



stonebranch

Universal Agent 7.4.x

Universal Data Mover 7.4.x Quick Reference Guide

© 2023 by Stonebranch, Inc. All Rights Reserved.

Universal Data Mover 7.4.x Quick Reference Guide

Command Format

command [**parameter**[=**value**]]...

appenddata **_data_element_name_** [**_value_1_**]... [**_value_n_**]

| Parameter | Description |
|--------------------------|---|
| <i>data_element_name</i> | Name of a data element. |
| <i>value_1</i> | First value in the line of text to be appended. |
| <i>value_n</i> | Last value in the line of text to be appended. |

attrib **_logical-name_**=[{***dd*** | ***dsn*** | ***hfs*** | ***lib***}] [**attribute-name=attribute-value**]...

| Parameter | Description |
|--|--|
| <i>logical-name</i> | Logical name of the transfer server to which the attribute(s) applies. |
| dd dsn hfs lib | File system for which the attribute(s) applies. |
| <i>attribute-name</i> | Name of an attribute. |
| <i>attribute-value</i> | Value to be set for the attribute. |

break

(There are no parameters used with this command.)

call **script-file** [**parameter-name=parameter-value**]...

| Parameter | Description |
|------------------------|--|
| <i>script-file</i> | Name of the script file to execute. |
| <i>parameter-name</i> | Name of a parameter to pass to the script. |
| <i>parameter-value</i> | Value to be set for the parameter. |

cd **_logical-name_** [**=_directory_**]

| Parameter | Description |
|---------------------|---|
| <i>logical-name</i> | Logical name of the transfer server to execute the cd command. |
| <i>directory</i> | Working directory to change to on the server. |

close

(There are no parameters used with this command.)

closelog

(There are no parameters used with this command.)

compare STRING_1 STRING_2 [case=*yes* | *no*] [length=*length*]

| Parameter | Description |
|-------------------------------|---|
| STRING_1 | String to be compared to STRING_1. |
| STRING_2 | String to be compared to STRING_2. |
| case = <i>yes</i> <i>no</i> | Specification for whether or not the comparison is case-sensitive. If set to <i>yes</i> , comparison is case-sensitive. If set to <i>no</i> , comparison is not case-sensitive. (If the case parameter is not used, the comparison is case insensitive.) |
| length= <i>length</i> | First <i>n</i> characters to be compared. |

copy source-logical-name=*file-specification* [destination-logical-name=*file specification*]

| Parameter | Description |
|--------------------------|---|
| source-logical-name | Logical name of the source transfer server in the copy operation. |
| file-specification | File specification for the source file. |
| destination-logical-name | Logical name of the destination transfer server in the copy operation. |
| file-specification | File specification for the destination file. |

copydir source-logical-name=*file-specification* [destination-logical-name=*file-specification*]

| Parameter | Description |
|--------------------------|--|
| source-logical-name | Logical name of the source transfer server in the copy operation. |
| file-specification | File specification for the source file directory. |
| destination-logical-name | Logical name of the destination transfer server in the copy operation. |
| file-specification | File specification for the destination file or directory. |

data [name|print=*name*] [resolve={*all*|*defined*|*no*}] [*data-element*] [end=*end-sequence*]

| Parameter | Description |
|---|--|
| name print= <i>name</i> | Specifies either the name of the in-stream data element being defined (<i>name</i>) or a request to print the lines of that data element. |
| resolve={ <i>all</i> <i>defined</i> <i>no</i> } | Variable resolution method: <ul style="list-style-type: none"> all: Resolve all variable references in the data. defined: Resolve only defined variable references in the data. no: Do not resolve any variable references in the data. |
| <i>data-element</i> | Contents of the in-stream data element being defined. |
| end= <i>end-sequence</i> | Sequence indicating the end of the data. |

debug [EXPRESSION_SHOW_POSTFIX=*yes*|*no*] [EXPRESSION_SHOW_EVALUATION=*yes*|*no*] [COMMAND_SHOW_STRUCTURE=*yes*|*no*]

