Universal Command 7.0.x

Reference Guide

Universal Command 7.0.x Reference Guide

Universal Command

Universal Command is the central Remote Execution component of the Universal Agent business solution.

These pages provide operating system-specific detailed technical information for Universal Command:

- Usage
- Configuration Options
- Command line syntax
- Command references
- Component Definition options
- Universal Access Control List entries

Universal Command Utilization

For information how Universal Command is utilized, see the Universal Agent 7.0.x User Guide.

Detailed Information

The following pages provide detailed information for Universal Command:

- Universal Command Manager for z/OS
- Universal Command Manager for Windows
- Universal Command Manager for UNIX
- Universal Command Manager for IBM i
- Universal Command Manager Configuration Options
- Universal Command Server for z/OS
- Universal Command Server for Windows
- Universal Command Server for UNIX
- Universal Command Server for IBM i
- Universal Command Server Configuration Options
- Universal Command Component Definition Options
- Universal Command UACL Entries
- Universal Command - Command References
- Universal Command Additional Information

Universal Command Examples

See Universal Command - Remote Execution for examples of how to implement Universal Command.

Universal Command Manager for z/OS

Overview

This page provides information on Universal Command (UCMD) Manager specific to the z/OS operating system.

UCMD Manager executes commands on any computer running the UCMD Server component.

You indicate to the UCMD Manager what command(s) to execute and how the standard input and output and error data should be processed. The UCMD Manager connects to the UCMD Server and processes your request.

The z/OS Batch Manager provides a batch job interface to remote computers running the UCMD Server component. The UCMD Manager executes remote commands as they would be if you entered them directly on the remote command line. Standard input and output and error files are supplied using JCL DD statements.

UCMD Manager registers with a locally running Universal Broker. Consequentially, a Universal Broker must be running in order for a UCMD Manager to execute.

Usage

UCMD Manager for z/OS executes as a batch job. It consists of:

- Batch JCL
- Configuration options

This section describes the JCL, configuration and configuration options, and command line syntax of UCMD Manager for z/OS.

JCL Procedure

The following figure illustrates the UCMD Manager for z/OS JCL procedure (**UCMDPRC**, located in the **SUNVSAMP** library) that is provided with the Universal Agent for z/OS installation to simplify the execution JCL and future maintenance.

```
//UCMDPRC  PROC UPARM=,               -- UCMD options
//             STDOUT='SYSOUT=*',     -- stdout of remote command
//             STDERR='SYSOUT=*',     -- stderr of remote command
//             STDIN='DUMMY',         -- stdin of remote command
//             UCMDPRE=#SHLQ.UNV
//*
//PS1      EXEC PGM=UCMD,PARM='ENVAR(TZ=EST5EDT)/&UPARM'
//STEPLIB  DD  DISP=SHR,DSN=&UCMDPRE..SUNVLOAD
//*
//UNVNLS   DD  DISP=SHR,DSN=&UCMDPRE..SUNVNLS
//UNVTRACE DD  SYSOUT=*
//*
//UNVOUT   DD  &STDOUT-- Remote stdout
//UNVERR   DD  &STDERR-- Remote stderr
//UNVIN    DD  &STDIN-- Remote stdin
//\*
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
```

The procedure provides the parameters **STDIN**, **STDOUT**, and **STDERR** to specify the **stdin**, **stdout**, and **stderr** files, respectively, of the remote command.

The **UPARM** parameter is used to specify EXEC PARM keyword values for the UCMD program. The **PARM** values to the left of the slash (**/**) character are IBM Language Environment parameters.

For information regarding the Time Zone (TZ) environment variable, see z/OS Installation - Time Zone Environment Variable.

The **UCMDPRE** parameter specifies the data set name prefix of Universal Agent installation data sets.

DD Statements Used in JCL Procedure

The following table describes the DD statements used in the UCMD Manager for z/OS JCL procedure, above.

ddname	DCB Attributes *	Mode	Description

STEPLIB	DSORG=PO,
	RECFM=U
		input	Universal Agent load library containing the program being executed.

UNVNLS	DSORG=PO,
	RECFM=(F, FB, V, VB)
		input	Universal Agent national language support library. Contains message catalogs and code page translation tables.

UNVTRACE	DSORG=PS,
	RECFM=(F, FB, V, VB)
		output	UCMD trace output.

UNVOUT	DSORG=PS,
	RECFM=(F, FB, V, VB)
		output	Remote command's stdout file. When the remote command writes to its stdout file, this is the file to which it writes.

UNVERR	DSORG=PS,
	RECFM=(F, FB, V, VB)
		output	Remote command's stderr file. When the remote command writes to its stderr file, this is the file to which it writes.

UNVIN	DSORG=PS,
	RECFM=(F, FB, V, VB)
		input	Remote command's stdin file. When the remote command reads from its stdin file, this is the file from which it reads.

SYSPRINT	DSORG=PS,
	RECFM=(F, FB, V, VB)
		output	Standard output file for the UCMD program. UCMD does not write any messages to SYSPRINT.

SYSOUT	DSORG=PS,
	RECFM=(F, FB, V, VB)
		output	Standard error file for the UCMD program. UCMD writes its messages to SYSOUT.

***** The C runtime library determines the default DCB attributes. See the IBM manual *OS/390 C/C++Programming Guide* for details on default DCB attributes for stream I/O.

DD Statement Categories

UCMD Manager DD statements are organized into the following categories:

Category	DD Statements
Runtime specifications	STEPLIB and UNVNLS.
Remote command standard files (stdin, stdout, and stderr)	UNVIN, UNVOUT, and UNVERR.
UCMD message and command files (stdin, stdout, and stderr)	SYSPRINT, SYSOUT, and UNVTRACE.
Command files	User-defined DD statements to which Universal Command commands refer.

UNVIN Concatenation Requirements

Sequential data sets can be concatenated on the UNVIN ddname. When the UNVIN ddname is open, it is treated as a single data set with the following attributes:

1. The RECFM of the first data set in the concatenation.
2. The LRECL of the first data set in the concatenation.
3. The BLKSIZE of the largest block size in the concatenation.

IBM does not support concatenating data sets with differing RECFM values or LRECL values larger then the first data set in the concatenation. Unpredictable results, including ABENDs, may occur when these rules are not followed.

IBM documents concatenation rules in the IBM C/C++ Programming Guide.

JCL

The following figure illustrates the UCMD Manager for z/OS JCL using the **UCMDPRC**JCL procedure, above.

```
//jobname  JOB CLASS=A,MSGCLASS=X
//STEP1    EXEC UCMDPRC,
//         STDOUT='DISP=SHR,DSN=my.local.file'
//SYSIN    DD  *
 -cmd 'ucopy file' -host dallas -userid joe -pwd akkSdiq
\*
```

Job step STEP1 executes the procedure **UCMDPRC** with the **STDOUT** parameter set to a value that allocates a local file.

The command options are specified on the SYSIN DD.

Configuration

Configuration consists of:

- Setting default options and preferences for all executions of UCMD Manager.
- Setting options and preferences for a single execution of UCMD Manager.

These configuration options are read from the following sources:

1. PARM keyword (command line)
2. SYSIN ddname (command line)
3. Command file ddname
4. Configuration file

The order of precedence is the same as the list above; PARM keyword options being the highest and configuration file being the lowest. That is, options specified via a PARM keyword override options specified via a SYSIN ddname, and so on.

For detailed information on these methods of configuration, see Configuration Management.

Configuration File

The configuration file provides the simplest method of specifying default configuration options whose values you do not want changed with each command invocation. These values are used if the options are not read from one or more other sources.

Some options only can be specified in the configuration file; they have no corresponding command line equivalent. Other options cannot be specified in the configuration file; they must be specified via one or more other sources for a single execution of UCMD Manager.

The configuration file is provided to the UCMD Manager by the local Universal Broker with which it registers. The UCMD Manager configuration file is located in the **UCMCFG00** member of the PDSE allocated to the **UNVCONF** ddname in the Universal Broker started task.

Note

For any changes to the UCMD Manager configuration file to become active, a Universal Broker refresh is required, or the Universal Broker started task must be restarted.

Configuration Options

This section describes the configuration options used to execute UCMD Manager for z/OS.

Configuration Options Categories

The following table categorizes the configuration options into logical areas of application. Each **Category** name is a link to a table of options in that category. Each **Option Name** in those tables is a link to detailed information about that option.

Category	Description
Certificates	X.509 certificate-related options.
Command	Command or script to execute on the remote system. If a script is being executed, the script may reside on the local host on which the Manager is running or the remote host on which the Server is running. It also includes options to control the process environment in which the command executes.
Events	Options used to define event generation.
Local	Options required for local Broker registration.

Messages	Universal Command message options.
Miscellaneous	Options used to display command help and program versions.
Network	Processing options for all the data transferred between the remote and local systems.
Options	Alternative methods to specify configuration options.
Remote	Network address of the remote system and connection options.
Spool	Options that control whether or not the Manager spools its standard files and how they are processed.
Standard File	Processing options for the standard files transferred between the remote and local systems. The STDFILE options are specified differently then the other options. There are three types of standard files: stdin, stdout, and stderr. Each standard file can have a different set of options applied. In order to distinguish between the standard files, the options must start with a standard file specification option (STDERR_SPEC, STDIN_SPEC, or STDOUT_SPEC). The standard file options (names starting with **SIO**_) follow the standard file specification option.
User	User account that the command executes with on the remote system.

Certificate Category Options

Option Name	Description
CA_CERTIFICATES	ddname of the PEM-formatted trusted CA X.509 certificates.
CERTIFICATE	ddname of Manager's PEM-formatted X.509 certificate.
CERTIFICATE_EXPIRATION_NOTICE	Number of days prior to certificate expiration to begin issuing informational messages about the expiration.
CERTIFICATE_REVOCATION_LIST	ddname of the PEM-formatted CRL.
PRIVATE_KEY	ddname of Manager's PEM formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Manager's PRIVATE_KEY.
SAF_KEY_RING	SAF certificate key ring name.
SAF_KEY_RING_LABEL	SAF key ring certificate label.
SSL_IMPLEMENTATION	SSL/TLS implementation.
VERIFY_HOST_NAME	Specifies that the Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specifies that the Broker's X.509 certificate serial number field must be verified.

Command Category Options

Option Name	Description
COMMAND	Remote command to execute.
COMMAND_ID	Unique command ID associated the unit of work.
COMMAND_TYPE	Type of command specified with option COMMAND.
EXIT_CODE_MAP	Allows exit codes from the user process executed by UCMD Server to be translated (mapped) to a corresponding exit code for UCMD Manager.
LOGIN	Specifies whether or not the command runs in a login environment.
MANAGER_FAULT_TOLERANT	Specification for whether or not the manager fault tolerant feature is used.
SCRIPT_FILE	Local script file to execute on the remote system.
SCRIPT_LINE_IGNORE_MASK	Regular expression that is matched against each line of the script being sent to the remote system. Any line that matches the mask will be ignored and not sent to the remote system as part of the script.
SCRIPT_OPTIONS	Command line options passed to the script file.
SCRIPT_TYPE	Type of script file specified by option SCRIPT_FILE.

Events Category Options

Option Name	Description

ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
EVENT_GENERATION	Events to be generated as persistent events.

Local Category Options

Option Name	Description
SYSTEM_ID	Local Universal Broker with which the UCMD Manager must register.

Messages Category Options

Option Name	Description
MESSAGE_LANGUAGE	Language of messages written.
MESSAGE_LEVEL	Level of messages written.
MSG_SUPPRESSION_LIST	List of message IDs representing Universal messages to be suppressed.
TRACE_FILE_LINES	Maximum number of lines written to a trace file before it wraps around.
TRACE_TABLE	Memory trace table specification.

Miscellaneous Category Options

Option Name	Description
COMMENT	User-defined string.
HELP	Write command option help.
VERSION	Write program version.

Network Category Options

Option Name	Description
CODE_PAGE	Code page used for text translation.
CTL_SSL_CIPHER_LIST	SSL/TLS cipher list for the control session.
DATA_AUTHENTICATION	Specification for whether or not data integrity checks are performed on all standard I/O files.
DATA_COMPRESSION	Specification for whether or not data is compressed on all standard I/O files.
DATA_ENCRYPTION	Specification for whether or not data is encrypted on all standard I/O files.
DATA_SSL_CIPHER_LIST	SSL/TLS cipher list for the data sessions.
DEFAULT_CIPHER	Default SSL/TLS cipher used for data sessions.
FORCE_COMPLETE	Forces a manager fault tolerant server in a PENDING communication state to COMPLETED state without retrieving the spooled data.
JOB_RETENTION	Specifies how long a restartable Server waits for a reconnect after the user process completes.
MIN_SSL_PROTOCOL	Minimum SSL/TLS protocol level that will be negotiated and used for communications channels.
NETWORK_DELAY	Maximum number of seconds considered acceptable to wait for data communications.
NETWORK_FAULT_TOLERANT	Specification for whether or not the network fault tolerant protocol is used.
RECONNECT_RETRY_COUNT	Maximum number of network fault tolerant reconnect attempts.
RECONNECT_RETRY_INTERVAL	Number of seconds between network fault tolerant reconnect attempts.
RESTART	Specification for whether or not the manager is requesting restart.
SERVER_STOP_CONDITIONS	Exit codes that cause Universal Broker to cancel the corresponding UCMD Server of the exited UCMD Manager.

Options Category Options

Option Name	Description
ASSIGN_PROCESS_TO_JOB	Specification for whether or not UCMD Server assigns child processes to a single Windows job object.

COMMAND_FILE_ENCRYPTED	Encrypted command file.
COMMAND_FILE_PLAIN	Plain text command file.
ELEVATE_USER_TOKEN	Overrides the UCMD Server ELEVATE_USER_TOKEN configuration option that determines whether a Windows process executes with highest available privileges. This option allows a process to obtain a user access token that is not subject to User Account Control (UAC) restrictions.
ENCRYPTION_KEY	Encryption key used to decrypt an encrypted command file specified by option COMMAND_FILE_ENCRYPTED.
SERVER_OPTIONS	Universal Command Server options that can be overridden by Managers.
UENCRYPTED_CODEPAGE	Character code page to be used when translating characters found within the encrypted command file.

Remote Category Options

Option Name	Description
CONNECT_RETRY_COUNT	Number of times that a UCMD Manager will attempt to establish a connection with a remote Universal Broker.
CONNECT_RETRY_INTERVAL	Number of seconds between failed attempts to establish a connection with a remote Universal Broker.
CONNECT_TIMEOUT	Amount of time that a UCMD Manager will wait for a connection to a remote Universal Broker to complete.
DNS_EXPAND	Number of IP addresses returned to UCMD Manager following a DNS query issued to resolve a host name.
HOST_SELECTION	Host in the REMOTE_HOST list that the UCMD Manager will choose to begin its attempts to connect to a remote Universal Broker.
HOSTNAME_RETRY_COUNT	Number of times that UCMD will attempt to resolve the host name of a specified Universal Broker before it ends with a connect error.
MFT_SAFE_MODE	Situations in which more than one host may be specified in the REMOTE_HOST list when manager fault tolerance (MFT) is enabled.
MSG_SUPPRESSION_LIST	List of message IDs representing Universal messages to be suppressed.
OUTBOUND_IP	Host or IP address to use for all outgoing IP connections.
REMOTE_HOST	List of one or more hosts upon which a command may run.
REMOTE_PORT	TCP/IP port number of the remote Broker.

Standard File Category Options

Option Name	Description
SIO_DATA_AUTHENTICATION	Specifies if data integrity checks are performed on a standard file.
SIO_DATA_COMPRESSION	Specifies if and how data is compressed on a standard file.
SIO_DATA_ENCRYPTION	Specifies if data is encrypted on a standard file.
SIO_LOCAL_CODE_PAGE	Code page used for local text translation on a standard file.
SIO_LOCAL_FILE	Local file used for a standard file instead of the default.
SIO_MODE	Translation mode of a standard file.
SIO_REMOTE_CODE_PAGE	Code page used for remote text translation on a standard file.
SIO_TRAILING_SPACES	Read trailing spaces of z/OS fixed format records.
STDERR_FILE_SPEC	Start of standard error file specification options.
STDIN_FILE_SPEC	Start of standard input file specification options.
STDOUT_FILE_SPEC	Start of standard output file specification options.

User Category Options

Option Name	Description

USER_ID	User ID or account with which to execute the remote command.
USER_PASSWORD	Password associated with USER_ID.

