on the remote support network.

Example for STRTIESSN

```
STRTIESSN SPTUSRID(ACME) SPTPWD(11111)
ACCOUNT(11420880)
```

This command displays the TIE main menu for account number 11420880.

No error messages.

STRTRC (Start Trace) Command Description

STRTRC Command syntax diagram

Purpose

» The Start Trace (STRTRC) command starts traces of original program model (OPM) programs, Integrated Language Environment (ILE) procedures and JAVA (compiled and JIT). Tracing can be done for multiple jobs using this command. Any number of trace sessions can be started, but active trace session identifiers must be unique across the system. This command can trace call-return flow, data returned by trace points defined in the operating system, component trace information or all three. «

The trace session continues until ended with the End Trace (ENDTRC) command. A trace session can be ended from the same job or a different job.

Restrictions:

1. To use this command you must have *SERVICE special authority, or be authorized to the Service Trace function of the operating system through iSeries Navigator's Application Administration support. The Change Function Usage Information (QSYCHFUI) API, 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.
2. A job can be part of only one active trace session. Once the trace session ends (ENDTRC command), the job can be part of another trace session.
3. Only one generic job specification can be active across all active trace sessions on your system. Once the trace ends that included a generic job specification, another trace session can be started specifying a generic job name.

Required Parameter

SSNID

Specifies a session identifier for this trace. This identifier must be unique across all active traces on the system.

***GEN:** The system will generate a unique session identifier for this trace.
session-identifier: Specify the session identifier for this trace.

Optional Parameters

JOB Specifies which jobs are to be traced.

Single Value

*****: Only the job that issues the STRTRC (Start Trace) command is traced.

» Element 1: Job Identifier «

Job Name Qualifier

job-name: Specify the name of the job to be traced. Up to eight job names may be specified, but only one generic job name is allowed.

generic-job-name:* Specify the generic name of the job to be traced. 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 job name specifies all jobs with job names that begin with the generic prefix. Only one active trace session can have a generic job specification.

Job User Name Qualifier

user-name: Specify the name of the user of the job to be traced.

Job Number Qualifier

***ALL:** All jobs with the specified job name and user name are traced. *ALL for the job number is considered to be a generic job specification because it will trace all jobs that meet the job name and job user name qualifiers that you specified. Only one active trace session can have a generic job specification.

job-number: Specify the job number to further qualify the job name and user name. You cannot specify a job number if a generic job name qualifier is specified.

» Element 2: Thread ID to Include

Specifies a list of up to twenty threads in the job whose information is to be included.

***ALL:** All threads within the specified job name and user name are traced

***INITIAL:** Only the initial thread within the specified job name and user name is traced.

***SELECT:** A list of thread identifiers is shown from which the user can select up to twenty to be traced. This value is only valid if the command is run in an interactive job.

thread-identifier: Specify the identifiers of up to twenty threads whose information is to be included.



MAXSTG

Specifies the maximum amount of storage, in kilobytes (K), to use for the collected trace records. The system will calculate the minimum amount of storage that is necessary for the trace; this minimum storage size calculation is dependent upon the system's processor configuration. The minimum amount of storage may be significantly larger than the size specified on the MAXSTG parameter; the system will use the larger of the two values.

10000: Up to 10,000 kilobytes of storage is used.

maximum-K-bytes: Specify the maximum amount of storage used to store trace records (one kilobyte equals 1024 bytes).

Note: The maximum amount of storage is 4,000,000 kilobytes.

TRCFULL

Specifies whether the trace records wrap (replace the oldest records with new records) or whether the trace stops when all of the storage specified by the MAXSTG parameter has been used.

***WRAP:** When the trace storage is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

***STOPTRC:** Tracing stops when the trace storage is full of trace records.

JOBTRCTYPE

Specifies the types of job trace data to be stored in the trace file.

***ALL:** All the trace data collected is stored in trace records. This includes tracing the flow of control and the trace data itself.