| Parameter | Description |
|--|---|
| EXPRESSION_SHOW_POSTFIX= <i>yes</i> <i>no</i> | Postfix version of an expression after it has been converted from infix notation. |
| EXPRESSION_SHOW_EVALUATION= <i>yes</i> <i>no</i> | Evaluation of an expression. |
| COMMAND_SHOW_STRUCTURE= <i>yes</i> <i>no</i> | Different elements of a command after it has been parsed. |

delete logical-name=file-specification

| Parameter | Description |
|--------------------|---|
| logical-name | Logical name of the server on which to delete the file(s). |
| file-specification | File(s) to be deleted; single filename or a complete path to a file or directory. The filename (or filename portion of the path) can contain wildcard characters: <ul style="list-style-type: none"> • Wildcard * represents zero or more characters. • Wildcard ? indicates a single character. |

deletestring variable_name {pos=position} | {startseq=sequence [startseqnum=number] } {length=length} | {endseq=sequence [endseqnum=number] } {case=yes|no}

| Parameter | Description |
|--------------------|---|
| variable-name | Name of an existing variable. |
| pos=position | Starting position of sequence to be deleted (one-based index). |
| startseq=sequence | Starting position of sequence to be deleted (following a specific character sequence). |
| startseqnum=number | Occurrence number of starting position of sequence to be deleted (following a specific character sequence). |
| length=length | Length of the substring to be deleted. |
| endseq=sequence | Ending position of sequence to be deleted (preceding a specific character sequence). |
| endseqnum=number | Occurrence number of ending position of sequence to be deleted (preceding a specific character sequence). |
| case=yes no | Indicates whether or not the comparisons of the sequence are case-insensitive. |

diskspace logical_name [=path] | [qty=nnn{b|k|m|g}] [cond=GT|LT]

| Parameter | Description |
|------------------|--|
| logical-name | Name of the system where UDM checks available space. If path is specified, diskspace checks the volume or file system where that directory resides. Otherwise, the current working directory is used.. |
| qty=nnn{b k m g} | Amount of space that diskspace should look for. To specify a unit for the quantity, add one of the following letters to the quantity amount: <ul style="list-style-type: none"> • b - bytes • k - kilobytes (1,024 bytes) • m - megabytes (1,048,576 bytes) • g - gigabytes (1,073,741,824 bytes) If no unit is specified, the default (b - bytes) is used. If no quantity is specified, diskspace displays the amount of available space. |
| cond=GT LT | Instructs diskspace to check for an amount that is greater than (GT) or less than (LT) the specified quantity. This parameter is optional. If no condition is specified, GT is used. |



echo [parm_1]... [parm_n]

| Parameter | Description |
|-----------|-----------------------------|
| parm_1 | First parameter to echo. |
| parm_n | Other parameter(s) to echo. |



echolog [*value_1*]... [*value_n*]

| Parameter | Description |
|----------------|-----------------------------|
| <i>value_1</i> | First parameter to echo. |
| <i>value_n</i> | Other parameter(s) to echo. |

exec *logical-name*|*host-name* {*cmd=command*| *cmdref=command-ref* *stc=started-task*} [*user=user-id*] [*pwd=password*] [*port=port*] [*codepage=codepage*] [*file=filename*] [*xfile=filename*] [*key=key*]] [*option=option*] [*mergelog=yes|no*] [*trace=yes|no*] [*input=data-element*] [*svropt=server-options*] [*stdout=data-element*] [*stderr=data-element*]