Command Line Syntax

The following figure illustrates the command line syntax — using the command line, long form of the configuration options — of UCMD Manager for z/OS.

```
ucmd
{ -cmdcommand [-cmd_type {cmdref|shell|stc} ] | -scriptddname
  [-optionsoptions] [-script_typetype] [-script_line_ignore_maskmask] }
[-hosthostlist
[-connect_retry_countnumber]
[-connect_retry_intervalseconds]
[-connect_timeoutseconds]
[-dns_expand {yes|no} ]
[-host_selection {sequential|random} ]
[-mft_safe_mode {yes|no} ]
[-fileddname | -encryptedfileddname [-keykey] ]  *
[-portport]
[-system_idID]
[-useriduser [-pwdpwd] ]
[-hostname_retry_countcount]
[-outboundiphost]
[-serveroptions]
[-uencrypted_codepagecodepage]
[-elevate_user_token {yes|no} ]
[-assign_process_to_joboption]
[-managerft {yes|no} ]
[-cmdidid]
[-login {yes|no} ]
[-langlanguage ]
[-level {trace|audit|info|warn|error}[,{time|notime} ]
[-msg_suppression_listlist ]
[-tracefilelineslines]
[-trace_tablesize,{error|always|never} ]
[-ssl_implementation {openssl|system} ]
[-ca_certsddname [-verify_host_name {yes|no|hostname} ]
   [-verify_serial_numbernumber] ]
[-certddname-private_keyddname [-private_key_pwdpassword] ]
[-daysnumber]
[-crlddname]
[-saf_key_ringname [-saf_key_ring_labellabel] ]
[-ctl_ssl_cipher_listcipherlist]
[-data_ssl_cipher_listcipherlist]
[-default_ciphercipher]
[-forcecomplete {yes|no} ]
[-job_retentionseconds]
[-delayseconds]
[-min_ssl_protocoloption]
[-networkft {yes|no} ]
[-retry_countnumber]
[-retry_intervalseconds]
[-restart {yes|no|auto} [-managerft {yes|no} [-cmdidid] ] ]
[-server_stop_conditionscodes]
[-codepagecodepage]
[-compress {yes|no}[,{zlib|hasp} ] ]
[-encrypt {yes|no} ]
[-authenticate {yes|no} ]
[-stdin | -stdout | -stderr]
   [-codepagecodepage]
   [-compress {yes|no}[,{zlib|hasp} ] ]
   [-encrypt {yes|no} ]
   [-authenticate {yes|no} ]
   [-localfileddname]
   [-mode {text|binary}[,{ucs|direct} ] [-trailingspaces {yes|no} ] ]
   [-remotecodepagecodepage]
[-exit_code_mapmap]
[-commenttext]

ucmd
{ -help | -version }
```

* The command file (-file or -encryptedfile) can contain some or all required and/or optional configuration options, including -cmd (or -script) and -host. If a command file is specified on the command line, and it contains the required -cmd (or -script) and -host options, those options do not have to be specified additionally on the command line.

Shutdown Procedure

When the UCMD Manager and UCMD Server are configured to use the network fault tolerant protocol, the z/OS Manager must be shut down properly in order for the UCMD Server and user command to shut down as well. If not properly shut down, the UCMD Server will wait for a period of time for the UCMD Manager to reestablish network connections.

If the UCMD Manager and UCMD Server are not configured to use the network fault tolerant protocol, the z/OS Manager can be shut down with any available method.

Fault Tolerant Shutdown

The z/OS Manager is a z/OS UNIX System Services (USS) application. A shutdown sequence is initiated by sending a USS termination signal to the job. The UCMD Manager detects the signal and promptly sends a terminate message to the UCMD Server. Upon receiving the terminate message, the UCMD Server starts its shutdown sequence.

A termination signal is sent to a z/OS Manager, as shown in the following steps:

Step 1	For the z/OS Manager job to be shut down, obtain its USS process identifier (PID). The PID of a job is displayed with the DISPLAY OMVS console command:
	`D OMVS,U=userid`
	(userid is the user identifier with which the batch job executes.)
	If you do not know the user identifier, all USS work can be displayed with the following command:
	`D OMVS,A=ALL`
	The batch job's name and address space identifier are listed. Find the job name and address space identifier of the job to be terminated and obtain its process identifier from the PID column.

Step 2	Send a termination signal to the process identifier with the MODIFY BPXOINIT command: `F BPXOINIT,TERM=pid` (pid is the process identifier obtained from Step 1.)

Non-Fault Tolerant Shutdown

Since the UCMD Server shutdown sequence is started as soon as a network connection is prematurely closed, the z/OS Manager shutdown can be initiated with any available z/OS job termination method.

The most common and safest method is the CANCEL console command:

```
C jobname
```

(**`jobname`** is the name of the z/OS Manager batch job to terminate.)

Universal Command Manager for Windows

Overview

This page provides information on Universal Command (UCMD) Manager specific to the Windows operating system.

UCMD Manager provides a command line interface to remote computers running the UCMD Server component. The UCMD Manager executes remote commands as they would be if you entered the command directly on the remote command line.

On the command line, you must specify a command to execute and a remote Universal Broker. Additional input to each execution of the UCMD Manager command is made via configuration options, which control product behavior and resource allocation for that execution.

Remote standard input and output files are redirected to the UCMD Manager's standard input and output files.

UCMD Manager registers with a locally running Universal Broker. Consequentially, a Universal Broker must be running in order for a UCMD Manager to execute.

Usage

UCMD Manager for Windows executes as a command line application. It consists of the command line program followed by a list of configuration options. This section describes the command input, configuration and configuration options, and command line syntax.

Standard Input

The UCMD Manager command is executed from the Command Prompt window or a batch file. The **ucmd** command reads from standard input and writes it to the UCMD Server for the remote command to read as its standard input.

When UCMD Manager is executed from the Command Prompt window, standard input is allocated to the window itself. Any characters typed in the Command Prompt window are read as standard input by **ucmd** and transmitted to the UCMD Server.

If **ucmd** is executing a remote command that is reading standard input, it will read the characters being typed in the Command Prompt window until it receives an end-of-file indicator. To enter end-of-file in a Command Prompt window, press <Ctrl+Z*>* <Enter> at the start of a new line.

The allocation of standard input can be changed with a Command Prompt redirection operator. The redirection operators instruct Windows to change the allocation of the standard files.

To change the allocation of standard input, use the **<** operator. Windows can redirect to a special file referred to as NUL. The NUL file is always empty if read from, and never full if written to (all data written to NUL is never saved on disk or in memory).

To allocate standard input to NUL, the command syntax is as follows:

ucmd [OPTIONS...] < NUL

Configuration

Configuration consists of:

- Setting default options and preferences for all executions of UCMD Manager.
- Setting options and preferences for a single execution of UCMD Manager.

Configuration options are read from the following sources:

1. Command line
2. Command file
3. Environment variables
4. Configuration file

The order of precedence is the same as the list above; command line being the highest and configuration file being the lowest. That is, options specified via a command line override options specified via a command file, and so on.

For detailed information on these methods of configuration, see Configuration Management.

Configuration File

The configuration file provides the simplest method of specifying configuration options whose values you do not want changed with each command invocation. These default values are used if the options are not read from one or more other sources.

Although configuration files can be edited with any text editor (for example, Notepad), the Universal Configuration Manager application, accessible via the Control Panel, is the recommended way to set configuration options. The Universal Configuration Manager provides a graphical interface and context-sensitive help, and helps protect the integrity of the configuration file by validating all changes to configuration option values.

Some options only can be specified in the configuration file; they have no corresponding command line equivalent. Other options cannot be specified in the configuration file; they must be specified via one or more other sources for a single execution of UCMD Manager.

Configuration Options

This section describes the configuration options used to execute UCMD Manager for Windows.

Configuration Options Categories

The following table categorizes the configuration options into logical areas of application. Each **Category** name is a link to a table of options in that category. Each **Option Name** in those tables is a link to detailed information about that option.

Category	Description
Certificates	X.509 certificate-related options.
Command	Command or script to execute on the remote system. If a script is being executed, the script may reside on the local host on which the UCMD Manager is running or the remote host on which the UCMD Server is running. The command category also includes options to control the process environment in which the command executes.
Events	Options used to define event generation.
Installation	Options that specify installation requirements, such as directory locations.
Messages	Universal Command message options.
Miscellaneous	Options use to display command help and program versions.
Network	Processing options for all the data transferred between the remote and local systems.
Options	Alternative methods to specify command options.
Remote	Network address of the remote system and connection options.
Spool	Options that control whether or not the UCMD Manager spools its standard files and how they are processed.
Standard File	Processing options for the standard files transferred between the remote and local systems. The STDFILE options are specified differently then the other options. There are three types of standard files: stderr, stdin, and stdout. Each standard file can have a different set of options applied. In order to distinguish between the standard files, the options must start with a standard file specification option (STDIN_SPEC, STDOUT_SPEC, or STDERR_SPEC). The standard file options (names starting with SIO_) follow the standard file specification option.
User	User account that the command executes with on the remote system.

Certificate Category Options

Option Name	Description
CA_CERTIFICATES	Location of the PEM-formatted trusted CA X.509 certificates
CERTIFICATE	Location of Manager's PEM-formatted X.509 certificate.
CERTIFICATE_EXPIRATION_NOTICE	Number of days prior to certificate expiration to begin issuing informational messages about the expiration.
CERTIFICATE_REVOCATION_LIST	Location of Manager's PEM-formatted CRL
PRIVATE_KEY	Location of Manager's PEM-formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Manager's PRIVATE_KEY.
VERIFY_HOST_NAME	Specification for whether or not the Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specification for whether or not the Broker's X.509 certificate serial number field must be verified.

Command Category Options

Option Name	Description
COMMAND	Remote command to execute.
COMMAND_ID	Unique command ID associated the unit of work.
COMMAND_TYPE	Type of command specified with option COMMAND.
EXIT_CODE_MAP	Allows exit codes from user process executed by UCMD Server to be mapped to a corresponding exit code for UCMD Manager
LOGIN	Specification for whether or not the command runs in a login environment.
MANAGER_FAULT_TOLERANT	Specification for whether or not manager fault tolerance is used.
SCRIPT_FILE	Local script file to execute on the remote system.
SCRIPT_OPTIONS	Command line options passed to the script file.
SCRIPT_TYPE	Type of script file specified by option SCRIPT_FILE.

Events Category Options

Option Name	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
EVENT_GENERATION	Events to be generated as persistent events.

Installation Category Options

Option Name	Description
INSTALLATION_DIRECTORY	Directory in which UCMD Manager is installed.

Messages Category Options

Option Name	Description
MESSAGE_LANGUAGE	Language of messages written.
MESSAGE_LEVEL	Level of messages written.
MSG_SUPPRESSION_LIST	List of message IDs representing Universal messages to be suppressed.
NLS_ DIRECTORY	Location of UMC and UTT files
TRACE_FILE_LINES	Maximum number of lines written to a trace file before it wraps around.
TRACE_TABLE	Memory trace table specification.

Miscellaneous Category Options

Option Name	Description

COMMENT	User-defined string
HELP	Write command option help.
VERSION	Write program version.

Network Category Options

Option Name	Description
CODE_PAGE	Code page used for text translation.
CTL_SSL_CIPHER_LIST	SSL/TLS cipher list for the control session.
DATA_AUTHENTICATION	Specification for whether or not data integrity checks are performed on all standard I/O files.
DATA_COMPRESSION	Specification for whether or not data is compressed on all standard I/O files.
DATA_ENCRYPTION	Specification for whether or not data is encrypted on all standard I/O files.
DATA_SSL_CIPHER_LIST	SSL/TLS cipher list for the data sessions.
DEFAULT_CIPHER	Default SSL/TLS cipher used for data sessions.
FORCE_COMPLETE	Forces a manager fault tolerant server in a PENDING communication state to COMPLETED state without retrieving the spooled data.
JOB_RETENTION	Length of time that a Server waits for a reconnect after the user process completes.
MIN_SSL_PROTOCOL	Minimum SSL/TLS protocol level that will be negotiated and used for communications channels.
NETWORK_DELAY	Maximum number of seconds considered acceptable to wait for data communications.
NETWORK_FAULT_TOLERANT	Specification for whether network fault tolerant protocol is used.
RECONNECT_RETRY_COUNT	Maximum number of network fault tolerant reconnect attempts.
RECONNECT_RETRY_INTERVAL	Number of seconds between network fault tolerant reconnect attempts.
RESTART	Specifies if the manager is requesting restart or not.

Options Category Options

Option Name	Description
ASSIGN_PROCESS_TO_JOB	Specification for whether or not UCMD Server assigns child processes to a single Windows job object.
COMMAND_FILE_ENCRYPTED	Encrypted command file.
COMMAND_FILE_PLAIN	Pain text command file.
ELEVATE_USER_TOKEN	Overrides the UCMD Server ELEVATE_USER_TOKEN configuration option that determines whether a Windows process executes with highest available privileges. This option allows a process to obtain a user access token that is not subject to User Account Control (UAC) restrictions.
ENCRYPTION_KEY	Encryption key used to decrypt an encrypted command file specified by option COMMAND_FILE_ENCRYPTED.
SERVER_OPTIONS	Universal Command Server options that can be overridden by Managers.
UENCRYPTED_CODEPAGE	Character code page that Universal Command Manager uses to translate characters within a command file that has been encrypted with the Universal Encrypt utility.
UENCRYPTED_CODEPAGE_PATH	Location of the code page specified by the UENCRYPTED_CODEPAGE option.

Remote Category Options

Option Name	Description
CONNECT_RETRY_COUNT	Number of times a UCMD Manager will attempt to establish a connection with a remote Universal Broker.
CONNECT_RETRY_INTERVAL	Number of seconds between failed attempts to establish a connection with a remote Universal Broker.
CONNECT_TIMEOUT	Amount of time that a UCMD Manager will wait for a connection to a remote Universal Broker to complete.

DNS_EXPAND	Number of IP addresses returned to UCMD Manager following a DNS query issued to resolve a host name.
HOST_SELECTION	Host in the REMOTE_HOST list that the UCMD Manager will choose to begin its attempts to connect to a remote Universal Broker.
HOSTNAME_RETRY_COUNT	Number of times that UCMD will attempt to resolve the host name of a specified Universal Broker before it ends with a connect error.
MFT_SAFE_MODE	Situations in which more than one host may be specified in the REMOTE_HOST list when manager fault tolerance (MFT) is enabled.
OUTBOUND_IP	Host or IP address to use for all outgoing IP connections.
REMOTE_HOST	List of one or more hosts upon which a command may run.
REMOTE_PORT	TCP/IP port number of the remote Broker.

Standard File Category Options

Option Name	Description
SIO_DATA_AUTHENTICATION	Specification for whether or not data integrity checks are performed on a standard file.
SIO_DATA_COMPRESSION	Specification for whether or not data is compressed on a standard file (and if so, how).
SIO_DATA_ENCRYPTION	Specification for whether or not data is encrypted on a standard file.
SIO_LOCAL_CODE_PAGE	Code page used for local text translation on a standard file.
SIO_LOCAL_FILE	Local file used for a standard file instead of the default.
SIO_MODE	Translation mode of a standard file.
SIO_REMOTE_CODE_PAGE	Code page used for remote text translation on a standard file.
STDERR_FILE_SPEC	Start of standard error file specification options.
STDIN_FILE_SPEC	Start of standard input file specification options.
STDOUT_FILE_SPEC	Start of standard output file specification options.

User Category Options

Option Name	Description
USER_ID	User ID or account with which to execute the remote command.
USER_PASSWORD	Password associated with USER_ID.

Command Line Syntax

The following figure illustrates the command line syntax - using the command line, long form of the configuration options - of UCMD Manager for Windows.

```
ucmd
{ -cmdcommand [-cmd_type {cmdref|shell|stc} ] | -scriptfile
  [-optionsoptions] [-script_typetype] }
[-hosthostlist
[-connect_retry_countnumber]
[-connect_retry_intervalseconds]
[-connect_timeoutseconds]
[-dns_expand {yes|no} ]
[-host_selection {sequential|random} ]
[-mft_safe_mode {yes|no} ]
[-filefile | -encryptedfilefile [-keykey] ]  *
[-portport]
[-useriduser [-pwdpwd] ]
[-hostname_retry_countcount]
[-outboundiphost]
[-serveroptions]
[-uencrypted_codepagecodepage]
[-uencrypted_codepage_pathcodepage]
[-elevate_user_token {yes|no} ]
[-assign_process_to_joboption]
[-managerft {yes|no} ]
[-cmdidid]
[-login {yes|no} ]
[-langlanguage]
[-level {trace|audit|info|warn|error}[,{time|notime} ]
[-msg_suppression_listlist ]
[-tracefilelineslines]
[-trace_tablesize,{error|always|never} ]
[-ssl_implementation {openssl|system} ]
[-ca_certsfile [-verify_host_name {yes|no|hostname} ]
   [-verify_serial_numbernumber] ]
[-certfile-private_keyddname [-private_key_pwdpassword] ]
[-daysnumber]
[-crlfile]
[-ctl_ssl_cipher_listcipherlist]
[-data_ssl_cipher_listcipherlist]
[-default_ciphercipher]
[-forcecomplete {yes|no} ]
[-job_retentionseconds]
[-min_ssl_protocoloption]
[-delayseconds]
[-networkft {yes|no} ]
[-retry_countnumber]
[-retry_intervalseconds]
[-restart {yes|no|auto} [-managerft {yes|no} [-cmdidid] ] ]
[-codepagecodepage]
[-compress {yes|no}[,{zlib|hasp} ] ]
[-encrypt {yes|no} ]
[-authenticate {yes|no} ]
[-stdin | -stdout | -stderr]
   [-codepagecodepage]
   [-compress {yes|no}[,{zlib|hasp} ] ]
   [-encrypt {yes|no} ]
   [-authenticate {yes|no} ]
   [-localfileddname]
   [-mode {text|binary}[,{ucs|direct} ]
   [-remotecodepagecodepage]
[-exit_code_mapmap]
[-commenttext]

ucmd
{ -help | -version }
```

* The command file (-file or -encryptedfile) can contain some or all required and/or optional configuration options, including -cmd (or -script) and -host. If a command file is specified on the command line, and it contains the required -cmd (or -script) and -host options, those options do not have to be specified additionally on the command line.

Universal Command Manager for UNIX

Overview

This page provides information on Universal Command (UCMD) Manager specific to the UNIX operating system.

UCMD Manager provides a command line interface to remote computers running the UCMD Server component. The UCMD Manager executes remote commands as they would be if you entered the command directly on the remote command line.

