

# Universal Data Mover 4.3.0

## Quick Reference

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.
<b>dd</b>   <b>dsn</b>   <b>hfs</b>   <b>lib</b>	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 <code>cd</code> 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 = <b>yes</b>   <b>no</b>	Specification for whether or not the comparison is case-sensitive <ul style="list-style-type: none"> <li>• If set to yes, comparison is case-sensitive.</li> <li>• If set to no, comparison is not case-sensitive.</li> </ul> (If the case parameter is not used, the comparison is case insensitive.)
length=length	First <b>n</b> 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 <b>copy</b> operation.
file-specification	File specification for the source file.
destination-logical-name	Logical name of the destination transfer server in the <b>copy</b> 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
<b>name</b>   <b>print=name</b>	Specifies either the <b>name</b> of the in-stream data element being defined ( <i>name</i> ) or a request to <b>print</b> the lines of that data element.
resolve={ <b>all</b>   <b>defined</b>   <b>no</b> }	Variable resolution method: <ul style="list-style-type: none"> <li>• <b>all</b> Resolve all variable references in the data.</li> <li>• <b>defined</b> Resolve only defined variable references in the data.</li> <li>• <b>no</b> Do not resolve any variable references in the data.</li> </ul>
data-element	Contents of the in-stream data element being defined.
end=end-sequence	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=yes   no	Postfix version of an expression after it has been converted from infix notation.
EXPRESSION_SHOW_EVALUATION=yes   no	Evaluation of an expression.
COMMAND_SHOW_STRUCTURE=yes   no	Different elements of a command after it has been parsed.

delete *logical-name=file-specification*

Parameter	Description
<i>logical-name</i>	Logical name of the server on which to delete the file(s).
<i>file-specification</i>	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"> <li>• Wildcard * represents zero or more characters.</li> <li>• Wildcard ? indicates a single character.</li> </ul>

deletestring *variable\_name* {pos=*position*} | {startseq=*sequence* [startseqnum=*number*] } {length=*length*} |  
{endseq=*sequence* [endseqnum=*number*] } [case=yes | no]

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

echo [*parm\_1*]... [*parm\_n*]

Parameter	Description
<i>parm_1</i>	First parameter to echo.
<i>parm_n</i>	Other parameter(s) to echo.

echolog [value\_1]... [value\_n]

Parameter	Description
value_1	First parameter to echo.
value_n	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 <b>cmd</b> .
<i>cmdref=command-ref</i>	Name of command reference to execute of command type <b>cmdref</b> .
<i>stc=started-task</i>	Name of started task to execute of command type <b>stc</b> .
<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: <b>port</b> , <b>user</b> , <b>pwd</b> , and <b>codepage</b> .  <b>Note:</b> The options in this file override any corresponding options specified for the transfer server in the <b>open</b> command.
<i>xfile=filename</i>	Universal Encrypted text file containing the options for the remote execution server. These options include: <b>port</b> , <b>user</b> , <b>pwd</b> , and <b>codepage</b> .  <b>Note:</b> The options in this file override any corresponding options specified for the transfer server in the <b>open</b> 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</i>   <b>no</b>	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"> <li><b>yes</b> Merge data streams in UDM transaction log.</li> <li><b>no</b> Do not merge data; keep as separate data streams.</li> </ul>
<i>trace=yes</i>   <b>no</b>	Specification for whether or not trace functionality is used: <ul style="list-style-type: none"> <li><b>yes</b> Tracing turned on in the call to UCMD.</li> <li><b>no</b> Tracing turned off in the call to UCMD.</li> </ul>
<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= <b>event</b>	Type of SAP operation to perform ( <b>event</b> 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: <b>port</b> , <b>user</b> , <b>pwd</b> , and <b>codepage</b> . <b>Note:</b> The options in this file override any corresponding options specified for the transfer server in the <b>open</b> command.
xfile= <i>filename</i>	Universal Encrypted text file containing the options for the remote execution server. These options include: <b>port</b> , <b>user</b> , <b>pwd</b> , and <b>codepage</b> . <b>Note:</b> The options in this file override any corresponding options specified for the transfer server in the <b>open</b> command.
key= <i>key</i>	Key used to decrypt an encrypted option file.
mergelog= <b>yes</b>   <b>no</b>	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"> <li><b>yes</b> Merge data streams in UDM transaction log.</li> <li><b>no</b> Do not merge data; keep as separate data streams.</li> </ul>
trace= <b>yes</b>   <b>no</b>	Specification for whether or not trace functionality is used: <ul style="list-style-type: none"> <li><b>yes</b> Tracing turned on in the call to UCMD.</li> <li><b>no</b> Tracing turned off in the call to UCMD.</li> </ul>

exit (There are no parameters used with this command.)