| Parameter | Description |
|------------------------------|---|
| <i>logical-name</i> | Logical name of the server on which to execute the command. |
| <i>host-name</i> | Host name of the server on which to execute the command. |
| <i>cmd=command</i> | Name of command to execute of command type cmd . |
| <i>cmdref=command-ref</i> | Name of command reference to execute of command type cmdref . |
| <i>stc=started-task</i> | Name of started task to execute of command type stc . |
| <i>user=user-id</i> | User ID with which the command is to be executed. |
| <i>pwd=password</i> | Password for the specified user ID. |
| <i>port=port</i> | Port that the broker is listening on for the machine the command is to be executed on. |
| <i>codepage=codepage</i> | Codepage used for executing the command. |
| <i>file=filename</i> | Plain text file containing the options for the remote execution server. These options include: port , user , pwd , and codepage .  Note The options in this file override any corresponding options specified for the transfer server in the open command. |
| <i>xfile=filename</i> | Universal Encrypted text file containing the options for the remote execution server. These options include: port , user , pwd , and codepage .  Note The options in this file override any corresponding options specified for the transfer server in the open command. |
| <i>key=key</i> | Key used to decrypt an encrypted option file. |
| <i>option=option</i> | Options to be passed to the UCMD Server. |
| <i>mergelog=yes no</i> | Specification for whether or not to merge standard out and standard error output streams from a remote command to the UDM transaction log: <ul style="list-style-type: none"> • yes: Merge data streams in UDM transaction log. • no: Do not merge data; keep as separate data streams. |
| <i>trace=yes no</i> | Specification for whether or not trace functionality is used: <ul style="list-style-type: none"> • yes: Tracing turned on in the call to UCMD. • no: Tracing turned off in the call to UCMD. |
| <i>input=data-element</i> | Name of the data element that is used as input to the command being executed. |
| <i>svropt=server-options</i> | Options used to override Universal Command Server options. |
| <i>stdout=data-element</i> | Data element to be used as standard out from the remote command. |
| <i>stderr=data-element</i> | Data element to be used as standard error from the remote command. |

execsap [host=*host-name*] destination type=**event|generic** [eventid=*event-id*] [parm=*event-parm*] [client=*client*] [user=*user-id*] [pwd=*password*] [codepage=*codepage*] [file=*filename*] xfile=*filename* [key=*key*]] [mergelog=**yes|no**] [trace=**yes|no**]

| Parameter | Description |
|----------------------------------|---|
| host | SAP destination entry. |
| destination | Destination in the SAP RFC file. |
| type=event | Type of SAP operation to perform (event is the only valid type). |
| eventid | ID of the SAP event to trigger. |
| parm | Parameter to the SAP operation. |
| client | Name of the SAP client. |
| user= <i>user-id</i> | User ID with which the command is to be executed. |
| pwd= <i>password</i> | Password for the specified user ID. |
| codepage= <i>codepage</i> | Codepage used for executing the command. |
| file= <i>filename</i> | Plain text file containing the options for the remote execution server. These options include: port , user , pwd , and codepage .  Note The options in this file override any corresponding options specified for the transfer server in the open command. |
| xfile= <i>filename</i> | Universal Encrypted text file containing the options for the remote execution server. These options include: port , user , pwd , and codepage .  Note The options in this file override any corresponding options specified for the transfer server in the open command. |
| key= <i>key</i> | Key used to decrypt an encrypted option file. |
| mergelog= yes no | Specification for whether or not to merge standard out and standard error output streams from a remote command to the UDM transaction log: <ul style="list-style-type: none"> • yes: Merge data streams in UDM transaction log. • no: Do not merge data; keep as separate data streams. |
| trace= yes no | Specification for whether or not trace functionality is used: <ul style="list-style-type: none"> • yes: Tracing turned on in the call to UCMD. • no: Tracing turned off in the call to UCMD. |

exit

(There are no parameters used with this command.)