On the command line, you must specify a command to execute and a remote Universal Broker. Additional input to each execution of the UCMD Manager command is made via configuration options, which control product behavior and resource allocation for that execution.

Remote standard input and output files are redirected to the UCMD Manager's standard input and output files.

UCMD Manager registers with a locally running Universal Broker. Consequentially, a Universal Broker must be running in order for a UCMD Manager to execute.

Usage

This section describes the command input, configuration and configuration options, and command line syntax of UCMD Manager for UNIX.

Standard Input

The UCMD Manager command is executed from an interactive UNIX shell or as a shell script. The **ucmd** command reads from standard input and writes it to the UCMD Server for the remote command to read as its standard input.

When the UCMD Manager is executed from an interactive shell, standard input is allocated to the terminal. Any characters typed in the terminal are read as standard input by ucmd and transmitted to the UCMD Server. If **ucmd** is executing a remote command that is reading standard input, it will read the characters being typed in the terminal until it receives an end-of-file indicator. To enter end?of?file in an interactive shell, press <Ctrl+D*>* at the start of a new line.

The allocation of standard input can be changed with a shell redirection operator. The redirection operators instruct the shell to change the allocation of the standard files. To change the allocation of standard input, use the **<** operator. The shell can redirect to a special file referred to as **/dev/null**. The **/dev/null** file is always empty if read from and never full if written to (all data written to **/dev/null** is never saved on disk or in memory).

To allocate standard input to **/dev/null**, the command syntax is as follows:

```
ucmd [OPTIONS...] < /dev/null
```

If **ucmd** is executed as a background job (using the **&** operator), it will receive the SIGTTIN signal when **ucmd** tries to read from standard input. Background jobs cannot read their standard input from the terminal since the foreground job (or the shell) has it allocated. The **ucmd** job is stopped until it is brought to the foreground.

To run a **ucmd** job that does not require terminal input in the background, redirect its standard input from **/dev/null**.

Configuration

Configuration consists of:

Setting default options and preferences for all executions of UCMD Manager.

Setting options and preferences for a single execution of UCMD Manager.

UCMD Manager for UNIX configuration options are read from the following sources:

1. Command line
2. Command file

3. Environment variables
4. Configuration file

The order of precedence is the same as the list above; command line being the highest, and configuration file being the lowest. That is, options specified via a command line override options specified via a command file, and so on.

For detailed information on these methods of configuration, see Configuration Management.

Configuration File

The configuration file, **ucmd.conf**, provides the simplest method of specifying configuration options whose values you do not want changed with each command invocation. These default values are used if the options are not read from one or more other sources.

Some options only can be specified in the configuration file; they have no corresponding command line equivalent. Other options cannot be specified in the configuration file; they must be specified via one or more other sources for a single execution of UCMD Manager.

Configuration Options

This section describes the configuration options used to execute UCMD Manager for UNIX.

Configuration Options Categories

The following table categorizes the configuration options into logical areas of application. Each **Category** name is a link to a table of options in that category. Each **Option Name** in those tables is a link to detailed information about that option.

Category	Description
Certificates	X.509 certificate related options.
Command	Command or script to execute on the remote system. If a script is being executed, the script may reside on the local host on which the Manager is running or the remote host on which the Server is running. It also includes options to control the process environment in which the command executes.
Events	Options used to define event generation.
Installation	Options that specify installation requirements, such as directory locations.
Local	Options required for local broker registration.
Messages	Universal Command message options.
Miscellaneous	Options use to display command help and program versions.
Network	Processing options for all the data transferred between the remote and local systems.
Options	Alternative methods to specify command options.

Remote	Network address of the remote system and connection options.
Standard File	Processing options for the standard files transferred between the remote and local systems. The STDFILE options are specified differently then the other options. There are three types of standard files: stderr, stdin, and stdout. Each standard file can have a different set of options applied. In order to distinguish between the standard files, the options must start with a standard file specification option (STDERR_SPEC, STDIN_SPEC, or STDOUT_SPEC). The standard file options (names starting with SIO_) follow the standard file specification option.
User	User account the command executes with on the remote system.

Certificate Category Options

Option Name	Description
CA_CERTIFICATES	Location of PEM-formatted trusted CA X.509 certificates
CERTIFICATE	Location of Manager's PEM-formatted X.509 certificate.
CERTIFICATE_EXPIRATION_NOTICE	Number of days prior to certificate expiration to begin issuing informational messages about the expiration.
CERTIFICATE_REVOCATION_LIST	Location of Manager's PEM-formatted CRL.
PRIVATE_KEY	Location of Manager's PEM-formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Manager's PRIVATE_KEY.
VERIFY_HOST_NAME	Specification for whether or not the Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specification for whether or not the Broker's X.509 certificate serial number field must be verified.

Command Category Options

Option Name	Description
COMMAND	Remote command to execute.
COMMAND_ID	Unique command ID associated the unit of work.
COMMAND_TYPE	Type of command specified with option COMMAND.
EXIT_CODE_MAP	Allows exit codes from the user process executed by UCMD Server to be translated (mapped) to a corresponding exit code for UCMD Manager.
LOGIN	Specification for whether or not the command runs in a login environment.
MANAGER_FAULT_TOLERANT	Specification for whether or not manager fault tolerance is used.
SCRIPT_FILE	Local script file to execute on the remote system.
SCRIPT_OPTIONS	Command line options passed to the script file.
SCRIPT_TYPE	Type of script file specified by option SCRIPT_FILE.

Events Category Options

Option Name	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
EVENT_GENERATION	Events to be generated as persistent events.

Installation Category Options

Option Name	Description
INSTALLATION_DIRECTORY	Directory in which UCMD Manager is installed.

Local Category Options

Option Name	Description
BIF_DIRECTORY	Broker Interface File (BIF) directory where the Universal Broker interface file is located.
PLF_DIRECTORY	Program Lock File (PLF) directory where the program lock files are located.

Messages Category Options

Option Name	Description
MESSAGE_LANGUAGE	Language of messages written.
MESSAGE_LEVEL	Level of messages written.
MSG_SUPPRESSION_LIST	List of message IDs representing Universal messages to be suppressed.
NLS_ DIRECTORY	Location of UMC and UTT files
TRACE_FILE_LINES	Maximum number of lines written to a trace file before it wraps around.
TRACE_TABLE	Memory trace table specification.

Miscellaneous Category Options

Option Name	Description
COMMENT	User-defined string.
HELP	Write command option help.
VERSION	Write program version.

Network Category Options

Option Name	Description
CODE_PAGE	Code page used for text translation.
CTL_SSL_CIPHER_LIST	SSL/TLS cipher list for the control session.
DATA_AUTHENTICATION	Specification for whether or not data integrity checks are performed on all standard I/O files.
DATA_COMPRESSION	Specification for whether or not data is compressed on all standard I/O files.
DATA_ENCRYPTION	Specification for whether or not data is encrypted on all standard I/O files.
DATA_SSL_CIPHER_LIST	SSL/TLS cipher list for the data sessions.
DEFAULT_CIPHER	Default SSL/TLS cipher used for data sessions.
FORCE_COMPLETE	Forces a manager fault tolerant server in a PENDING communication state to COMPLETED state without retrieving the spooled data.
JOB_RETENTION	Length of time that a Server waits for a reconnect after the user process completes.
MIN_SSL_PROTOCOL	Minimum SSL/TLS protocol level that will be negotiated and used for communications channels.
NETWORK_DELAY	Maximum number of seconds considered acceptable to wait for data communications.
NETWORK_FAULT_TOLERANT	Specification for whether or not the network fault tolerant protocol is used.
RECONNECT_RETRY_COUNT	Maximum number of network fault tolerant reconnect attempts.
RECONNECT_RETRY_INTERVAL	Number of seconds between network fault tolerant reconnect attempts.
RESTART	Specification for whether or not the manager is requesting restart.

Options Category Options

Option Name	Description
ASSIGN_PROCESS_TO_JOB	Specification for whether or not UCMD Server assigns child processes to a single Windows job object.
COMMAND_FILE_ENCRYPTED	Encrypted command file.
COMMAND_FILE_PLAIN	Plain text command file.
ELEVATE_USER_TOKEN	Overrides the UCMD Server ELEVATE_USER_TOKEN configuration option that determines whether a Windows process executes with highest available privileges. This option allows a process to obtain a user access token that is not subject to User Account Control (UAC) restrictions.
ENCRYPTION_KEY	Encryption key used to decrypt an encrypted command file specified by option COMMAND_FILE_ENCRYPTED.
SERVER_OPTIONS	UCMD Server options that can be overridden by UCMD Managers.
UENCRYPTED_CODEPAGE	Character code page that Universal Command Manager uses to translate characters within a command file that has been encrypted with the Universal Encrypt utility.
UENCRYPTED_CODEPAGE_PATH	Location of the code page specified by the UENCRYPTED_CODEPAGE option.

Remote Category Options

Option Name	Description
CONNECT_RETRY_COUNT	Number of times a UCMD Manager will attempt to establish a connection with a remote Universal Broker.
CONNECT_RETRY_INTERVAL	Number of seconds between failed attempts to establish a connection with a remote Universal Broker.
CONNECT_TIMEOUT	Amount of time that a UCMD Manager will wait for a connection to a remote Universal Broker to complete.
DNS_EXPAND	Number of IP addresses returned to UCMD Manager following a DNS query issued to resolve a host name.
HOST_SELECTION	Host in the REMOTE_HOST list that the UCMD Manager will choose to begin its attempts to connect to a remote Universal Broker.
HOSTNAME_RETRY_COUNT	Number of times that UCMD will attempt to resolve the host name of a specified Universal Broker before it ends with a connect error.
MFT_SAFE_MODE	Situations in which more than one host may be specified in the REMOTE_HOST list when manager fault tolerance (MFT) is enabled.
OUTBOUND_IP	Host or IP address to use for all outgoing IP connections.
REMOTE_HOST	List of one or more hosts upon which a command may run.
REMOTE_PORT	TCP/IP port number of the remote Broker.

Standard File Category Options

Option Name	Description
SIO_DATA_AUTHENTICATION	Specification for whether or not data integrity checks are performed on a standard file.
SIO_DATA_COMPRESSION	Specification for whether or not data is compressed on a standard file (and if so, how).
SIO_DATA_ENCRYPTION	Specification for whether or not data is encrypted on a standard file.
SIO_LOCAL_CODE_PAGE	Code page used for local text translation on a standard file.
SIO_LOCAL_FILE	Local file used for a standard file instead of the default.
SIO_MODE	Translation mode of a standard file.
SIO_REMOTE_CODE_PAGE	Code page used for remote text translation on a standard file.
STDERR_FILE_SPEC	Start of standard error file specification options.

STDIN_FILE_SPEC	Start of standard input file specification options.
STDOUT_FILE_SPEC	Start of standard output file specification options.

User Category Options

Option Name	Description
USER_ID	User ID or account with which to execute the remote command.
USER_PASSWORD	Password associated with USER_ID.

Command Line Syntax

The following figure illustrates the command line syntax – using the command line, long form of the configuration options – of UCMD Manager for UNIX.

```
ucmd
{ -cmdcommand [-cmd_type {cmdref|shell|stc} ] | -scriptfile
  [-optionsoptions] [-script_typetype] }
[-hosthostlist
[-connect_retry_countnumber]
[-connect_retry_intervalseconds]
[-connect_timeoutseconds]
[-dns_expand {yes|no} ]
[-host_selection {sequential|random} ]
[-mft_safe_mode {yes|no} ]
[-filefile | -encryptedfilefile [-keykey] ]  *
[-portport]
[-useriduser [-pwdpwd] ]
[-hostname_retry_countcount]
[-outboundiphost]
[-bif_directorydirectory]
[-plf_directorydirectory]
[-serveroptions]
[-uencrypted_codepagecodepage]
[-uencrypted_codepage_pathcodepage]
[-elevate_user_token {yes|no} ]
[-assign_process_to_joboption]
[-managerft {yes|no} ]
[-cmdidid]
[-login {yes|no} ]
[-langlanguage]
[-level {trace|audit|info|warn|error}[,{time|notime} ]
[-msg_suppression_listlist ]
[-tracefilelineslines]
[-trace_tablesize,{error|always|never} ]
[-ca_certsfile [-verify_host_name {yes|no|hostname} ]
   [-verify_serial_numbernumber] ]
[-certfile-private_keyddname [-private_key_pwdpassword] ]
[-daysnumber]
[-crlfile]
[-ctl_ssl_cipher_listcipherlist]
[-data_ssl_cipher_listcipherlist]
[-default_ciphercipher]
[-forcecomplete {yes|no} ]
[-job_retentionseconds]
[-delayseconds]
[-min_ssl_protocoloption]
[-networkft {yes|no} ]
[-retry_countnumber]
[-retry_intervalseconds]
[-restart {yes|no|auto} [-managerft {yes|no} [-cmdidid] ] ]
[-codepagecodepage]
[-compress {yes|no}[,{zlib|hasp} ] ]
[-encrypt {yes|no} ]
[-authenticate {yes|no} ]
[-stdin | -stdout | -stderr]
   [-codepagecodepage]
   [-compress {yes|no}[,{zlib|hasp} ] ]
   [-encrypt {yes|no} ]
   [-authenticate {yes|no} ]
   [-localfileddname]
   [-mode {text|binary}[,{ucs|direct} ]
   [-remotecodepagecodepage]
[-exit_code_mapmap]
[-commenttext]

ucmd
{ -help | -version }
```

* The command file (-file or -encryptedfile) can contain some or all required and/or optional configuration options, including -cmd (or -script) and -host. If a command file is specified on the command line, and it contains the required -cmd (or -script) and -host options, those options do not have to be specified additionally on the command line.

Universal Command Manager for IBM i

Currently, IBM i runs Workload Automation 5.1.1.   These pages provide information for that version.

Overview

This page provides information on Universal Command (UCMD) Manager specific to the IBM i operating system.

UCMD Manager for IBM i provides an IBM i command interface to remote computers running the UCMD Server component. UCMD Manager executes remote commands as they would be if you entered the command directly on the remote command line.

On the command line, you must specify a command to execute and a remote Universal Broker. Additional input to each execution of the UCMD Manager command is made via configuration options, which control product behavior and resource allocation for that execution.

Remote standard input and output files are redirected to the UCMD Manager's standard input and output files.

Usage

UCMD Manager for IBM i executes as a CL command. This section describes the command environment, configuration and configuration options, and command line syntax.

Workload Automation 5 for IBM i Commands

The names of the Workload Automation 5 for IBM i commands that are installed in the IBM i **QSYS** library are tagged with the Workload Automation 5 for IBM i version / release / modification number, **511**. The names of the commands installed in the Workload Automation 5 for IBM i product library, **UNVPRD511**, are untagged.

To maintain consistency across releases, you may prefer to use the untagged names in your production environment. The **UCHGRLS** (Change Release Tag) program lets you change the tagged command names in **QSYS** to the untagged command names in **UNVPRD511**.

(See UCHGRLS (Change Release Tag) Program for detailed information on **UCHGRLS**.)

These pages references the IBM i commands by their untagged names. If you are using commands with tagged names to run UCMD, substitute the tagged names for the untagged names in these references.

Command Execution Environment

The command is valid in all environments:

1. Batch input streams
2. CL programs
3. REXX procedures
4. CL ILE modules
5. Interactive processing
6. Passed to the system program QCMDEXC (or QCAEXEC) for processing

Configuration

Configuration consists of:

- Setting default options and preferences for all executions of UCMD Manager.
- Setting options and preferences for a single execution of UCMD Manager.

UCMD Manager for IBM i configuration options are read from the following sources:

1. STRUCM parameters
2. Environment variables
3. Configuration file

The order of precedence is the same as the list above; STRUCM parameters being the highest, and configuration file being the lowest. That is, options specified via STRUCM parameters override options specified via environment variables, and so on.

For detailed information on these methods of configuration, see Configuration Management.

Configuration File

The configuration file, **UNVPRD511/UNVCONF(UCMD)**, provides the simplest method of specifying configuration options whose values you do not want changed with each command invocation. These default values are used if the options are not read from one or more other sources.

Some options only can be specified in the configuration file; they have no corresponding command line equivalent. Other options cannot be specified in the configuration file; they must be specified via one or more other sources for a single execution of UCMD Manager.

Configuration Options

This section describes the configuration options used to execute UCMD Manager for IBM i.

Configuration Options Categories

The following table categorizes the configuration options into logical areas of application. Each **Category** name is a link to a table of options in that category. Each **Option Name** in those tables is a link to detailed information about that option.

Category	Description
Certificates	X.509 certificate related options.
Command	Command or script to execute on the remote system. If a script is being executed, the script may reside on the local host on which the Manager is running or the remote host on which the Server is running. It also includes options to control the process environment in which the command executes.
Events	Options used to define event generation.
Local	Options required for local broker registration.
Messages	Universal Command message options.
Miscellaneous	Options use to display command help and program versions.
Network	Processing options for all the data transferred between the remote and local systems.
Options	Alternative methods to specify command options.
Remote	Network address of the remote system and connection options.
Standard File	Processing options for the standard files transferred between the remote and local systems. At the program interface level, the STDFILE options are specified differently then the other options. There are three types of standard files: stdin, stdout, and stderr. Each standard file can have a different set of options applied. In order to distinguish between the standard files, the options must start with a standard file specification option (STDERR_FILE_SPEC, STDIN_FILE_SPEC, or STDOUT_FILE_SPEC). The standard file options (see #Standard File Category Options) follow the standard file specification option. The STRUCM command interface hides this difference from the user.
User	User account the command executes with on the remote system.