***FLOW:** The flow of control is traced when OPM programs and ILE procedures are called and when they return.

***DATA:** The data that is provided at predefined trace points within the operating system is stored in trace records. This includes trace records for the CL commands that have run.

➤ ***TRCTYPE:** Trace specific components specified in the trace type parameter.

TRCTYPE

Specifies the component to trace and the level to use.

***NONE:** No component trace is specified

Element 1: Trace Type Component

Each trace type is identified by a special value. For a complete list of components, position the cursor on the field for the **Trace type** prompt (TRCTYPE parameter), and press F4. Specify up to 50 trace components from the following table

Table 1. Trace Type Table

Trace Description	Special Value
ASP Management	*ASPMGT
Directory Services	*DIRSRV
Environment Variable APIs	*ENVVAR
HTTP Server powered by Apache	*HTTP
Interprocess Communication APIs	*IPC
Lock Space Management	*LOCKSPACE
POSIX Miscellaneous APIs	*POSIXMISC
POSIX Process-Related APIs	*POSIXPROC
POSIX Thread APIs	*PTHREAD
Qshell	*QSHELL
Software Security	*SECURITY
Sockets APIs	*SOCKETS
Thread Management	*THREADMGT

Element 2: Trace Level

Specifies the trace level to be used for the specified component

***ERROR:** The trace level ERROR will be used. Components typically trace error return codes, exception conditions, and invalid input. The amount of data is usually small.

***INFO:** The trace level INFO will be used, which also includes ERROR trace level data. Components typically trace entry and exit from interfaces, parameters and return codes, and major changes of flow or semantics caused by input or other decisions. The amount of data is moderate.

***VERBOSE:** The trace level VERBOSE will be used, which also includes INFO and ERROR trace level data. Components typically trace detailed data that could assist in debugging control flow, data corruption, data structures, environment, call stacks, and resource allocations. The amount of data can be large.

TRCFTR

Specifies the trace filter to be used. The Add Trace Filter (ADDTRCFTR) command must be issued to define the trace filter. The filter determines what information is collected in the trace based on the filter values. The filter values are compared to the actual trace data. If a filter is not specified, then all trace information is collected.

The trace filter parameter can only be specified if *ALL or *FLOW has been specified on the job trace type parameter.

Note: The trace filter applies to the *FLOW (call/return) trace only.

***NONE:** No trace filter will be used.

trace_filter_name: The name of the trace filter created with ADDTRCFTR command. <<

Examples for STRTRC

Example 1: Start a Trace on Your Job

```
STRTRC SSNID(*GEN)
```

This command starts a trace on the job that called the STRTRC command. A unique trace session identifier will be generated. The session identifier will be returned in the message data of a completion message sent after the trace session starts successfully.

Example 2: Start a Trace of Another Job

```
STRTRC JOB(123456/QSYS/QSYSARB) SSNID(ARBTRACE)
```

This command starts a new trace on the job with job name QSYSARB, job user name of QSYS, and job number of 123456. Only this one job is traced. The trace session identifier is ARBTRACE.

Example 3: Start a Generic Job Trace

```
STRTRC SSNID(*GEN) JOB(*ALL/QSYS/QTVTELNET*)  
MAXSTG(80000)
```

This command starts a new trace on all jobs on the system that have a job name that begins with QTVTELNET and are running under the QSYS user profile. The maximum storage used for the trace will be 80000 kilobytes. A unique session identifier will be generated.

>> Example 4: Start a Job Trace with a Filter

```
ADDTRCFTR FTR(MYFTR) PGMTRG(*EQ MYLIB/MYPGM *ALL *PGM *ENTRY)
STRTRC SSNID(MYTRACE)JOB(123456/MYUSER/MYJOB)
TRCFTR(MYFTR)
```

These commands add a trace filter for MYPGM in MYLIB. When the trace is started, call/return trace records will not be collected until MYLIB/MYPGM is called. Once MYLIB/MYPGM is called, the trace will collect call/return trace records until the trace is ended.