filesys logical-name=[{**dd|dsn|hfs|lib**}]

| Parameter | Description |
|--|--|
| <i>logical-name</i> | Logical name of the transfer server on which to change the file system. |
| dd dsn hfs lib | File system to set for the specified server: <ul style="list-style-type: none"> • dd: ddnames defined with JCL DD statements (z/OS only) • dsn: data set name (z/OS only) • hfs: UNIX System Services file system (z/OS and IBM i only) • lib: LIB file system (IBM i only) |

filetype [**binary**|**text=***filemask_1*]...[**binary**|**text=***filemask_n*] [**remove=***filemask_1*]...[**remove=***filemask_n*] [**resetall**]

| Parameter | Description |
|---|---|
| binary text= <i>filemask_1</i> | Sets the specified file mask (<i>filemask_1</i>) so that source file names matching this mask will be transferred as binary files. |
| binary text= <i>filemask_n</i> | Sets the specified file mask (<i>filemask_n</i>) so that source file names matching this mask will be transferred as text files. |
| remove= <i>filemask_1</i> | Removes an entry that matches the specified file mask (<i>filemask_1</i>). |
| remove= <i>filemask_n</i> | Removes an entry that matches the specified file mask (<i>filemask_n</i>). |
| resetall | Removes all file mask entries. |

find string seq=*sequence* [**pos=***index*] [**case=****yes**|**no**] [**num=***number*|**last**]

| Parameter | Description |
|----------------------------------|---|
| <i>string</i> | String in which to search for the sequence. |
| seq= <i>sequence</i> | Sequence for which the search is being made. |
| pos= <i>index</i> | One-based index of the string where the find operation begins. |
| case= yes no | Specification of whether the search is case-sensitive (yes) or case-insensitive (no). [Default is no .] |
| num= <i>number</i> last | Instance of the sequence for which the search is being made. |

format*variable_name* [{*string_1*|*expression_1*} [**align=**{**center**|**left**|**right**|**justify**}] [**pad=***sequence*] [**trunc=****yes**|**no**] [**length=***length*]]...[{*string_n*|*expression_n*} [**align=****center**|**left**|**right**|**justify**] [**pad=***sequence*] [**trunc=****yes**|**no**] [**length=***length*]]

| Parameter | Description |
|--|---|
| <i>variable_name</i> | Variable or list element in which the newly formatted string is stored |
| <i>string_1</i> <i>expression_1</i> | First section of the string |
| align= center left right justify | Method of how the value is aligned within the given space. [Default alignment is left.] |
| pad= <i>sequence</i> | Character(s) used to pad out the formatted value. [Default is space character.] |
| trunc= yes no | Truncates the value to the length given by length if it is longer than that value. [Default is to not truncate.] |
| length= <i>length</i> | Length of the formatted field. [Default is length of the value specified.] |

insertstring*variable_names*sequence {**pos=***position*} startseq=*sequence* startseqnum=*number*} [**case=****yes**|**no**]

| Parameter | Description |
|------------------------------|---|
| <i>variable_name</i> | Name of the existing variable or list element into which to insert the sequence. |
| <i>sequence</i> | Sequence to be inserted. |
| pos= <i>position</i> | Position at which sequence is inserted. |
| startseq= <i>sequence</i> | Sequence after which specified sequence is inserted. |
| startseqnum= <i>number</i> | Instance of sequence after which specified sequence is inserted. |
| case= yes no | Specification of whether or not the comparison used to find the start sequence is case-insensitive. [Default is no.] |

loaddata *data-element-name=file-path*

| Parameter | Description |
|--------------------------|---|
| <i>data-element_name</i> | Data element into which the file contents is loaded. |
| <i>file_path</i> | File from which contents is loaded into the data element. |

logdata *data_element_name*

| Parameter | Description |
|--------------------------|--|
| <i>data_element_name</i> | Name of the data element to write to the open log. |

lower *variable_name*

| Parameter | Description |
|----------------------|---|
| <i>variable_name</i> | Variable or list element in which to force alphabetic characters to lower case. |

mode [type={**text**|**binary**}] [trim={ **yes**|**no**}]


| Parameter | Description |
|-----------------------------------|---|
| type= text binary | Type of transfer mode to be used: <ul style="list-style-type: none"> • text: UDM sends source data as is. • binary: UDM performs text translation on data. [Default is binary when a new session is opened.] |
| trim= yes no | Trim trailing spaces during a text transfer: <ul style="list-style-type: none"> • yes: Spaces at the end of each line (record) are removed from the data when it is written. • no: Spaces at the end of each line (record) are not removed. [Default is no when a transfer session is initiated.] |