Certificate Category Options

Option Name	Description

CA_CERTIFICATES	Location of the PEM formatted trusted CA X.509 certificates
CERTIFICATE	Location of Manager's PEM formatted X.509 certificate.
CERTIFICATE_REVOCATION_LIST	Location of Manager's PEM-formatted CRL.
PRIVATE_KEY	Location of Manager's PEM formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Manager's PRIVATE_KEY.
VERIFY_HOST_NAME	Specification for whether or not the Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specification for whether or not the Broker's X.509 certificate serial number field must be verified.

Command Category Options

Option Name	Description
COMMAND	Remote command to execute.
COMMAND_ID	Unique command ID associated the unit of work.
COMMAND_TYPE	Type of command specified with option COMMAND.
EXIT_CODE_MAP	Allows exit codes from the user process executed by UCMD Server to be translated (mapped) to a corresponding exit code for UCMD Manager.
LOGIN	Specification for whether or not the command runs in a login environment.
MANAGER_FAULT_TOLERANT	Specification for whether or not manager fault tolerance is used.
SCRIPT_FILE	Local script file to execute on the remote system.
SCRIPT_OPTIONS	Command line options passed to the script file.
SCRIPT_TYPE	Type of script file specified by option SCRIPT_FILE.

Events Category Options

Option Name	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
EVENT_GENERATION	Events to be generated as persistent events.

Local Category Options

Option Name	Description
PLF_DIRECTORY	Program Lock File (PLF) directory where the program lock files are located.

Messages Category Options

Option Name	Description
MESSAGE_LANGUAGE	Language of messages written.
MESSAGE_LEVEL	Level of messages written.
TRACE_FILE_LINES	Maximum number of lines written to a trace file before it wraps around.
TRACE_TABLE	Memory trace table specification.

Miscellaneous Category Options

Option Name	Description
COMMENT	User-defined string.
VERSION	Write program version.

Network Category Options

Option Name	Description
CODE_PAGE	Code page used for text translation.
CTL_SSL_CIPHER_LIST	SSL/TLS cipher list for the control session.
DATA_AUTHENTICATION	Specification for whether or not data integrity checks are performed on all standard I/O files.
DATA_COMPRESSION	Specification for whether or not data is compressed on all standard I/O files.
DATA_ENCRYPTION	Specification for whether or not data is encrypted on all standard I/O files.
DATA_SSL_CIPHER_LIST	SSL/TLS cipher list for the data sessions.
DEFAULT_CIPHER	Default SSL/TLS cipher used for data sessions.
FORCE_COMPLETE	Forces a manager fault tolerant server in a PENDING communication state to COMPLETED state without retrieving the spooled data.
JOB_RETENTION	Length of time that a Server waits for a reconnect after the user process completes.
MIN_SSL_PROTOCOL	Minimum SSL/TLS protocol level that will be negotiated and used for communications channels. Note ⚠ This option was introduced to IBM i in version 5.1.1.0.
NETWORK_DELAY	Maximum number of seconds considered acceptable to wait for data communications.
NETWORK_FAULT_TOLERANT	Specification for whether or not the network fault tolerant protocol is used.
RECONNECT_RETRY_COUNT	Maximum number of network fault tolerant reconnect attempts.
RECONNECT_RETRY_INTERVAL	Number of seconds between network fault tolerant reconnect attempts.
RESTART	Specification for whether or not the manager is requesting restart.

Options Category Options

Option Name	Description
ASSIGN_PROCESS_TO_JOB	Specification for whether or not UCMD Server assigns child processes to a single Windows job object.
COMMAND_FILE_ENCRYPTED	Encrypted command file.
COMMAND_FILE_PLAIN	Plain text command file.
ENCRYPTION_KEY	Encryption key used to decrypt an encrypted command file specified by option COMMAND_FILE_ENCRYPTED.
SERVER_OPTIONS	UCMD Server options that can be overridden by UCMD Managers.
UENCRYPTED_CODEPAGE	Character code page that Universal Command Manager uses to translate characters within a command file that has been encrypted with the Universal Encrypt utility.

Remote Category Options

Option Name	Description
CONNECT_TIMEOUT	Amount of time that a UCMD Manager will wait for a connection to a remote Universal Broker to complete.
DNS_EXPAND	Number of IP addresses returned to UCMD Manager following a DNS query issued to resolve a host name.
HOST_SELECTION	Host in the REMOTE_HOST list that the UCMD Manager will choose to begin its attempts to connect to a remote Universal Broker.
HOSTNAME_RETRY_COUNT	Number of times that Universal Command will attempt to resolve the host name of a specified Universal Broker before it ends with a connect error.
MFT_SAFE_MODE	Situations in which more than one host may be specified in the REMOTE_HOST list when manager fault tolerance (MFT) is enabled.

OUTBOUND_IP	Host or IP address to use for all outgoing IP connections.
REMOTE_HOST	TCP/IP host name of the remote Broker.
REMOTE_PORT	TCP/IP port number of the remote Broker.

Standard File Category Options

Option Name	Description
SIO_LOCAL_CODE_PAGE	Code page used for local text translation on a standard file.
SIO_LOCAL_FILE	Local file used for a standard file instead of the default.
SIO_MODE	Translation mode of a standard file.
SIO_REMOTE_CODE_PAGE	Code page used for remote text translation on a standard file.

User Category Options

Option Name	Description
USER_ID	User ID or account with which to execute the remote command.
USER_PASSWORD	Password associated with USER_ID.

Command Line Syntax

The following figure illustrates the command line syntax — using the STRUCM parameter form of the configuration options — of UCMD Manager for IBM i.

After the positional options, which appear immediately after the STRUCM command, the options are organized by category.

```
STRUCM
{ CMD(command) [CMDTYPE({cmd|cmdref|rexx})] | SCRFILE(file) [SCRMBR(member)]
   [OPTIONS(options)] [SCRTYPE(type)] }
HOST(hostaddress)
[PORT(port)]
[USERID(user) [PWD(pwd)] ]

REMOTE CATEGORY:
[CONNECTTO(seconds)]
[DNSEXPAND(*option)]
[HOSTSELECT(*option)]
[HSTNMRTYCT(count)]
[MFTSAFMODE(*option)]
[OUTBOUNDIP(host)]

OPTIONS CATEGORY:
[ASSIGNPROC(*options)]
[CMDFILE(file) [CMDMBR(member)] ] | [ECMFILE(file) [ECMMBR(member)]
   [KEY(key)] ]
[ECMFILECP(codepage)]
[SERVER(options)]

COMMAND CATEGORY:
[CMDID(id)]
[EXITCDMAP(option)]
[LOGIN({yes|no})]
[MANAGERFT({yes|no})]

MESSAGES CATEGORY:
[MSGLANG(language)]
[MSGLEVEL(*{trace|audit|info|warn|error})]                NOTE: Value *trace* turns trace on.
[TRCLINES(lines)]
[TRCTBL(size,{error|always|never}) ]
[CACERTS(file [lib] ) [CACERTSMBR(member)] [VFYHSTNM({yes|no|hostname})]
   [VFYSERNUM(number)] ]
[CERT(file [lib] ) [CERTMBR(member)]
   PVTKEYF(file [lib] ) [PVTKEYFMBR(member)] [PVTKEYPWD(password)] ]
[CRLFILE(file [lib]) [CRLMBR(member)] ]

NETWORK CATEGORY:
[AUTH(*{yes|no})]
[CODEPAGE(codepage)]
[COMPRESS(*{yes|no}[CMPRSMTH*{zlib|hasp}] ) ]
[CTLCPHRLST(cipherlist)]
[DELAY(seconds)]
[DFTCPHR(cipher)]
[DTACPHRLST(cipherlist)]
[ENCRYPT(*{yes|no})]
[FRCCMPLT({yes|no})]
[JOBRTN(seconds)]
[MINSSLPROTOCOL(option)]
[NETWORKFT({yes|no})]
[RESTART({yes|no|auto}) [MANAGERFT({yes|no}) [CMDID(id)] ] ]
[RETRYCNT(number)]
[RETRYINT(seconds)]
[SERFILE(output_file)]
[SERMBR(member)]
[SERMODE(*{text|binary})]
[SERCPG(codepage)]
[SERRCP(codepage)]
[SINFILE(input_file)]
[SINMBR(member)]
[SINMODE(*{text|binary})]
[SINCPG(codepage)]
[SINRCP(codepage)]
[SOTCPG(codepage)]
[SOTFILE(output_file)]
[SOTMBR(member)]
[SOTMODE(*{text|binary})]
[SOTRCP(codepage)]

LOCAL CATEGORY:
[PLFDIR(directory)]

MISCELLANEOUS CATEGORY:
[COMMENT(user-defined string)]



STRUCM

VERSION(*{yes|no})
```

* The command file (CMDFILE or ECMFILE) can contain some or all required and/or optional configuration options, including CMD (or SCRFILE) and HOST. If a command file is specified on the command line, and it contains the required CMD (or SCRFILE) and HOST options, those options do not have to be specified additionally on the command line.

Universal Command Manager Configuration Options

Overview

This page provides links to detailed information on the configuration options available for use with the Universal Command Manager.

The options are listed alphabetically, without regard to any specific operating system.

Configuration Options Information

For each configuration option, the following information is provided.

Description

Describes the option and how it is used.

Usage

Provides a table of the following information:

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	<Format / Value>				
Command Line, Long Form	<Format / Value>				
Environment Variable	<Format / Value>				
Configuration File Keyword	<Format / Value>				
STRUCM Parameter	<Format / Value>				

Method

Identifies the different methods used to specify Universal Command Manager configuration options:

- Command Line Option, Short Form
- Command Line Option, Long Form
- Environment Variable
- Configuration File Keyword
- STRUCM Parameter

Note

Each option can be specified using one or more methods.

Syntax

Identifies the syntax of each method that can be used to specify the option:

- Format: Specific characters that identify the option.
- Value: Type of value(s) to be supplied for this method.

Note

If a Method is not valid for specifying the option, the Syntax field contains **n/a**.

(Operating System)

Identifies the operating systems for which each method of specifying the option is valid:

- IBM i
- UNIX
- Windows
- z/OS

Values

Identifies all possible values for the specified value type.

Defaults are identified in **[bracketed bold type]**.

<Additional Information>

Identifies any additional information specific to the option.

Configuration Options List

The following table identifies all Universal Command Manager configuration options.

Option	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
ASSIGN_PROCESS_TO_JOB	Specification for whether or not UCMD Server assigns child processes to a single Windows job object.
BIF_DIRECTORY	Broker Interface Directory that specifies the location of the Universal Broker interface file
CA_CERTIFICATES	File name / ddname of the PEM-formatted trusted CA X.509 certificates.
CERTIFICATE	File name / ddname of UCMD Manager's PEM-formatted X.509 certificate.
CERTIFICATE_EXPIRATION_NOTICE	Number of days prior to certificate expiration to begin issuing informational messages about the expiration.
CERTIFICATE_REVOCATION_LIST	File name / ddname of the PEM-formatted CRL.
CODE_PAGE	Code page used for text translation.
COMMAND	Remote command to execute.
COMMAND_FILE_ENCRYPTED	Encrypted command file.
COMMAND_FILE_PLAIN	Plain text command file.
COMMAND_ID	Unique command ID associated the unit of work.
COMMAND_TYPE	Type of command specified with option COMMAND.
COMMENT	User-defined string.
CONNECT_RETRY_COUNT	Number of times a UCMD Manager will attempt to establish a connection with a remote Universal Broker.
CONNECT_RETRY_INTERVAL	Number of seconds between failed attempts to establish a connection with a remote Universal Broker.
CONNECT_TIMEOUT	Amount of time that a UCMD Manager will wait for a connection to a remote Universal Broker to complete.
CTL_SSL_CIPHER_LIST	SSL/TLS cipher list for the control session.
DATA_AUTHENTICATION	Specification for whether or not data integrity checks are performed on all standard I/O files.
DATA_COMPRESSION	Specification for whether or not data is compressed on all standard I/O files.
DATA_ENCRYPTION	Specification for whether or not data is encrypted on all standard I/O files.
DATA_SSL_CIPHER_LIST	SSL/TLS cipher list for the data sessions.
DEFAULT_CIPHER	Default SSL/TLS cipher used for data sessions.
DNS_EXPAND	Number of IP addresses returned to UCMD Manager following a DNS query issued to resolve a host name.

ELEVATE_USER_TOKEN	Overrides the UCMD Server ELEVATE_USER_TOKEN configuration option that determines whether a Windows process executes with highest available privileges. This option allows a process to obtain a user access token that is not subject to User Account Control (UAC) restrictions.
ENCRYPTION_KEY	Encryption key used to decrypt an encrypted command file specified by option COMMAND_FILE_ENCRYPTED.
EVENT_GENERATION	Events to be generated as persistent events.
EXIT_CODE_MAP	Translates (maps) exit codes from user process to exit codes for UCMD Manager.
FORCE_COMPLETE	Specification to force a manager fault tolerant server in a PENDING communication state to COMPLETED state without retrieving the spooled data.
HELP	Displays a description of the command line options and their format.
HOST_SELECTION	Host in the REMOTE_HOST list that the UCMD Manager will choose to begin its attempts to connect to a remote Universal Broker.
HOSTNAME_RETRY_COUNT	Number of times that UCMD will attempt to resolve the host name of a specified Universal Broker before it ends with a connect error.
INSTALLATION_DIRECTORY	Base directory in which UCMD Manager is installed.
JOB_RETENTION	Specification of how long a restartable Server waits for a reconnect after the user process completes.
LOGIN	Specification for whether or not the command runs in a login environment.
MANAGER_FAULT_TOLERANT	Specification for whether or not the manager fault tolerant feature is used.
MESSAGE_LANGUAGE	Language of messages written.
MESSAGE_LEVEL	Level of messages written.
MFT_SAFE_MODE	Situations in which more than one host may be specified in the REMOTE_HOST list when manager fault tolerance (MFT) is enabled.
MIN_SSL_PROTOCOL	Minimum SSL/TLS protocol level that will be negotiated and used for communications channels. Note ⚠ This option was introduced to IBM i in version 5.1.1.0.
MSG_SUPPRESSION_LIST	List of message IDs representing Universal messages to be suppressed.
NETWORK_DELAY	Maximum number of seconds considered acceptable to wait for data communications.
NETWORK_FAULT_TOLERANT	Specification for whether or not the network fault tolerant protocol is used.
NLS_DIRECTORY	Location of UMC and UTT files.
OUTBOUND_IP	Host or IP address to use for all outgoing IP connections.
PLF_DIRECTORY	Program Lock File directory that specifies the location of the UCMD Manager program lock file.
PRIVATE_KEY	ddname of Manager's PEM formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Manager's PRIVATE_KEY.
RECONNECT_RETRY_COUNT	Maximum number of network fault tolerant reconnect attempts.
RECONNECT_RETRY_INTERVAL	Number of seconds between network fault tolerant reconnect attempts.
REMOTE_HOST	List of one or more hosts upon which a command may run.
REMOTE_PORT	TCP/IP port number of the remote computer on which Universal Broker is running and accepting connections.
RESTART	Specification for whether or not the manager is requesting restart.
SAF_KEY_RING	SAF certificate key ring name.
SAF_KEY_RING_LABEL	SAF key ring certificate label.
SCRIPT_FILE	Local script file to execute on the remote system.
SCRIPT_LINE_IGNORE_MASK	Regular expression that is matched against each line of the script being sent to the remote system. Any line that matches the mask will be ignored and not sent to the remote system as part of the script.

SCRIPT_OPTIONS	Command line options passed to the script file.
SCRIPT_TYPE	Type of script file specified by option SCRIPT_FILE.
SERVER_OPTIONS	Universal Command Server options that can be overridden by Managers.
SERVER_STOP_CONDITIONS	Exit codes that cause Universal Broker to cancel the corresponding UCMD Server of the exited UCMD Manager.
SIO_DATA_AUTHENTICATION	Specification for whether or not data integrity checks are performed on a standard file.
SIO_DATA_COMPRESSION	Specification for whether or not data is compressed on a standard file, and if so, the method used.
SIO_DATA_ENCRYPTION	Specification for whether or not data is encrypted on a standard file.
SIO_LOCAL_CODE_PAGE	Code page used for local text translation on a standard file.
SIO_LOCAL_FILE	Local file used for a standard file instead of the default.
SIO_MODE	Translation mode of a standard file.
SIO_REMOTE_CODE_PAGE	Code page used for remote text translation on a standard file.
SIO_TRAILING_SPACES	Specification for whether not to read trailing spaces of z/OS fixed format records.
SSL_IMPLEMENTATION	SSL/TLS implementation.
STDERR_FILE_SPEC	Start of standard error file specification options.
STDIN_FILE_SPEC	Start of standard input file specification options.
STDOUT_FILE_SPEC	Start of standard output file specification options.
SYSTEM_ID	Local Universal Broker with which the UCMD Manager must register.
TRACE_FILE_LINES	Maximum number of lines written to a trace file before it wraps around.
TRACE_TABLE	Memory trace table specification.
UENCRYPTED_CODEPAGE	Character code page that Universal Command Manager uses to translate characters within a command file that has been encrypted with the Universal Encrypt utility.
UENCRYPTED_CODEPAGE_PATH	Location of the code page specified by the UENCRYPTED_CODEPAGE option.
USER_ID	User ID or account with which to execute the remote command.
USER_PASSWORD	Password associated with USER_ID.
VERIFY_HOST_NAME	Specification for whether or not the Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specification for whether or not the Broker's X.509 certificate serial number field must be verified.
VERSION	Write program version.