fileys 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.
<b>dd</b>   <b>dsn</b>   <b>hfs</b>   <b>lib</b>	File system to set for the specified server: <ul style="list-style-type: none"> <li><b>dd</b> ddnames defined with JCL DD statements (z/OS only)</li> <li><b>dsn</b> data set name (z/OS only)</li> <li><b>hfs</b> UNIX System Services file system (z/OS and IBM i only)</li> <li><b>lib</b> LIB file system (IBM i only)</li> </ul>

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 <b>binary</b> 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 <b>text</b> 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= <b>yes</b>   <b>no</b>	Specification of whether the search is case-sensitive ( <b>yes</b> ) or case-insensitive ( <b>no</b> ). [Default is <b>no</b> .]
num= <i>number</i>   <b>last</b>	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= <b>center</b>   <b>left</b>   <b>right</b>   <b>justify</b>	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= <b>yes</b>   <b>no</b>	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\_name 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= <b>yes</b>   <b>no</b>	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= <b>text</b>   <b>binary</b>	Type of transfer mode to be used: <ul style="list-style-type: none"> <li><b>text</b> UDM sends source data as is.</li> <li><b>binary</b> UDM performs text translation on data.</li> </ul> [Default is <b>binary</b> when a new session is opened.]
trim= <b>yes</b>   <b>no</b>	Trim trailing spaces during a text transfer: <ul style="list-style-type: none"> <li><b>yes</b> Spaces at the end of each line (record) are removed from the data when it is written.</li> <li><b>no</b> Spaces at the end of each line (record) are not removed.</li> </ul> [Default is <b>no</b> 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"> <li>• Wildcard * represents for zero or more characters.</li> <li>• Wildcard ? indicates a single character.</li> </ul>
<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]

Parameter	Description
primary=*   <b>local</b>   <i>hostname</i>	Logical name of the primary transfer server: <ul style="list-style-type: none"> <li>• * or <b>local</b> Manager is the primary server in a two-party transfer</li> <li>• <i>hostname</i> Host name or IP address of a transfer server</li> </ul>
port= <i>port</i>	Port that the Universal Broker is listening on that will initiate the primary transfer server.
user= <i>username</i>	Name of the user to authenticate on the primary transfer server.
pwd= <i>password</i>	Password used to authenticate the specified user.
codepage= <i>codepage</i>	Codepage to use for text translation by the primary transfer server.
file= <i>filename</i>	Plain text file that can contain open options for the primary transfer server.
xfile= <i>filename</i>	Universal Encrypted file that can contain open options for the primary transfer server.
key= <i>key</i>	Encryption key used to decrypt the specified encrypted open options file.
secondary= <i>hostname</i>	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. <b>Note:</b> Host name must be given from the perspective of the primary transfer server, not the UDM Manager.
port= <i>port</i>	Port that the Universal Broker is listening on that will initiate the secondary transfer server.
user= <i>username</i>	Name of the user to authenticate on the secondary transfer server.
pwd= <i>password</i>	Password used to authenticate the specified user.
codepage= <i>codepage</i>	Codepage to use for text translation by the secondary transfer server.
file= <i>filename</i>	Plain text file that can contain open options for the secondary transfer server.
xfile= <i>filename</i>	Universal Encrypted file that can contain open options for the secondary transfer server.
key= <i>key</i>	Encryption key used to decrypt the specified encrypted open options file.
encrypt= <b>yes</b>   <b>no</b>   <i>cipher</i>	Encryption method for the transfer session: <ul style="list-style-type: none"> <li>• <b>yes</b> Use cipher based on list in the <b>-data_ssl_cipher_list</b> option.</li> <li>• <b>no</b> Use NULL-MD5 cipher.</li> <li>• <i>cipher</i> Specific cipher to use.</li> </ul>