move *source-logical-name=source-file-specification* [*destination-logical-name=destination-file-specification*]

| Parameter | Description |
|---------------------------------|--|
| <i>source-logical-name</i> | Logical name of the server acting as the source of the move operation. |
| <i>file-specification</i> | File(s) to be copied. It can be a single file name or a complete path. The file name (or file name portion of the path) can contain wildcard characters: <ul style="list-style-type: none"> • Wildcard * represents for zero or more characters. • Wildcard ? indicates a single character. |
| <i>destination-logical-name</i> | Logical name of the destination server in the move operation. |
| <i>file-specification</i> | Complete path or file name for the destination file. |

open [[primary ={*|**local**|*hostname*} [port=*port*] [user=*username* [pwd=*password*]] [codepage=*codepage*] [{file=*filename*| xfile=*filename* [key=*key*] }] secondary =*hostname* [port=*port*] [user=*username* [pwd=*password*]] [codepage=*codepage*] [file=*filename*] [xfile=*filename* [key=*key*]] [encrypt={**yes**|**no**|*cipher*}] [compress={**yes**|**no**|**hasp**|**zlib**}] [nft=**yes**|**no**] [comment=*text*] [tt=**direct**|**deferred**]

| Parameter | Description |
|--|---|
| primary=* local <i>hostname</i> | Logical name of the primary transfer server: <ul style="list-style-type: none"> • local: Manager is the primary server in a two-party transfer |

| | |
|--|--|
| | <ul style="list-style-type: none"> • <i>hostname</i>: Host name or IP address of a transfer server |
| <i>port=port</i> | Port that the Universal Broker is listening on that will initiate the primary transfer server. |
| <i>user=username</i> | Name of the user to authenticate on the primary transfer server. |
| <i>pwd=password</i> | Password used to authenticate the specified user. |
| <i>codepage=codepage</i> | Codepage to use for text translation by the primary transfer server. |
| <i>file=filename</i> | Plain text file that can contain open options for the primary transfer server. |
| <i>xfile=filename</i> | Universal Encrypted file that can contain open options for the primary transfer server. |
| <i>key=key</i> | Encryption key used to decrypt the specified encrypted open options file. |
| <i>secondary=hostname</i> | <p>Logical name of the secondary transfer server. <i>hostname</i> is the Host name or IP address of the machine on which the server will be running.</p> <p> Note Host name must be given from the perspective of the primary transfer server, not the UDM Manager.</p> |
| <i>port=port</i> | Port that the Universal Broker is listening on that will initiate the secondary transfer server. |
| <i>user=username</i> | Name of the user to authenticate on the secondary transfer server. |
| <i>pwd=password</i> | Password used to authenticate the specified user. |
| <i>codepage=codepage</i> | Codepage to use for text translation by the secondary transfer server. |
| <i>file=filename</i> | Plain text file that can contain open options for the secondary transfer server. |
| <i>xfile=filename</i> | Universal Encrypted file that can contain open options for the secondary transfer server. |
| <i>key=key</i> | Encryption key used to decrypt the specified encrypted open options file. |
| <i>encrypt=yes no cipher</i> | <p>Encryption method for the transfer session:</p> <p>yes: Use cipher based on list in the -data_ssl_cipher_list option.</p> <ul style="list-style-type: none"> • no: Use NULL-MD5 cipher. • cipher: Specific cipher to use. |
| <i>compress=yes no hasp zlib</i> | <p>Data compression method for the transfer session:</p> <p>yes: Use method as specified in the -compress option.</p> <ul style="list-style-type: none"> • no: Do not use data compression. • hasp: Use HASP compression. • zlib: Use ZLIB compression. |
| <i>nft=yes no</i> | <p>Specification for network fault tolerance:</p> <ul style="list-style-type: none"> • yes: Session is network fault tolerant. • no: Session is not network fault tolerant. |
| <i>comment=text</i> | Comment for a single session (or overrides a comment specified by the COMMENT option). |
| <i>tt= direct deferred</i> | <p>Specification for where codepage translation is performed for files transferred in text mode:</p> <ul style="list-style-type: none"> • direct: codepage translation will be performed in the component sending the file - prior to transmission. • deferred: codepage translation will be performed in the component receiving the file - after transmission. |