ACTIVITY_MONITORING - UCMD Manager configuration option

Description

The ACTIVITY_MONITORING option specifies whether or not product activity monitoring events are generated.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line	-uag_enable_ssl *option*		✓	✓	
Environment Variable	UAGENABLESSL=*option*		✓		

Values

option is the specification for whether or not product activity monitoring events are generated.

Valid values for *option* are:

- **yes**
  Activate product activity monitoring events
- **no**
  Deactivate product activity monitoring events

Default is yes.

ASSIGN_PROCESS_TO_JOB - UCMD Manager configuration option

Description

The ASSIGN_PROCESS_TO_JOB option controls the startup and shutdown behavior of UCMD Server processes executed on Windows.

- If ASSIGN_PROCESS_TO_JOB is set to **yes**, UCMD Server assigns all of its child processes to a system resource known as a job object.
- If ASSIGN_PROCESS_TO_JOB is set to **no**, child processes are not assigned to a job object, and no relationship among parent / child processes is maintained.

Note

ASSIGN_PROCESS_TO_JOB overrides a UCMD Server for Windows ASSIGN_PROCESS_TO_JOB option. It is available only for Windows UCMD Servers; it is ignored on all other Server platforms.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-assign_process_to_job *option*		✓	✓	✓
Environment Variable	UCMDASSIGN PROCESSTOJOB=*option*	✓	✓	✓	✓
Configuration File Keyword	assign_process_to_job *option*	✓	✓	✓	✓
STRUCM Parameter	ASSIGNPROC(**option*)	✓			

Values

option is the specification for whether or not UCMD Server child processes are assigned to a job.

Valid values for *option* are:

- **yes**
  Override the UCMD Server ASSIGN_PROCESS_TO_JOB option and assign all UCMD Server child processes to a job object.
- **no**
  Override the UCMD Server ASSIGN_PROCESS_TO_JOB option but do not assign all UCMD Server child processes to a job object.

Default

There is no UCMD Manager default; the default value is specified by the UCMD Server ASSIGN_PROCESS_TO_JOB option.

BIF_DIRECTORY - UCMD Manager configuration option

Description

The BIF_DIRECTORY option specifies the Broker Interface File (BIF) directory where the Universal Broker interface file, **ubroker.bif**, is located.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-bif_directory *directory*		✓		
Environment Variable	UCMDBIFDIRECTORY=*directory*		✓		
Configuration File Keyword	n/a				
STRUCM Parameter	n/a				

Values

directory is the name of the BIF directory.

Default is /var/opt/universal.

CA_CERTIFICATES - UCMD Manager configuration option

Description

The CA_CERTIFICATES option specifies the location of the PEM-formatted trusted Certificate Authority (CA) X.509 certificates file.

Trust CA certificates are required if Universal Broker certificate authentication and verification is desired.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-ca_certs *ddname* or *file*		✓	✓	✓
Environment Variable	UCMDCACERTS=*file*	✓	✓	✓	
Configuration File Keyword	ca_certificates *ddname* or *file*	✓	✓	✓	✓
STRUCM Parameter	CACERTS(*file [lib]*) [CACERTSMBR (*member*)]	✓			

Values

z/OS	*ddname* is the ddname of the X.509 certificates. The value is used only when the SSL_IMPLEMENTATION option is set to *OPENSSL*. Allocated to the ddname must be either a sequential data set or a member of a PDS that has a variable record format.
UNIX	*file* is the path name of the X.509 certificates file. Relative paths are relative the current working directory.
Windows	*file* is the path name of the X.509 certificates file. Relative paths are relative the current working directory.
IBM i	*file* is the qualified file name of the X.509 certificates file. The file name can be qualified by a library name. If not, the library list ***LIBL** is searched for the first occurrence of the file name.

CERTIFICATE - UCMD Manager configuration option

Description

The CERTIFICATE option specifies the file / ddname name of the PEM-formatted X.509 certificate that identifies the UCMD Manager.

A UCMD Manager X.509 certificate is required if the Universal Broker requires client authentication.

Note

If the CERTIFICATE option is used, the PRIVATE_KEY option is required.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-cert *ddname* or *file*		✓	✓	✓
Environment Variable	UCMDCERT=*file*	✓	✓	✓	
Configuration File Keyword	certificate *ddname* or *file*	✓	✓	✓	✓
STRUCM Parameter	CERT(*file [lib]*) [CERTMBR (*member*)]	✓			

Values

z/OS	*ddname* is the ddname of the X.509 certificate. The value is used only when the SSL_IMPLEMENTATION option is set to *OPENSSL*. Allocated to the ddname must be either a sequential data set or a member of a PDS that has a variable record format.
UNIX	*file* is the path name of the X.509 certificate file. Relative paths are relative to the current working directory.
Windows	*file* is the path name of the X.509 certificate file. Relative paths are relative to the current working directory.
IBM i	*file* is the qualified file name of the X.509 certificate file. The file name can be qualified by a library name. If not, the library list ***LIBL** is searched for the first occurrence of the file name.

CERTIFICATE_EXPIRATION_NOTICE - UCMD Manager configuration option

Description

The CERTIFICATE_EXPIRATION_NOTICE option specifies the number of days prior to certificate expiration to begin issuing informational messages about the expiration.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-days *number*		✓	✓	✓
Environment Variable	UCMDDAYSTILEXPIRE=*number*		✓	✓	
Configuration File Keyword	days_til_expire *number*		✓	✓	✓
STRUCM Parameter	n/a				

z/OS

For OpenSSL only; not implemented for SystemSSL.

Values

number is the number of days prior to certificate expiration to begin issuing informational messages about the expiration.

Default is 15.

CERTIFICATE_REVOCATION_LIST - UCMD Manager configuration option

Description

The CERTIFICATE_REVOCATION_LIST option specifies the file name / ddname of the PEM-formatted file containing the Certificate Revocation List (CRL) issued by the trusted Certificate Authority.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-crl *file* or *ddname*		✓	✓	✓
Environment Variable	UCMDCRL=*file*	✓	✓	✓	
Configuration File Keyword	crl *file* or *ddname*	✓	✓	✓	✓
STRUCM Parameter	CRLFILE(*file [lib]*) [CRLMBR(*member*)]	✓			

Values

z/OS	*ddname* is the ddname of the file containing the CRL. The value is used only when the SSL_IMPLEMENTATION option is set to *OPENSSL* .
UNIX	*file* is the path name of the file containing the CRL. Relative paths are relative to the current working directory.
Windows	*file* is the path name of the file containing the CRL. Relative paths are relative to the current working directory.
IBM i	*file* is the qualified file name of the CRL file. The file name can be qualified by a library name. If not, the library list ***LIBL** is searched for the first occurrence of the file name.

CODE_PAGE - UCMD Manager configuration option

Description

The CODE_PAGE option specifies the character code page that is used to translate text data received and transmitted over the network.

The Universal Translate Table (UTT) files are used to translate between Unicode and the local single-byte code page.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-t *codepage*		✓	✓	✓
Command Line, Long Form	-codepage *codepage*		✓	✓	✓
Environment Variable	UCMDCODEPAGE=*codepage*	✓	✓	✓	
Configuration File Keyword	codepage *codepage*	✓	✓	✓	✓
STRUCM Parameter	CODEPAGE(*codepage*)	✓			

Values

codepage is the character code page that is used to translate data.

codepage references a Universal Translate Table (UTT) file provided with the product. UTT files are used to translate between Unicode and the local single-byte code page. (All UTT files end with an extension of **.utt**.)

Note

UTF-8 is not a supported *codepage* value for CODE_PAGE. UTF-8 codepage is valid only for standard I/O text file translation. Consequently, it can be specified only with the SIO_LOCAL_CODE_PAGE and SIO_REMOTE_CODE_PAGE options.

Default

Default is different for different operating systems:

- ISO8859-1 (8-bit ASCII): ASCII-based operating systems
- IBM1047 (EBCDIC): EBCDIC-based operating system

See Character Code Pages for a complete list of character code pages provided by Stonebranch Inc. for use with Universal Agent.

COMMAND - UCMD Manager configuration option

Description

The COMMAND option specifies the command to execute on the remote computer.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-c *command*		✓	✓	✓
Command Line, Long Form	-cmd *command*		✓	✓	✓
Environment Variable	UCMDCMD=*command*	✓	✓	✓	
Configuration File Keyword	n/a				
STRUCM Parameter	CMD(*command*)	✓			

Values

command is the command to be executed.

It can be any command that is valid for command line execution on the remote computer's operating system.

IBM i	If the command contains spaces, it must be enclosed in single (**'**) quotation marks.
Windows	If the command contains spaces, it must be enclosed in double (**"**) quotation marks.
UNIX	If the command contains spaces, it must be enclosed in single (**'**) or double (**"**) quotation marks.
z/OS	If the command contains spaces, it must be enclosed in single (**'**) or double (**"**) quotation marks.

z/OS

command can be continued onto multiple lines. No continuation character is required. *command* is read starting at the first enclosing quotation mark to the ending enclosing quotation mark.

If an enclosing character is part of the command, use two consecutive characters to produce one in the command.

The remote command shell processes command meta-characters (for example, redirection **<** and **>** and pipes **|**) as normal on the remote computer.

COMMAND_FILE_ENCRYPTED - UCMD Manager configuration option

Description

The COMMAND_FILE_ENCRYPTED option specifies the file / ddname containing encrypted values for command line options.

Command files specify an additional source of command line options. Storing options in a file can be used in situations where it is not desirable to explicitly specify them on the command line. The options read from the file are processed exactly like options specified on the command line. The options must be in their respective command line formats.

UCMD Manager can process command files that are either encrypted or in plain text (see the COMMAND_FILE_PLAIN option). Encrypted command files are an excellent place to store sensitive data such as user IDs and passwords. Command files (encrypted or not) that contain sensitive data should be protected from unauthorized read access with a security system, such as RACF.

Use the Universal Encrypt utility to encrypt a plain text command file (see Universal Encrypt). If a key was used to encrypt the file, the same key must be supplied using the ENCRYPTION_KEY option.

Note

If a ddname / file is specified in this option, it should not be specified additionally in the COMMAND_FILE_PLAIN option. If it is, the ddname/ file specified in COMMAND_FILE_PLAIN will be used.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-x *ddname* or *file*		✓	✓	✓
Command Line, Long Form	-encryptedfile *ddname* or *file*		✓	✓	✓
Environment Variable	UCMDENCRYPTEDFILE=*file*	✓			
Configuration File Keyword	n/a				
STRUCM Parameter	ECMFILE(*file*) [ECMMBR(*member*)]	✓			

Values

ddname or *file* is the name of the ddname or file, respectively, containing the encrypted command parameter values.

COMMAND_FILE_PLAIN - UCMD Manager configuration option

Description

The COMMAND_FILE_PLAIN option specifies the ddname (for z/OS) or file containing plain text values for command line options.

Command files specify an additional source of command line options. Storing options in a file can be used in situations where it is not desirable to explicitly specify them on the command line. The options read from the file are processed exactly like options specified on the command line. The options must be in their respective command line formats.

UCMD Manager can process command files that are either in plain text or encrypted (see the COMMAND_FILE_ENCRYPTED option). It is strongly recommended that plain text files be further protected from unauthorized access using a native operating system security method, such as RACF.

Note

If a ddname / file is specified in this option, it should not be specified additionally in the COMMAND_FILE_ENCRYPTED option. If it is, the ddname / file specified in COMMAND_FILE_PLAIN will be used.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-f *ddname* or *file*		✓	✓	✓
Command Line, Long Form	-file *ddname* or *file*		✓	✓	✓
Environment Variable	UCMDFILE=*file*	✓			
Configuration File Keyword	n/a				
STRUCM Parameter	CMDFILE(*file*) [CMDMBR(*member*)]	✓			

Values

ddname (for z/OS) or *file* (for IBM i and UNIX) is the name of the ddname or file name, respectively, containing the parameters and their values.

COMMAND_ID - UCMD Manager configuration option

Description

The COMMAND_ID option specifies an ID (identifier) that is used to identify the unit of work represented by the UCMD Manager, UCMD Server, and user command.

The command ID is saved by the Universal Broker to help associate a UCMD Server component with the UCMD Manager that requested it to execute.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-C *id*		✓	✓	✓
Command Line, Long Form	-cmdid *id*		✓	✓	✓
Environment Variable	UCMDCMDID *id*	✓	✓	✓	
Configuration File Keyword	command_id *id*	✓	✓	✓	✓
STRUCM Parameter	CMDID(*id*)	✓			

Values

id can be any value.

If *id* is an asterisk (*****), the UCMD Manager will generate a unique command ID.

Generated command IDs have the format **WORKID-DATE-TIME-RAND**, where:

- **WORKID** is the UCMD Manager's work ID
- **DATE** is the current date, in YYMMDD format
- **TIME** is the current time, in HHMM format
- **RAND** is a randomly-generated 4-byte value, represented as an 8-character hexadecimal string.

IBM i	If *id* contains non-alphanumeric characters (including spaces), it must be enclosed in single (**'**) quotation marks. To include a single quotation mark in the command ID, use two single quotation marks (**''**).
HP NonStop	If *id* contains spaces, it must be enclosed in single (**'**) or double (**"**) quotation marks.
Windows	If *id* contains spaces, it must be enclosed in double (**"**) quotation marks.
UNIX	If *id* contains spaces, it must be enclosed in single (**'**) or double (**"**) quotation marks.
z/OS	If *id* contains spaces, it must be enclosed in single (**'**) or double (**"**) quotation marks.

If no value is saved for COMMAND_ID in the UCMD Manager configuration file, *id* is set to the value of either:

- COMMAND option
- SCRIPT_FILE option

COMMAND_TYPE - UCMD Manager configuration option

Description

The COMMAND_TYPE option identifies the type of command specified by the COMMAND option.

The UCMD Server uses the COMMAND_TYPE value to identify how the COMMAND value is to be processed.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-cmd_type *type*		✓	✓	✓
Environment Variable	UCMDCMDTYPE=*type*	✓	✓	✓	
Configuration File Keyword	cmd_type *type*	✓	✓	✓	✓
STRUCM Parameter	CMDTYPE(*type*)	✓			

Values

type is the command type.

Valid values for *type* on the UCMD Server operating system are:

Command Type	**IBM i**	**UNIX**	**Windows**	**z/OS**
cmd	✓			
cmdref	✓	✓	✓	✓
shell		✓	✓	✓
rexx	✓			
stc				✓

Defaults

- cmd is the default command type for IBM i.
- shell is the default command type for UNIX, Windows, and z/OS.

COMMENT - UCMD Manager configuration option

Description

The COMMENT option specifies a user-defined string that can contain any value.

This string is shown in lists of active Universal Agent components, such as those displayed by the Universal Query utility or I-Activity Monitor.

For example, COMMENT could be used to provide a brief description of the UCMD Manager process.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-comment *text*		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCM Parameter	COMMENT(*user-defined string*)	✓			

Values

text is the user-defined string.

CONNECT_RETRY_COUNT - UCMD Manager configuration option

Description

The CONNECT_RETRY_COUNT option specifies the number of times a UCMD Manager will attempt to establish a connection to a remote Universal Broker that resides on the host(s) specified by the REMOTE_HOST option.

UCMD Manager uses CONNECT_RETRY_COUNT at two different places within its start-up processing. It first uses the option as it tries to establish the initial connection to a remote Universal Broker. It also uses the option as it attempts to create each of the three connections used to forward standard I/O redirection sockets via the Broker to the UCMD Server.

The precise behavior as UCMD Manager tries to create the initial connection to the remote Broker depends on the number of hosts specified by the REMOTE_HOST option.

- When only a single host is provided and the UCMD Manager's attempt to establish a connection to the Universal Broker installed on that system fails, the UCMD Manager will continue to retry the connection attempt until a connection is established or until the number of attempts exceeds the value specified for this option.
- When multiple hosts are provided, the UCMD Manager will attempt to connect to the Universal Broker on each host until a connection is established on one of the hosts or the number of attempts exceeds the value specified for this option. If the CONNECT_RETRY_COUNT value is greater than the number of hosts specified, the UCMD Manager will try each host in the list and then wrap around to the start of the list to try each host again. In other words, CONNECT_RETRY_COUNT specifies the total number of connection attempts, rather than the number of attempts that should be made for each host.

UCMD Manager also uses CONNECT_RETRY_COUNT when it attempts to connect to the remote Broker in order to forward the standard I/O redirection sockets to the UCMD Server. At this point, the UCMD Manager only attempts additional connections to the Broker to which it is already connected. An additional connection is required to forward each of the three standard I/O redirection sockets. If any of those connection attempts fail, the UCMD Manager will retry the attempt until a connection is established or the number of attempts exceeds the specified retry count.

Note

This option should not be confused with the RECONNECT_RETRY_COUNT option, which controls behavior when network fault tolerance is enabled and the UCMD Manager loses an established connection to a remote UCMD Server.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-connect_retry_count *number*		✓	✓	✓
Environment Variable	UCMDCONNECTRETRYCOUNT=*number*		✓	✓	✓
Configuration File Keyword	connect_retry_count *number*		✓	✓	✓
STRUCM Parameter	n/a				

Values

number is the number of times the UCMD Manager will try to establish a connection with a remote Universal Broker residing on a host specified with the REMOTE_HOST option.

Valid values for *number* are 0 (zero) to 20.

When the value for this option is 0 (zero), the UCMD Manager will not retry failed connection attempts.

Default is 0.