Parameter	Description
compress= <b>yes</b>   <b>no</b>   <b>hasp</b>   <b>zlib</b>	Data compression method for the transfer session: <ul style="list-style-type: none"> <li>• <b>yes</b> Use method as specified in the -compress option.</li> <li>• <b>no</b> Do not use data compression.</li> <li>• <b>hasp</b> Use HASP compression.</li> <li>• <b>zlib</b> Use ZLIB compression.</li> </ul>
nft= <b>yes</b>   <b>no</b>	Specification for network fault tolerance: <ul style="list-style-type: none"> <li>• <b>yes</b> Session is network fault tolerant.</li> <li>• <b>no</b> Session is not network fault tolerant.</li> </ul>

openlog *log\_file\_path* [append=yes|no]

Parameter	Description
<i>log_file_path</i>	Path name of the log file to open.
append= <b>yes</b>   <b>no</b>	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= <b>yes</b>   <b>no</b>	Specification for whether all instances ( <b>yes</b> ) of the sequence are replaced or only the first instance ( <b>no</b> ) of the sequence.
case= <b>yes</b>   <b>no</b>	Specification for whether the comparison to match the old sequence is case sensitive ( <b>yes</b> ) or case insensitive ( <b>no</b> ). [Default is <b>no</b> .]

report *progress=yes* | *no*

Parameter	Description
progress= <b>yes</b>   <b>no</b>	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 <b>_rc</b> 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
<i>_echo</i>	Each command is echoed back after it is read for processing: <b>yes</b> or <b>no</b> (default).
<i>_lines</i>	Line number of each command is reported if an invalid command is received or fails to be parsed or processed by the UDM Manager: <b>yes</b> or <b>no</b> (default).
<i>_halton</i>	Halt condition setting: if the return code of any command is greater than the value of <b>halton</b> , and <b>halton</b> is not zero, UDM issues a <b>quit</b> command and exits.  Values are a simple integer value or one of the following predefined values: <ul style="list-style-type: none"> <li>• <b>none</b>: equivalent to a value of 0; UDM does not automatically quit, regardless of a command's return value.</li> <li>• <b>warn</b>: equivalent to a value of 4; usually returned for invalid commands and command parse errors.</li> <li>• <b>error</b>: equivalent to a value of 8; command fails to execute successfully.</li> <li>• <b>fatal</b>: equivalent to a value of 16; unrecoverable error occurs and the UDM Manager or one of the transfer servers must exit.</li> </ul> [Default is <b>0</b> .]
<i>_rc</i>	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 <b>rc</b> , <b>rc</b> is set to that return value.  Values are a simple integer value or one of the following predefined values: <ul style="list-style-type: none"> <li>• <b>none</b>: equivalent to an integer value of 0; command completed successfully.</li> <li>• <b>warn</b>: equivalent to a value of 4; usually returned for invalid commands and command parse errors.</li> <li>• <b>error</b>: equivalent to a value of 8; command fails to execute successfully.</li> <li>• <b>fatal</b>: equivalent to a value of 16; unrecoverable error occurs and the UDM Manager or one of the transfer servers must exit.</li> </ul> [Default is <b>0</b> .]
<i>_keepalive</i>	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.  <b>Note:</b> This value must be set BEFORE a session is established.
<i>_global_name</i>	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= <b>yes</b>   <b>no</b>	Specification for whether all instances ( <b>yes</b> ) of the sequence are replaced or only the first instance ( <b>no</b> ) of the sequence.
case= <b>yes</b>   <b>no</b>	Specification for whether the comparison to match the old sequence is case sensitive ( <b>yes</b> ) or case insensitive ( <b>no</b> ). [Default is <b>no</b> .]

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= <b>yes</b>   <b>no</b>	Specification for whether or not the comparisons of the start and end sequences are case sensitive ( <b>yes</b> ) or case insensitive ( <b>no</b> ). [Default is <b>no</b> .]

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.