openlog*log_file_path* [append=**yes|no**]

| Parameter | Description |
|-----------------------|--|
| <i>log_file_path</i> | Path name of the log file to open. |
| append= yes no | Specification for whether or not to append log statements to an existing log file. |

pad*variable_name* length=*length* [seq=*sequence*]

| Parameter | Description |
|-----------------------|--|
| <i>variable_name</i> | Name of an existing variable or list element to pad. |
| length= <i>length</i> | Desired length of the padded string. |
| seq= <i>sequence</i> | Sequence used for padding the string. |

parse *variable_name* length=*length* [seq=*sequence*]

| Parameter | Description |
|--------------------|---|
| <i>string_name</i> | String to parse. |
| <i>var_1</i> | First component of the parsed string placed into a variable. |
| <i>seq_1</i> | Sequence of the string serving as a delimiter between <i>var_1</i> and <i>var_2</i> . |
| <i>var_2</i> | Second component of the parsed string placed into a variable. |
| <i>seq_n</i> | Sequence of the string serving as a delimiter between the preceding <i>var_n</i> and the following <i>var_n</i> . |
| <i>var_n</i> | Next component of the string after the preceding <i>seq_n</i> . |

print msg=*message*

| Parameter | Description |
|---------------------|--|
| msg= <i>message</i> | Message that should be printed by the UDM Manager in its transaction output. |

query

(There are no parameters used with this command.)

quit

(There are no parameters used with this command.)

rename *logical-name* *old-filename* *new-filename*

| Parameter | Description |
|---------------------|---|
| <i>logical-name</i> | Logical name of the transfer server on which you want to rename a file. |
| <i>old-filename</i> | Current name (filename, path, or absolute path) for the file that you want to rename. |
| <i>new-filename</i> | New name (filename, path, or absolute path) to which the file is to be renamed. |

replace *variable_name* *oldsequence* *newsequence* [num=*index*] [all=**yes**|**no**] [case=**yes**|**no**]

| Parameter | Description |
|------------------------------|--|
| <i>variable_name</i> | Name of the existing variable or list element on which to perform the replace. |
| <i>oldsequence</i> | Sequence to be replaced. |
| <i>newsequence</i> | Sequence replacing <i>oldsequence</i> . |
| num= <i>index</i> | One-based sequence number that identifies the occurrence of <i>oldsequence</i> . |
| all= yes no | Specification for whether all instances (yes) of the sequence are replaced or only the first instance (no) of the sequence. |
| case= yes no | Specification for whether the comparison to match the old sequence is case sensitive (yes) or case insensitive (no). [Default is no .] |

report progress=**yes** | **no**

| Parameter | Description |
|----------------------------------|---|
| progress= yes no | Turns on or off the reporting of periodic transfer progress messages by UDM Manager during a file transfer operation. |

resetattribs *logical-name*

| Parameter | Description |
|---------------------|---|
| <i>logical-name</i> | Logical name of the transfer server whose attributes are to be reset to their default values. |

return [*return-value*]

| Parameter | Description |
|---------------------|---|
| <i>return-value</i> | Sets _rc variable with a return value to specify when to return. |

reverse [*variable_name*]


| Parameter | Description |
|----------------------|--|
| <i>variable_name</i> | String in which all characters will be reversed. |

savedata [*data_element_name=**file_spec*]

| Parameter | Description |
|--------------------------|---|
| <i>data_element_name</i> | Name of the data element. |
| <i>file_spec</i> | Name of the file on which to write each line of the data element. |