CONNECT_RETRY_INTERVAL - UCMD Manager configuration option

Description

The CONNECT_RETRY_INTERVAL option specifies the number of seconds a UCMD Manager will wait before it retries a failed attempt to establish a connection to a remote Universal Broker that resides on the host(s) specified by the REMOTE_HOST option.

UCMD Manager only uses this option when the value specified for the CONNECT_RETRY_COUNT option is greater than 0 (zero).

UCMD Manager uses the connection retry options at two different places within its start-up processing. It first uses the option as it tries to establish the initial connection to a remote Universal Broker. It also uses the option as it attempts to create each of the three connections used to forward standard I/O redirection sockets via the Broker to the UCMD Server.

The precise behavior as UCMD Manager tries to create the initial connection to the remote Broker depends on the number of hosts specified by the REMOTE_HOST option.

- When only a single host is provided, and the UCMD Manager's attempt to establish a connection to the Universal Broker installed on that system fails, the UCMD Manager will retry the connection attempt as soon as the retry interval elapses.
- When multiple hosts are specified, the UCMD Manager will try each host in the list as soon as the attempt to connect to the Universal Broker on the preceding host fails. However, once a connection attempt to a host in the list has been tried, the UCMD Manager will not retry that host until the retry interval expires.

UCMD Manager also uses CONNECT_RETRY_INTERVAL when it attempts to connect to the remote Broker in order to forward the standard I/O redirection sockets to the UCMD Server. At this point, the UCMD Manager only attempts additional connections to the Broker to which it is already connected. An additional connection is required to forward each of the three standard I/O redirection sockets. If any of those connection attempts fail, the UCMD Manager will retry the attempt as soon as the retry interval elapses.

In all cases, the retry attempts continue until a connection is established or the number of attempts exceeds the value specified for the CONNECT_RETRY_COUNT option.

Note

⚠ CONNECT_RETRY_INTERVAL should not be confused with the RECONNECT_RETRY_INTERVAL option, which controls behavior when network fault tolerance is enabled and the UCMD Manager loses an established connection to a remote UCMD Server.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-connect_retry_interval *seconds*		✓	✓	✓
Environment Variable	UCMDCONNECTRETRYINTERVAL=*seconds*		✓	✓	✓
Configuration File Keyword	connect_retry_interval *seconds*		✓	✓	✓
STRUCM Parameter	n/a				

Values

seconds is the number of seconds the UCMD Manager will wait before it retries an attempt to establish a connection with a remote Universal Broker that resides on a host specified with the REMOTE_HOST option.

Valid values for *seconds* are 1 to 60.

Default is 0.

CONNECT_TIMEOUT - UCMD Manager configuration option

Description

The CONNECT_TIMEOUT option specifies how long a UCMD Manager will wait for a connection to a remote Universal Broker to complete.

CONNECT_TIMEOUT is particularly helpful when more than one host is specified by the REMOTE_HOST option. By default, connection time-outs are controlled by the TCP/IP stack. Depending on this value, it may take several minutes to process a list of hosts before a connection actually succeeds. Setting a CONNECT_TIMEOUT value allows connection attempts to unreachable Universal Brokers to fail quickly, decreasing the time required to process a list of one or more hosts.

Note

CONNECT_TIMEOUT is most beneficial when set to a value that is less than the TCP/IP stack's default timeout, which is implementation dependent. A relatively small CONNECT_TIMEOUT value is recommended, to make sure it - and not the TCP/IP default - is applied.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-connect_timeout *seconds*		✓	✓	✓
Environment Variable	UCMDCONNECTTIMEOUT=*seconds*	✓	✓	✓	
Configuration File Keyword	connect_timeout *seconds*	✓	✓	✓	✓
STRUCM Parameter	CONNECTTO (*seconds*)	✓			

Values

seconds is the time, in seconds, that the UCMD Manager will wait for a connection to a Universal Broker to complete. This value applies to each host contained in the resolved, expanded, and scrubbed REMOTE_HOST list.

Valid values for *seconds* are 0 (zero) to 300.

If the time specified by *seconds* elapses before a successful connection to a Universal Broker on the specified system is established, the UCMD Manager will select the next host in the list. If no more hosts are available, the application will end with an error to indicate that no connection was made.

Note

It is possible for the total time required to attempt connections to all hosts in the REMOTE_HOST list to exceed the number of seconds specified in this option.

Default is 0.

(This mean that each connection attempt will use the implementation-defined TCP/IP time-out value. This is the behavior of connection attempts prior to version 3.2.0 of Universal Command.)

CTL_SSL_CIPHER_LIST - UCMD Manager configuration option

Description

The CTL_SSL_CIPHER_LIST option specifies one or more SSL/TLS cipher suites that are acceptable to use for network communications on the control session, which is used for component internal communication.

CTL_SSL_CIPHER_LIST also can be used to disable the SSL/TLS protocol.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-ctl_ssl_cipher_list *cipherlist*		✓	✓	✓
Environment Variable	UCMDCTLSSLCIPHERLIST=*cipherlist*	✓	✓	✓	
Configuration File Keyword	ctl_ssl_cipher_list *cipherlist*	✓	✓	✓	✓
STRUCM Parameter	CTLCPHRLST(*cipherlist*)	✓			

Values

cipherlist is a comma-separated list of SSL/TLS cipher suites. The following table identifies the list of SSL/TLS cipher suites supported for this option.

The list is in default order, with the most preferred suite first and the least preferred suite last.

Cipher Suite Name	**Description**
AES256-GCM-SHA384	256-bit AES encryption in Galois Counter Mode, SHA-2 384-bit message digest.
AES256-SHA	256-bit AES encryption with SHA-1 message digest.
AES128-GCM-SHA256	128-bit AES encryption in Galois Counter Mode, SHA-2 256-bit message digest.
AES128-SHA	128-bit AES encryption with SHA-1 message digest.
RC4-SHA	128-bit RC4 encryption with SHA-1 message digest.
RC4-MD5	128-bit RC4 encryption with MD5 message digest.
DES-CBC3-SHA	128-bit Triple-DES encryption with SHA-1 message digest.
DES-CBC-SHA	128-bit DES encryption with SHA-1 message digest. Note ⚠ As of Universal Agent 6.7.0.0, DES-CBC-SHA is supported only on HP-UX. Additionally, any Agents on HP-UX that accept connections from, or attempt connections to, Agents on other platforms must be configured with at least one currently supported cipher suite besides DES-CBC-SHA. Therefore, those HP-UX Agents cannot be configured only with DES-CBC-SHA in their list of cipher suites.
NULL-SHA256	No encryption and SHA-2 256-bit message digest.

NULL-SHA	No encryption and SHA-1 message digest.
NULL-MD5	No encryption and MD5 message digest.
NULL-NULL	No encryption, no data authentication, SSL is not used; instead, Universal V2 Protocol (**UNVv2**) is used.

A single value of NULL-NULL instead of the list disables the SSL/TLS protocol. The legacy Universal Products (UNVv2) protocol without encryption and message authentication is used instead of SSL/TLS.

No data privacy or data integrity is provided with the UNVv2 network communications protocol.

NULL-NULL can be specified if the UCMD Server ENCRYPT_CONTROL_SESSION configuration option value is **no**.

DATA_AUTHENTICATION - UCMD Manager configuration option

Description

The DATA_AUTHENTICATION option specifies whether or not all data sent over the network is authenticated when using the **UNVv2** protocol.

Generating a checksum value for each data block performs authentication. The checksum value is sent with the data block. The receiver generates a second checksum value for the data block. If the checksum values are not equal, the authentication fails. Failed authentication closes the network connection.

The checksum is generated with the MD5 Message Digest Algorithm by RSA Data Security, Inc.

DATA_AUTHENTICATION does not have any effect on the SSL/TLS protocol. See the DATA_SSL_CIPHER_LIST option for SSL/TLS data authentication.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-a *option*		✓	✓	✓
Command Line, Long Form	-authenticate *option*		✓	✓	✓
Environment Variable	UCMDAUTHENTICATE=*option*	✓	✓	✓	
Configuration File Keyword	authenticate *option*	✓	✓	✓	✓
STRUCM Parameter	AUTH(**option*)	✓			

Values

option is the specification for whether or not data is authenticated.

Valid values for *option* are:

- **yes**
  Data authentication is required for the **UNVv2** protocol. All network data transfers are authenticated regardless of UCMD Server's DATA_AUTHENTICATION option.
- **no**
  Data authentication is not required. However, the UCMD Server still can request data authentication via its DATA_AUTHENTICATION option.

Default is no.

DATA_COMPRESSION - UCMD Manager configuration option

Description

The DATA_COMPRESSION option specifies whether or not the data in standard I/O file transmissions across the network should be compressed.

Optionally, it also can specify the compression method to use.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-k *option[,method]*		✓	✓	✓
Command Line, Long Form	-compress *option[,method]*		✓	✓	✓
Environment Variable	UCMDCOMPRESS=*option[,method]*	✓	✓	✓	
Configuration File Keyword	compress *option[,method]*	✓	✓	✓	✓
STRUCM Parameter	COMPRESS(_*option_) [CMPRSMTH(,_*method)_]	✓			

Values

option is either of the following values:

- **yes**
  Data compression is required. All data in standard I/O file transmissions is compressed regardless of the UCMD Server DATA_COMPRESSION option value.
- **no**
  Data compression is not required. However, data compression still can be requested via the UCMD Server DATA_COMPRESSION option.

Default is no.

method is either of the following values:

- **zlib**
  Data is compressed using ZLIB compression algorithm. This method usually results in a very high compression rate, but tends to be somewhat CPU-intensive. It is recommended in environments where controlling a process's CPU usage is not necessarily a priority.
- **hasp**
  Data is compressed using the HASP compression algorithm. This method is less CPU-intensive than the ZLIB method. It is recommended in environments where controlling CPU usage is a priority. With HASP, the compression rate, while still very good, tends to be a little less than what is possible with the ZLIB.

Default is zlib.

DATA_ENCRYPTION - UCMD Manager configuration option

Description

The DATA_ENCRYPTION option specifies whether or not all data sent over the network is encrypted.

Encryption protects the privacy of the data. **UNVv2** data encryption uses one of several encryption algorithms, such as the Data Encryption Standard (DES) algorithm.

SSL/TLS data encryption uses one of the SSL/TLS cipher suites specified with the DATA_SSL_CIPHER_LIST option.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-e *option*		✓	✓	✓
Command Line, Long Form	-encrypt *option*		✓	✓	✓
Environment Variable	UCMDENCRYPT=*option*	✓	✓	✓	
Configuration File Keyword	encrypt *option*	✓	✓	✓	✓
STRUCM Parameter	ENCRYPT(**option*)	✓			

Values

option is the specification for whether or not data is encrypted.

Valid values for *option* are:

- **yes**
  Data encryption is required. All network data transfers are encrypted regardless of UCMD Server's DATA_ENCRYPTION option.
- **no**
  Data encryption is not required. However, UCMD Server still can request data encryption via its DATA_ENCRYPTION option.

Default is no.

DATA_SSL_CIPHER_LIST - UCMD Manager configuration option

Description

The DATA_SSL_CIPHER_LIST option specifies one or more SSL/TLS cipher suites that are acceptable to use for network communications on the data session, which is used for standard I/O file transmission.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-data_ssl_cipher_list *cipherlist*		✓	✓	✓
Environment Variable	UCMDDATASSLCIPHERLIST=*cipherlist*	✓	✓	✓	
Configuration File Keyword	data_ssl_cipher_list *cipherlist*	✓	✓	✓	✓
STRUCM Parameter	DTACPHRLST(*cipherlist*)	✓			

Values

cipherlist is a comma-separated list of SSL/TLS cipher suites. The following table identifies the list of SSL/TLS cipher suites supported for this option.

The list is in default order, with the most preferred suite first and the least preferred suite last.

Cipher Suite	Description
AES256-GCM-SHA384	256-bit AES encryption in Galois Counter Mode, SHA-2 384-bit message digest.
AES256-SHA	256-bit AES encryption with SHA-1 message digest.
AES128-GCM-SHA256	128-bit AES encryption in Galois Counter Mode, SHA-2 256-bit message digest.
AES128-SHA	128-bit AES encryption with SHA-1 message digest.
RC4-SHA	128-bit RC4 encryption with SHA-1 message digest.
RC4-MD5	128-bit RC4 encryption with MD5 message digest.
DES-CBC3-SHA	128-bit Triple-DES encryption with SHA-1 message digest.
DES-CBC-SHA	128-bit DES encryption with SHA-1 message digest. Note As of Universal Agent 6.7.0.0, DES-CBC-SHA is supported only on HP-UX. Additionally, any Agents on HP-UX that accept connections from, or attempt connections to, Agents on other platforms must be configured with at least one currently supported cipher suite besides DES-CBC-SHA. Therefore, those HP-UX Agents cannot be configured only with DES-CBC-SHA in their list of cipher suites.
NULL-SHA256	No encryption and SHA-2 256-bit message digest.
NULL-SHA	No encryption and SHA-1 message digest.

NULL-MD5	No encryption and MD5 message digest.

Note

To create an unencrypted data session, you must use the DEFAULT_CIPHER configuration option.

DEFAULT_CIPHER - UCMD Manager configuration option

Description

The DEFAULT_CIPHER option specifies the SSL/TLS cipher suite to be used for data sessions when the DATA_ENCRYPTION configuration option is set to **no**.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-default_cipher *cipher*		✓	✓	✓
Environment Variable	UCMDDEFAULTCIPHER=*cipher*	✓	✓	✓	
Configuration File Keyword	default_cipher *cipher*	✓	✓	✓	✓
STRUCM Parameter	DFTCPHR(*cipher*)	✓			

Values

cipher is the SSL/TLS cipher suite to be used.

The following table identifies the list of SSL/TLS cipher suites supported for this option.

Cipher Suite	**Description**
AES256-GCM-SHA384	256-bit AES encryption in Galois Counter Mode, SHA-2 384-bit message digest.
AES256-SHA	256-bit AES encryption with SHA-1 message digest.
AES128-GCM-SHA256	128-bit AES encryption in Galois Counter Mode, SHA-2 256-bit message digest.
AES128-SHA	128-bit AES encryption with SHA-1 message digest.
RC4-SHA	128-bit RC4 encryption with SHA-1 message digest.
RC4-MD5	128-bit RC4 encryption with MD5 message digest.
DES-CBC3-SHA	128-bit Triple-DES encryption with SHA-1 message digest.
DES-CBC-SHA	128-bit DES encryption with SHA-1 message digest. Note ⚠ As of Universal Agent 6.7.0.0, DES-CBC-SHA is supported only on HP-UX. Additionally, any Agents on HP-UX that accept connections from, or attempt connections to, Agents on other platforms must be configured with at least one currently supported cipher suite besides DES-CBC-SHA. Therefore, those HP-UX Agents cannot be configured only with DES-CBC-SHA in their list of cipher suites.
NULL-SHA256	No encryption and SHA-2 256-bit message digest.
NULL-SHA	No encryption and SHA-1 message digest.
NULL-MD5	No encryption and MD5 message digest.

NULL-NULL	No encryption, no data authentication, SSL/TLS is not used; instead, Universal V2 Protocol (**UNVv2**) is used.

Default is NULL-MD5.

DNS_EXPAND - UCMD Manager configuration option

Description

The DNS_EXPAND option specifies how many IP addresses are returned to UCMD Manager following a DNS query, which is issued to resolve a host name.

If the UCMD Manager is configured to expand the results of the query, all IP addresses defined for a particular host name are returned and expanded (in-place) within the list of hosts specified for the REMOTE_HOST option. Otherwise, only the first host is returned, and no expansion is performed.

For example, if a host list contains six host names, and the name in the 3rd position resolves to five IP addresses, those addresses will occupy positions 3-7 in the expanded list. Subsequent hosts specified by the user will begin at the 8th position in the expanded list.

That is:

- If the REMOTE_HOST list contains:
  -host *host1,host2,host3,host4,host5,host6*

- And *host3* resolves to:
  ip3a, ip3b, ip3c, ip3d, and ip3e

- Then after the other hosts are resolved, the list will be expanded to:
  -host *ip1,ip2,ip3a,ip3b,ip3c,ip3d,ip3e,ip4,ip5,ip6*

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-dns_expand *option*		✓	✓	✓
Environment Variable	UCMDDNSEXPAND=*option*	✓	✓	✓	
Configuration File Keyword	dns_expand *option*	✓	✓	✓	✓
STRUCM Parameter	DNSEXPAND(**option*)	✓			

Values

option specifies whether or not DNS query results are expanded.

Valid values for *option* are:

- **yes**
  All IP addresses returned by a DNS for a given query are returned.
- **no**
  Only the first IP address returned by a DNS for a given query is returned.

Default is no.

ELEVATE_USER_TOKEN - UCMD Manager configuration option

Description

The ELEVATE_USER_TOKEN option allows a process to execute with the highest privileges available to it; the process will execute with an elevated user token (that is, one not subject to User Account Control (UAC) restrictions).

Note

This UCMD Manager ELEVATE_USER_TOKEN option overrides the UCMD Server for Windows ELEVATE_USER_TOKEN configuration option. That option is available only for Windows UCMD Servers; it is ignored on all other Server platforms.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-elevate_user_token *option*		✓	✓	✓
Environment Variable	UCMDELEVATEUSERTOKEN=*option*		✓	✓	✓
Configuration File Keyword	n/a				
STRUCM Parameter	n/a				

Values

option specifies whether or not UCMD Server child processes execute with an elevated user token.