Example 5: Start a Job Trace with Component Defined

```
STRTRC JOB(123456/MYUSER/MYJOB) SSNID(MYTRACE)
TRCTYPE(*ENVVAR *VERBOSE)
```

This command starts a new trace on the job with job name MYJOB, job user name of MYUSER, and job number of 123456. Only this one job is traced. The trace session identifier is MYTRACE. The trace will collect call/return trace information as well as information for the ENVVAR component at VERBOSE level.

Example 6: Start a Job Trace with Threads Selection

```
STRTRC JOB((123456/MYUSER/MYJOB (11111111 22222222))) SSNID(MYTRACE)
```

This command starts a new trace on the job with job name MYJOB, job user name of MYUSER, and job number of 123456. Only threads 11111111 and 22222222 are traced. The trace session identifier is MYTRACE. <<

Error messages for STRTRC

***ESCAPE Messages**

CPC3921

STRTRC session ID &1 successfully started.

CPC3922

STRTRC session ID &1 started but some jobs not found.

CPF39C6

A generic job STRTRC session is already active.

CPF39C7

STRTRC session ID &1 already exists.

CPF39C8

STRTRC job name conflict with session ID &1.

CPF39C9

Unexpected STRTRC failure, see low-level messages.

CPF39C5

Only one generic job name or job number allowed.

CPF39CC

No active jobs found, STRTRC session not started.

CPI36CE

Job &3/&2/&1 not found.

STRTRPMGR (Start Trap Manager) Command Description

STRTRPMGR Command syntax diagram

Purpose

The Start Trap Manager (STRTRPMGR) command allows you to start the OS/400 SNMP Manager Framework trap manager job. An optional Forward Trap parameter may be specified which enables traps that are received on other systems to be forwarded to other Network Management stations. The trap manager uses the trap generation and sending facilities provided in the Simple Network Management Protocol (SNMP) agent and Distributed Protocol Interface (DPI) interface.

Optional Parameter

FWDTRP

Specifies whether traps received on the system are to be forwarded to other network management stations.

***YES:** Received traps are forwarded using the facilities provided in the SNMP agent and DPI interface.

***NO:** Received traps are not forwarded. Traps are only enqueued.

Examples for STRTRPMGR

Example 1: Start Trap Manager Job (Enqueue Traps Only)

```
STRTRPMGR
```

This command starts the trap manager job. Traps received by the trap manager are enqueued only.

Example 2: Start Trap Manager Job (Enqueue & Forward Traps)

```
STRTRPMGR FWDTRP(*YES)
```

This command starts the trap manager job. Traps received by the trap manager are enqueued and forwarded.

Error messages for STRTRPMGR

*ESCAPE Messages

CPFA80D

Trap manager job already active.

CPFA802

Trap manager not started.

STRUSF (Start Ultimedia System Facilities) Command Description

STRUSF Command syntax diagram

Purpose

The Start Ultimedia System Facilities (STRUSF) command starts the Ultimedia Facilities server and router programs, QUMBVSEVR and QUMBVROUTR, in the QSYSWRK subsystem of the operating system. All associated multimedia devices are varied on when this command is run.

Optional Parameters

LOG Specifies whether messages are sent to the message queue, QUMBVMSQ in library QUMEDIA, to log Ultimedia Facilities activity.

***NO:** The activity is not logged. Messages are sent and activity is logged only if severe errors occur.

***YES:** The activity is logged.

CLRMSGQ

Specifies whether the message queue, QUMBVMSQ in library QUMEDIA, is cleared before starting the Ultimedia Facilities program.

***YES:** The message queue is cleared.

***NO:** The message queue is not cleared.

Example for STRUSF

```
STRUSF CLRMSGQ(*NO)
```

This command starts the Ultimedia System Facilities program. The message queue, QUMBVMSQ in library QUMEDIA, is not cleared when the program is started.

No error messages.



Printed in U.S.A.