set [**_echo**={**yes**|**no**}] [**_lines**={**yes**|**no**}] [**_halton**={**none**|**warn**|**error**|**fatal**|*integer-value*}] [**_rc**={**none**|**warn**|**error**|**fatal**|*integer-value*}] [**_keepalive**=*integer-value*] [*global_name=**global_value_1*]...[*global_name_n=**global_value_n*]

| Parameter | Description |
|----------------|--|
| _echo | Each command is echoed back after it is read for processing: yes or no (default). |
| _lines | Line number of each command is reported if an invalid command is received or fails to be parsed or processed by the UDM Manager: yes or no (default). |
| _halton | Halt condition setting: if the return code of any command is greater than the value of halton , and halton is not zero, UDM issues a quit command and exits. Values are a simple integer value or one of the following predefined values: none : equivalent to a value of 0; UDM does not automatically quit, regardless of a command's return value. <ul style="list-style-type: none">• warn: equivalent to a value of 4; usually returned for invalid commands and command parse errors.• error: equivalent to a value of 8; command fails to execute successfully.• fatal: equivalent to a value of 16; unrecoverable error occurs and the UDM Manager or one of the transfer servers must exit. [Default is 0 .] |
| _rc | Internal UDM value, returned to UDM when the UDM Manager exits, that indicates the success or failure of each command executed by UDM. If the returned value is greater than the current value of rc , rc is set to that return value. Values are a simple integer value or one of the following predefined values: none : equivalent to an integer value of 0; command completed successfully. <ul style="list-style-type: none">• warn: equivalent to a value of 4; usually returned for invalid commands and command parse errors.• error: equivalent to a value of 8; command fails to execute successfully.• fatal: equivalent to a value of 16; unrecoverable error occurs and the UDM Manager or one of the transfer servers must exit. [Default is 0 .] |
| | Interval (in seconds) at which the UDM Manager sends a keep alive message. Value is a simple integer value; 0 = no keep alive message is to be sent. |

| | |
|--------------|--|
| _keepalive |  Note This value must be set BEFORE a session is established. |
| _global_name | User-defined global variable. |

status

(There are no parameters used with this command.)

strip*variable_name*sequence [num=*index*] [all=**yes**|**no**] [case=**yes**|**no**]

| Parameter | Description |
|------------------------------|--|
| <i>variable_name</i> | Name of the existing variable or list element on which to perform the strip. |
| <i>sequence</i> | Sequence to be stripped |
| num= <i>index</i> | One-based sequence number that identifies the occurrence of the sequence to be stripped. |
| all= yes no | Specification for whether all instances (yes) of the sequence are replaced or only the first instance (no) of the sequence. |
| case= yes no | Specification for whether the comparison to match the old sequence is case sensitive (yes) or case insensitive (no). [Default is no .] |

substring*variable_name*sequence [num=*index*] [all=**yes**|**no**] [case=**yes**|**no**]

| Parameter | Description |
|------------------------------|--|
| <i>variable_name</i> | Name of the existing variable into which the substring is placed. |
| <i>string</i> | String from which the substring is taken. |
| pos= <i>position</i> | Starting position of string to be taken (one based index). |
| startseq= <i>sequence</i> | Starting position of string to be taken (following a specific character sequence). |
| startseqnum= <i>number</i> | Occurrence number of starting position of string to be taken (following a specific character sequence). |
| length= <i>length</i> | Length of the string to be taken. |
| endseq= <i>sequence</i> | Ending position of string to be taken (following a specific character sequence). |
| endseqnum= <i>number</i> | Occurrence number of ending position of string to be taken (following a specific character sequence). |
| case= yes no | Specification for whether or not the comparisons of the start and end sequences are case sensitive (yes) or case insensitive (no). [Default is no .] |

truncate*variable_name* length=*length*

| Parameter | Description |
|-----------------------|--|
| <i>variable_name</i> | Name of the existing variable or list element to truncate. |
| length= <i>length</i> | Length to which the string truncates. |

upper*variable_name*

| Parameter | Description |
|----------------------|---|
| <i>variable_name</i> | Variable or list element in which to force alphabetic characters to upper case. |