Valid values for *option* are:

- **yes**
  Override the UCMD Server ELEVATE_USER_TOKEN option and execute the UCMD Server for Windows child process with an elevated user token (that is, one not subject to User Account Control (UAC) restrictions).
- **no**
  Override the UCMD Server ELEVATE_USER_TOKEN option and execute the UCMD Server for Windows child process with a default user access token, which may or may not already be elevated. If it is not an elevated token, some operations subject to UAC restrictions may fail.

Default

There is no UCMD Manager default for this option; it is provided only to override the configured value for the UCMD Server ELEVATE_USER_TOKEN configuration option.

ENCRYPTION_KEY - UCMD Manager configuration option

Description

The ENCRYPTION_KEY option specifies the key used to encrypt the command file (see COMMAND_FILE_ENCRYPTED).

This key acts much like a password for the encrypted command file in that it can be used to protect the file from decryption by unauthorized users.

If a key was used to encrypt a command file (when Universal Encrypt was run), that same key must be specified to decrypt the file, or the decryption will fail.

If no key is specified, the default key is used.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-K key		✓	✓	✓
Command Line, Long Form	-key key		✓	✓	✓
Environment Variable	UCMDKEY=*key*	✓			
Configuration File Keyword	n/a				
STRUCM Parameter	KEY(*key*)	✓			

Values

key is the key used to encrypt the command file.

EVENT_GENERATION - UCMD Manager configuration option

Description

The EVENT_GENERATION option specifies which types of events are to be generated and processed as persistent events by the Universal Event Subsystem (UES).

A persistent event record is saved in a Universal Enterprise Controller (UEC) database, the UES database (***uec.evm.db***), for long-term storage.

For a list of all event types for all Universal Agent components, see Event Definition Details.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	n/a				
Environment Variable	n/a				
Configuration File Keyword	event_generation *types*	✓	✓	✓	✓
STRUCM Parameter	n/a				

Values

type specifies a comma-separated list of event types. It allows for all or a subset of all potential event message types to be selected.

Event type ranges can be specified by separating the lower and upper range values with a dash (**-**) character.

Event types can be selected for inclusion or exclusion:

- Exclusion operator is **X** or **x**.
- An asterisk (*) represents all event types.

Examples

100,101,102	Generate event types 100, 101, and 102.
100-102	Generate event types 100 through 102.
100-102,200	Generate event types 100 through 102 and 200.
*	Generate all event types.
*,X100	Generate all event types except for 100.
X*	Generate no event types.
*,X200-250,X300	Generate all event types except for 200 through 250 and 300.

Default is X* (no event types).

EXIT_CODE_MAP - UCMD Manager configuration option

Description

The EXIT_CODE_MAP option allows one or more exit codes from the user process executed by Universal Command Server to be translated (mapped) to a corresponding exit code for Universal Command Manager.

This overrides the default behavior of the UCMD Manager, which would normally exit with the same value as the user process.

EXIT_CODE_MAP is provided to simplify dependent processing in environments where exit code rules are enforced and/or where remote exit code values are invalid on the Manager platform.

IBM i

IBM i implementation of mapped exit codes is to issue an exception (escape) message that corresponds to the mapped exit code number; that is, the exit code is mapped to the severity of the IBM i exception message. However, if the mapped exit code is greater than 99, it is mapped to 99 (the highest allowed severity).

Before posting the exception message, UCMD posts an informational message, UNV2582, that explains mapped exit codes. Also, the original messages posted upon exit from UCMD are now issued as diagnostic messages. The new exception messages began with UCMFF to avoid collision with normal Universal Agent messages.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-exit_code_map *map*		✓	✓	✓
Environment Variable	UCMDEXITCODEMAP=*map*	✓	✓	✓	
Configuration File Keyword	exit_code_map *map*	✓	✓	✓	✓
STRUCM Parameter	EXITCDMAP(*option*)	✓			

Values

map is a comma-separated list of rules in the following format:

[status]range:exitcode[,[status]range:exitcode,...,[status]range:exitcode]

In this format:

- *status* is the termination status of the user process.
- *range* is the range of user process exit codes to which the rule applies.
- *exitcode* is the value with which the UCMD Manager will exit.

Valid values for *status* are:

- *a* (abnormal)
- *n* (normal)

If a value for *status* is not specified, the rule will apply to all termination types.

The exit codes mapped by a rule can consist of a single value or a range of values.

- A hyphen (**-**) can be used to define a finite, inclusive range of exit codes. For example, a range of 1-5 will map all exit codes from 1 through 5, inclusive.
- A greater than (**>**) or less than (**<**) symbol can be used to define an open-ended range of exit codes, where the value specified for range sets the lower limit and upper limit, respectively, of the range. For example, a range of >1 maps all user process exit codes greater than 1, while a range of <1 maps all exit codes with a value less than 1.
- An asterisk (*****) can be specified for range, which defines a mapping that applies to all user process exit codes for the given termination status (if specified).
- Negative values can be specified for the exit code(s) specified by range. (Negative values are not supported for *exitcode*.)
- For readability, spaces are allowed in the value specified for map, but will be ignored by UCMD Manager.

When entered from the command line, *map* should be enclosed in double (**"**) or single
(**'**) quotes if any of the mapping rules define an open-ended range. This will prevent the greater than (**>**) and less than (**<**) symbols from being interpreted by the command shell as an I/O redirection operator.

Exit code mapping entries are processed from left to right. The first entry that matches the termination status and exit code of a user process is the one applied.

Examples

Example 1

-exit_code_map "a:16"*

In this example, the map is interpreted as:

If the user process ends abnormally, the UCMD Manager exits with a value of 16.

If the user process ends normally, the UCMD Manager exits with the same value returned by the user process.

Example 2

-exit_code_map "1-5:4, a6-10:1,n6-10:2, >15:8, <20:12"

In this example, the map is interpreted as:

- If the user process exits with a value of 1 through 5, inclusive, the UCMD Manager exits with a value of 4.
- If the user process ends abnormally with a value of 6 through 10, inclusive, the UCMD Manager exits with a value of 1.
- If the user process ends normally with a value of 6 through 10, inclusive, the UCMD Manager exits with a value of 2.
- If the user process ends with a value greater than 15, the UCMD Manager exits with a value of 8.
- If the user process ends with a value less than 20, the UCMD Manager exits with a value of 12.
- If the user process exits with any other value, the UCMD Manager also exits with that value.

Example 3

-exit_code_map "a:16,n*:0,>0:4"*

In this example, the map is interpreted as:

- If the user process ends abnormally, the UCMD Manager exits with a value of 16.
- If the user process ends normally, the UCMD Manager exits with a value of 0.

By definition, the termination status of a process is either abnormal or normal. All exit codes returned by the user process will match one of these first two entries. The third entry in the map, which translates any user process exit code greater than 0 (zero) to a UCMD Manager exit code of 4, is ignored.

FORCE_COMPLETE - UCMD Manager configuration option

Description

The FORCE_COMPLETE option specifies whether or not to force a manager fault tolerant UCMD Server that is in a **PENDING** communication state into a **COMPLETE** state without retrieving the spooled files.

FORCE_COMPLETE is used in the case where a UCMD Manager that was started in manager fault tolerant mode was stopped, which left the UCMD Server executing.

After the user job and the UCMD Server ended, the UCMD Server entered a **PENDING** communication state. The UCMD Server remains **PENDING** until a UCMD Manager is restarted in order to retrieve the spooled data and exit information.

If, for some reason, you do not want to restart the UCMD Manager, FORCE_COMPLETE enables you to force the UCMD Server to complete without retrieving the spooled data and exit information.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-forcecomplete *option*		✓	✓	✓
Environment Variable	UCMDFORCECOMPLETE=*option*	✓	✓	✓	
Configuration File Keyword	forcecomplete *option*	✓	✓	✓	✓
STRUCM Parameter	FRCCMPLT(*option*)	✓			

Values

option is the specification for whether or not to force a completion.

Valid values for *option* are:

- **yes**
  Force the UCMD Server to complete.
- **no**
  Do not force the UCMD Server to complete.

Default is no.

HELP - UCMD Manager configuration option

Description

The HELP option displays a description of the Universal Command Manager command line options and their format.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-h		✓	✓	✓
Command Line, Long Form	-help		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCM Parameter	n/a				

Values

(There are no values for the HELP option.)

HOST_SELECTION - UCMD Manager configuration option

Description

The HOST_SELECTION option specifies how the UCMD Manager will select a host, from the list of hosts specified via the REMOTE_HOST option, with which the UCMD Manager will begin its attempts to connect to a remote Universal Broker.

Regardless of how the first host is selected, UCMD Manager processes the list sequentially until either a connection to a remote Universal Broker succeeds or all hosts in the list have been tried.

HOST_SELECTION is ignored if only one host is specified in the REMOTE_HOST list of hosts.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-host_selection *option*		✓	✓	✓
Environment Variable	UCMDHOSTSELECTION=*option*	✓	✓	✓	
Configuration File Keyword	host_selection *option*	✓	✓	✓	✓
STRUCM Parameter	HOSTSELECT(*option)	✓			

Values

option specifies how UCMD Manager will select the first entry within a list of hosts specified by the REMOTE_HOST option.

Valid values for *option* are:

- **sequential**
  UCMD Manager will select the first host in the list, and then proceed through the hosts in the order in which they appear within the list.
- **random**
  UCMD Manager will select any host in the list, in no particular order.

  Note

  ⚠ Attempts to connect to a remote Universal Broker begin with this randomly-selected host, and then proceed in a sequential manner, wrapping around the list (if necessary) until the first host selected is reached again.

Default is sequential.

HOSTNAME_RETRY_COUNT - UCMD Manager configuration option

Description

The HOSTNAME_RETRY_COUNT option specifies the number of times that UCMD will attempt to resolve the host name of a specified Universal Broker before it ends with a connect error.

The UCMD Manager will sleep for one second between resolution attempts.

Connection errors occur for several reasons. A common reason is a failure to resolve the Universal Broker host name specified with the REMOTE_HOST option. This error can occur intermittently due to a temporary resource shortage or a temporary DNS problem. If your system is prone to host name resolution errors, it may help to have UCMD Manager retry the connection several times.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-hostname_retry_count *count*		✓	✓	✓
Environment Variable	UCMDHOSTNAMERETRYCOUNT=*count*	✓	✓	✓	
Configuration File Keyword	hostname_retry_count *count*	✓	✓	✓	✓
STRUCM Parameter	HSTNMRTYCT(*count*)	✓			

Values

count is the number of times that UCMD will attempt to resolve the host name.

Default is*1*.

INSTALLATION_DIRECTORY - UCMD Manager configuration option

Description

The INSTALLATION_DIRECTORY option specifies the location in which UCMD Manager is installed.

Note

This option is required and cannot be overridden.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	n/a				
Environment Variable	n/a				
Configuration File Keyword	installation_directory *directory*		✓	✓	
STRUCM Parameter	n/a				

Values

directory is the location in which UCMD Server is installed.

The full path name is required.

JOB_RETENTION - UCMD Manager configuration option

Description

The JOB_RETENTION option specifies the default number of seconds that a disconnected UCMD Server running with network fault tolerance (see the NETWORK_FAULT_TOLERANT option) waits for a reconnection request from the Manager after the user process has completed.

This situation occurs when the network connection between the UCMD Manager and UCMD Server is lost. Since the UCMD Server is using network fault tolerance, it waits for the UCMD Manager to reestablish the network connections. If the user process ends while the UCMD Server and UCMD Manager are disconnected, JOB_RETENTION specifies how long the UCMD Server should wait for the UCMD Manager to reconnect before it ends. When it ends, the UCMD Manager cannot reconnect.

Note

This option overrides the UCMD Server JOB_RETENTION option.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-j *seconds*		✓	✓	✓
Command Line, Long Form	-job_retention *seconds*		✓	✓	✓
Environment Variable	UCMDJOBRETENTION=*seconds*	✓	✓	✓	
Configuration File Keyword	job_retention *seconds*	✓	✓	✓	✓
STRUCM Parameter	JOBRTN(*seconds*)	✓			

Values

seconds is the number of seconds to wait.

Default

There is no UCMD Manager default for JOB_RETENTION; the default value is specified by the UCMD Server JOB_RETENTION option.

LOGIN - UCMD Manager configuration option

Description

The LOGIN option specifies whether or not the remote command environment is created as if the user account logged into the system.

The differences between a login environment versus a non-login environment depends on the UCMD Server operating system. (See the operating system-specific pages for UCMD Server for complete details.)

Note

Only UCMD Servers of version 2.2.0 and above processes the option.

Windows	A Windows login environment results in the user's specific registry settings to be mapped to the **HKEY_CURRENT_USER** key, and that user's environment block to be loaded.
UNIX	A UNIX login environment invokes the user's shell as a login shell, which reads and executes commands in the system profile and user profile files.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-G *option*		✓	✓	✓
Command Line, Long Form	-login *option*		✓	✓	✓
Environment Variable	UCMDLOGIN=*option*	✓	✓	✓	
Configuration File Keyword	login *option*	✓	✓	✓	✓
STRUCM Parameter	LOGIN(*option*)	✓			

Values

option is the specification for whether or not a login environment is established.

Valid values for *option* are:

- **yes**
  Login environment is established.
- **no**
  Non-login environment is established.

Default is no.

MANAGER_FAULT_TOLERANT - UCMD Manager configuration option

Description

The MANAGER_FAULT_TOLERANT option specifies whether or not the manager fault tolerant feature is used.

See Manager Fault Tolerance - Universal Command for details on manager fault tolerance.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-M *option*		✓	✓	✓
Command Line, Long Form	-managerft *option*		✓	✓	✓
Environment Variable	UCMDMANAGERFT *option*	✓	✓	✓	
Configuration File Keyword	manager_fault_tolerant *option*	✓	✓	✓	✓
STRUCM Parameter	MANAGERFT(**option*)	✓			

Values

option is the specification for whether or not manager fault tolerance is used.

Valid values for *option* are:

- **yes**
  Manager fault tolerant feature is enabled. The COMMAND_ID option is required.
- **no**
  Manager fault tolerant feature is disabled.

Default is no.

MESSAGE_LANGUAGE - UCMD Manager configuration option

Description

The MESSAGE_LEVEL option specifies the Universal Message Catalog (UMC) that is used to format messages.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-L *language*		✓	✓	✓
Command Line, Long Form	-lang *language*		✓	✓	✓
Environment Variable	UCMDLANG=*language*	✓	✓	✓	
Configuration File Keyword	language *language*	✓	✓	✓	✓
STRUCM Parameter	MSGLANG(*language*)	✓			

Values

language is any UMC file provided by Stonebranch Inc.

IBM i	Each UMC file is for a specific language. The first three characters of the language name are used as a three-character suffix in the UMC member base name **UCMMC**. UMC files are located in the source physical file **UNVPRD510/UNVNLS**.
UNIX	Each UMC file is for a specific language. The first three characters of the language name are used as a three-character suffix in the UMC file base name. All UMC files have a **.UMC** extension.
Windows	Each UMC file is for a specific language. The first three characters of the language name are used as a three-character suffix in the UMC file base name. All UMC files have a **.UMC** extension.
z/OS	*language* translates to a member name of the library allocated on the UNVNLS ddname. The first three characters of the language name are used as a three-character suffix of the member name. Universal Command message catalog member names start with **UCMMC**.

Default is ENGLISH.

MESSAGE_LEVEL - UCMD Manager configuration option

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

It also specifies, optionally, whether or not to include a date and time stamp with each message.

Usage

Method	Syntax*****	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-l *level[,time]*		✓	✓	✓
Command Line, Long Form	-level *level[,time]*		✓	✓	✓
Environment Variable	UCMDLEVEL=*level[,time]*	✓	✓	✓	
Configuration File Keyword	message_level *level[,time]*	✓	✓	✓	✓
STRUCM Parameter	MSGLEVEL(**level*)	✓			
******time* is not valid for the IBM i STRUCM Parameter.					

Values

level indicates either of the following level of messages:

- **trace**
  Activates tracing and generates a trace file to which UCDM writes trace messages used for debugging (see Trace Files, below).

Note

⚠

Use **trace** only as directed by Stonebranch, Inc. Customer Support.

- **audit**
  Issues audit, informational, warning, and error messages.
- **info**
  Issues informational, warning, and error messages.
- **warn**
  Issues warning and error messages.
- **error**
  Issues error messages only.

Default

IBMi	Default is info.
UNIX	Default is warn.
Windows	Default is warn.
z/OS	Default is info.

time is the specification for whether or not to include a time stamp with each message.

Valid values for *time* are:

- **time**
  Include a time and date stamp on each message.
- **notime**
  Do not include a time and date stamp on each message.

Default

IBM i	Default is time.
UNIX	Default is notime.
Windows	Default is notime.
z/OS	Default is time.

Trace Files

IBM i	Trace file name is ***CURLIB/UNVTRCUCM(UCMxxxxxx)** where **xxxxxx** is the job number of the job invoking Universal Command. The default library for *TRACE* is the current library (**curlib**) of the Manager process.
UNIX	Trace file name is **ucmd.trc**. It is created in the working directory of UCMD Manager.
Windows	Trace file name is **ucmd.trc**. It is created in the working directory of UCMD Manager.
z/OS	Trace file is written to the data set referenced by the **UNVTRACE** ddname.

MFT_SAFE_MODE - UCMD Manager configuration option

Description

The MFT_SAFE_MODE controls the behavior of manager fault tolerance (MFT) by specifying whether or not multiple hosts can be specified in the REMOTE_HOST option.

This MFT safe mode protects against problems that might arise when automated processing is used for manager fault tolerant UCMD Managers that specify multiple entries in a REMOTE_HOST list.

By default, MFT_SAFE_MODE is enabled, allowing only one host to be specified in REMOTE_HOST. If MFT_SAFE_MODE is disabled, multiple hosts can be specified.

For example, assume that a UCMD Manager, cancelled because of a failed network connection, was configured with the following:

- REMOTE_HOST specifies multiple entries.
- COMMAND_ID is static.
- MANAGER_FAULT_TOLERANT = **yes**.
- RESTART = **no**.

If the network connection is back up when the job is resubmitted, the Broker on the original host will only accept a restart request from the UCMD Manager. In this situation, the UCMD Manager would fail (as it should), and the operator would know to set the RESTART option to **auto** or **yes**, and to provide the original host for REMOTE_HOST.

However, if the network connection is still down when the job is resubmitted, or the UCMD Manager begins its connection attempts with a different host, a new UCMD Server instance — very likely duplicating work already performed by the original UCMD Server — will be started on a different system.

MFT_SAFE_MODE is ignored when the MANAGER_FAULT_TOLERANT option is set to **no**.

Valid MFT_SAFE_MODE values, and the MANAGER_FAULT_TOLERANT and REMOTE_HOST values that may be specified for each, are identified in the following #Values section.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-mft_safe_mode *option*		✓	✓	✓
Environment Variable	UCMDMFTSAFEMODE=*option*	✓	✓	✓	
Configuration File Keyword	mft_safe_mode *option*	✓	✓	✓	✓
STRUCM Parameter	MFTSAFMODE(*option)	✓			

Values

option is the specification for whether or not Manager Fault Tolerant (MFT) Safe Mode is enabled:

- **yes**
  MFT Safe Mode is enabled.
- **no**
  MFT Safe Mode is disabled.

Default is yes.

Combinations of MANAGER_FAULT_TOLERANT and MFT_SAFE_MODE settings, and how each affects what may be specified for REMOTE_HOST, are listed below.

MANAGER_FAULT_TOLERANT	RESTART	MFT_SAFE_MODE	REMOTE_HOST
No	n/a	Ignored	Multiple entries allowed.
Yes	No	Yes	Only 1 entry allowed. UCMD Manager will fail if multiple entries reside in the host list after entry is resolved, expanded, and scrubbed.
Yes	No	No	Multiple entries allowed. Use with caution; introduces duplicate processing risks described above.

Yes	Yes	n/a	Only 1 entry allowed (must match original remote system). UCMD Manager will fail if multiple entries specified, regardless of MFT_SAFE_MODE value.
Yes	Auto	n/a	Only 1 entry allowed. UCMD Manager will fail if multiple entries specified, regardless of MFT_SAFE_MODE value.

MIN_SSL_PROTOCOL - UCMD Manager configuration option

Description

The MIN_SSL_PROTOCOL option specifies the minimum SSL/TLS protocol level that will be negotiated and used for communications channels.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-min_ssl_protocol *option*	✓	✓	✓	✓
Environment Variable	UCMDMINSSLPROTOCOL *option*	✓	✓	✓	
Configuration File Keyword	min_ssl_protocol *option*	✓	✓	✓	✓
STRUCM Parameter	MINSSLPROTOCOL *option*	✓			

Note

This option was introduced to IBM i in version 5.1.1.0.

Values

option is the specification for the minimum SSL/TLS protocol level that will be supported.

- **TLS1_0**
  Minimum SSL/TLS protocol is TLS 1.0.
- **TLS1_2**
  Minimum SSL/TLS protocol is TLS 1.2.

Default is TLS1_2.

MSG_SUPPRESSION_LIST - UCMD Manager configuration option

Description

The MSG_SUPPRESSION_LIST option specifies a list of message IDs representing Universal messages to be suppressed.

The list consists of zero or more comma-separated Universal message ID numbers. For example:

- 193 - Suppress message UNV0193W only.
- 192,193 - Suppress message UNV0192W and UNV0193W.

Suppressed messages are not printed to logs or output, even if a condition arises that normally would produce the message(s).

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-msg_suppression_list *list*		✓	✓	✓
Environment Variable	UCMDMSGSUPPRESSIONLIST=*list*		✓	✓	✓
Configuration File Keyword	msg_suppression_list *list*		✓	✓	✓
STRUCM Parameter	n/a				

Values

list is the list of message IDs representing Universal messages to be suppressed.

NETWORK_DELAY - UCMD Manager configuration option

Description

The NETWORK_DELAY option specifies the maximum acceptable delay in transmitting data over the network between the UCMD Manager and UCMD Server.

If a data transmission takes longer than this specified delay, the operation ends with a time-out error.

NETWORK_DELAY provides the ability to fine tune Universal Command's network protocol. When a data packet is sent over a TCP/IP network, the time it takes to reach the other end depends on many factors, such as network congestion and bandwidth. If the packet is lost before reaching the other end, the other end may wait indefinitely for the expected data. In order to prevent this situation, Universal Command times out waiting for a packet to arrive in the period of time specified by NETWORK_DELAY.

Universal Command considers a time-out error as a network fault. If the UCMD Manager and UCMD Server are running with the network fault tolerant protocol, they will reestablish network connections and continue.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-d *seconds*		✓	✓	✓
Command Line, Long Form	-delay *seconds*		✓	✓	✓
Environment Variable	UCMDNETWORKDELAY=seconds	✓	✓	✓	
Configuration File Keyword	network_delay *seconds*	✓	✓	✓	✓
STRUCM Parameter	DELAY(*seconds*)	✓			

Values

seconds is the number of seconds to delay before ending an operation with a time-out error.

Default is*120*.

NETWORK_FAULT_TOLERANT - UCMD Manager configuration option

Description

The NETWORK_FAULT_TOLERANT option specifies whether or not network fault tolerance should be activated.

Network fault tolerance enables a UCMD Manager and UCMD Server to recover from network faults and continue executing without any loss of data.

Network fault tolerance is not negotiated between the UCMD Manager and UCMD Server. The UCMD Server must have its NETWORK_FAULT_TOLERANT configuration option activated in order for the UCMD Manager and UCMD Server to execute with the network fault tolerant protocol.

If the UCMD Server does have network fault tolerance activated, this UCMD Manager NETWORK_FAULT_TOLERANT option controls whether or not it is used.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-N *option*		✓	✓	✓
Command Line, Long Form	-networkft *option*		✓	✓	✓
Environment Variable	UCMDNETWORKFT=*option*	✓	✓	✓	
Configuration File Keyword	network_fault_tolerant *option*	✓	✓	✓	✓
STRUCM Parameter	NETWORKFT(**option*)	✓			

Values

option is the specification for whether or not to activate network fault tolerance.

Valid values for *option* are:

- **yes**
  Fault tolerance is requested. If the UCMD Server has fault tolerance activated, the UCMD Manager and UCMD Server will use a fault tolerant network protocol.
- **no**
  Fault tolerance is not requested. The fault tolerant protocol will not be used.

Default is no.

NLS_DIRECTORY - UCMD Manager configuration option

Description

The NLS_DIRECTORY option specifies the name of the directory where the UCMD Manager message catalog and code page tables are located.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	n/a				
Environment Variable	n/a				
Configuration File Keyword	nls_directory *directory*		✓	✓	
STRUCM Parameter	n/a				

Values

directory is the name of the directory where the catalog and tables are located.

Full path names are recommended.

Relative path names are relative to the **universal** installation directory.

UNIX

Default is /opt/universal/nls.

Windows

Default is ..\nls.

OUTBOUND_IP - UCMD Manager configuration option

Description

The OUTBOUND_IP option sets the host or IP address that UCMD binds to when initiating outgoing connections.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-outboundip *host*		✓	✓	✓
Environment Variable	UCMDOUTBOUNDIP=*host*	✓	✓	✓	
Configuration File Keyword	outboundip *host*	✓	✓	✓	✓
STRUCM Parameter	OUTBOUNDIP(*host*)	✓			

Values

host is the host or IP address.

There is no default.

PLF_DIRECTORY - UCMD Manager configuration option

Description

The PLF_DIRECTORY option specifies the Program Lock File (PLF) directory where the program lock files are located.

A program lock file is created and used by the UCMD Manager process to store manager process termination information for the Universal Broker.

IBM i

Do not include this directory in any system or other backup that requires an exclusive lock on the directory while Universal Command is running.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-plf_directory *directory*		✓		
Environment Variable	UCMDPLFDIRECTORY=*directory*	✓	✓		
Configuration File Keyword	n/a				
STRUCM Parameter	PLFDIR(*IFS directory*)	✓			

Values

directory is the name of the PLF directory.

UNIX

Default is /var/opt/universal/tmp.

IBM i

Default is */tmp.

PRIVATE_KEY - UCMD Manager configuration option

Description

The PRIVATE_KEY option specifies the location of the PEM-formatted RSA private key that corresponds to the X.509 certificate specified by the CERTIFICATE option.

Note

PRIVATE_KEY is required only if a certificate is specified by CERTIFICATE.

z/OS

PRIVATE_KEY is used only when the SSL_IMPLEMENTATION option is set to *OPENSSL*.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-private_key *ddname* or *file*		✓	✓	✓
Environment Variable	UCMDPRIVATEKEY=*file*	✓	✓	✓	
Configuration File Keyword	private_key *ddname* or *file*	✓	✓	✓	✓
STRUCM Parameter	PVTKEYF(*file [lib]*) [PVTKEYFMBR (*member*)]	✓			

Values

z/OS	*ddname* is the ddname of the PEM formatted RSA private key that corresponds to the X.509 certificate. Allocated to the ddname must be either a sequential data set or a member of a PDS that has a variable record format.
UNIX	*file* is the path of the PEM-formatted RSA private key file that corresponds to the X.509 certificate.
Windows	*file* is the path of the PEM-formatted RSA private key file that corresponds to the X.509 certificate.
IBM i	*file* is the qualified name of the PEM-formatted RSA private key file that corresponds to the X.509 certificate. The file name can be qualified by a library name. If not, the library list ***LIBL** is searched for the first occurrence of the file name.

PRIVATE_KEY_PWD - UCMD Manager configuration option

Description

The PRIVATE_KEY_PWD option specifies the password or pass phrase for the PEM-formatted RSA private key specified with the PRIVATE_KEY option.

Note

Whether or not the password is required or not depends on whether or not it is required by the private key.

z/OS

PRIVATE_KEY_PWD is used only when the SSL_IMPLEMENTATION option is set to **openssl**.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-private_key_pwd *password*		✓	✓	✓
Environment Variable	UCMDPRIVATEKEYPWD=*password*	✓	✓	✓	
Configuration File Keyword	private_key_password=*password*	✓	✓	✓	✓
STRUCM Parameter	PVTKEYPWD(*password*)	✓			

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-private_key_pwd *password*		✓	✓	✓
Environment Variable	UCMDPRIVATEKEYPWD=*password*	✓	✓	✓	
Configuration File Keyword	private_key_password=*password*	✓	✓	✓	✓
STRUCM Parameter	PVTKEYPWD(*password*)	✓			

Values

password is the password for the private key.

IBM i

Characters may be incorrectly translated due to reverse representations under 037 and 1047 CCSIDs:

- hat (circumflex): logical not
- left bracket: Y acute
- right bracket: diaeresis (umlaut)

The hex/decimal exchanges are:

- 5F/95: B0/176
- AD/173: BA/186
- BD/189: BB/187

RECONNECT_RETRY_COUNT - UCMD Manager configuration option

Description

The RECONNECT_RETRY_COUNT option specifies the number of reconnect attempts that are performed before ending with an error.

RECONNECT_RETRY_COUNT is used when the UCMD Manager and UCMD Server are communicating with the fault tolerant protocol. If a network fault is detected, the UCMD Manager attempts to reestablish network connections with the UCMD Server. If a connection attempt fails due to a network fault, the UCMD Manager will wait for a specified period of time and then attempt the connection again.

This continues until the UCMD Manager and UCMD Server successfully reestablish network connections or until the number of attempts exceeds the number specified in RECONNECT_RETRY_COUNT.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-retry_count *number*		✓	✓	✓
Environment Variable	UCMDRETRYCOUNT=*number*	✓	✓	✓	
Configuration File Keyword	reconnect_retry_count *number*	✓	✓	✓	✓
STRUCM Parameter	RETRYCNT(*number*)	✓			

Values

number is the number of reconnect attempts to be performed before ending with an error.

A value of *0* indicates that no reconnect attempts are performed.

Default is 20.

RECONNECT_RETRY_INTERVAL - UCMD Manager configuration option

Description

The RECONNECT_RETRY_INTERVAL option specifies the number of seconds between each reconnect attempt.

RECONNECT_RETRY_INTERVAL is used when the UCMD Manager and UCMD Server are communicating with the fault tolerant protocol. If a network fault is detected, the UCMD Manager attempts to reestablish network connections with the UCMD Server. If a connection attempt fails due to a network fault, the UCMD Manager will wait the number seconds specified by RECONNECT_RETRY_INTERVAL and then attempt the connection again.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-retry_interval *seconds*		✓	✓	✓
Environment Variable	UCMDRETRYINTERVAL=*seconds*	✓	✓	✓	
Configuration File Keyword	reconnect_retry_interval *seconds*	✓	✓	✓	✓
STRUCM Parameter	RETRYINT(*seconds*)	✓			

Values

seconds is the number of seconds to wait between reconnect attempts.

Default is 60.

REMOTE_HOST - UCMD Manager configuration option

Description

The REMOTE_HOST option specifies a comma-delimited list of one or more hosts upon which a command can run.

Each host in the list can be specified as a host name or an IP address in dotted decimal notation.

Note

⚠

For purposes of this discussion, even if one host is specified, it is considered a list with a single entry.

If multiple hosts are specified, the UCMD Manager will try each one until it successfully connects to a Universal Broker. When a successful connection is established, no more hosts in the list are tried. If the UCMD Manager is unable to establish a connection to a Universal Broker on any of the hosts in the list, the UCMD Manager will fail.

Each host in the REMOTE_HOST list also can include a port number or service name that the UCMD Manager will use to establish a connection with the Universal Broker on that host. This is useful in situations where the remote Broker is configured to accept incoming connections on a port that is different from the UCMD Manager's configured value.

UCMD Manager automatically removes any duplicates found in the list of hosts before it makes its first connection attempt. This includes any host names that resolve to the same IP address, or duplicate IP addresses that are added to the list following DNS expansion (see the DNS_EXPAND option).

Note

⚠

Duplicate IP addresses may appear in the resolved, expanded, scrubbed list of hosts if a different port number is specified for each occurrence of a particular IP address (for example, 192.168.1.1:7887 and 192.168.1.1:7888 are considered distinct entries).

To set configuration options that control selection of the hosts in the REMOTE_HOST list, see the HOST_SELECTION and CONNECT_TIMEOUT options.

To set configuration options that control processing of the hosts in the REMOTE_HOST list, see the DNS_EXPAND and MFT_SAFE_MODE options.

The following text briefly describes each of these options.

HOST_SELECTION

UCMD Manager uses the HOST_SELECTION option to control which host in the REMOTE_HOST list is selected first. UCMD Manager can be configured to always select the first host or to select a randomly chosen host as it begins its attempts to connect to a remote Universal Broker. In either case, after a host is selected, UCMD Manager processes the list sequentially until either a connection succeeds or all hosts in the list have been tried.

CONNECT_TIMEOUT

UCMD Manager uses the CONNECT_TIMEOUT option to specify how long it will wait for a connection attempt to succeed before it moves on to the next host in the list.

DNS_EXPAND

To set configuration options that control processing of the hosts in the REMOTE_HOST list, the DNS_EXPAND and MFT_SAFE_MODE options are provided.

The DNS_EXPAND option controls the number of IP addresses returned when UCMD Manager issues a DNS query to resolve a host name. If the Manager is configured to expand the results of the query, all IP addresses defined for a particular host name are returned and expanded (in-place) within the list of hosts. Otherwise, only the first host is returned, and no expansion is performed.

MFT_SAFE_MODE

The MFT_SAFE_MODE option controls the situations in which more than one host may be specified in the REMOTE_HOST list when manager fault tolerance (MFT) is enabled. Automated restarts may result in an MFT restart being tried on a system that is different from the original system. If a restartable MFT process is not found on that system, new work may be started. The MFT_SAFE_MODE option guards against duplicate processing in this situation.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-i *hostlist*		✓	✓	✓
Command Line, Long Form	-host *hostlist*		✓	✓	✓
Environment Variable	UCMDHOST *hostlist*	✓	✓	✓	

Configuration File Keyword	host *hostlist*	✓	✓	✓	✓
STRUCM Parameter	HOST(*hostlist*)	✓			

Values

hostlist is a list of one or more hosts, in the following format:

host1[[:port1],host2[:port2],host3[:port3],...hostn[:portn]]

In this format:

- *host* is the IP address (in dotted decimal notation) or host name of the system upon which the command may run.
- *port* is an optional port number (or service name), which is necessary only if the remote Universal Broker is accepting incoming connections on a port that is different from the value specified by the REMOTE_PORT option.

Note

The *port* number, when specified, must be separated from the *host* by a colon (`:`).

HP NonStop

Only a single host can be specified, either as an IP address in dotted decimal notation or a host name.

In addition, a port number only can be specified with the REMOTE_PORT option. Specifying a port number in the host list is not supported.

REMOTE_PORT - UCMD Manager configuration option

Description

The REMOTE_PORT option specifies the TCP port on the remote computer on which to send the command.

The remote computer must have a Universal Broker running and accepting connections on the specified port.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-p *port*		✓	✓	✓
Command Line, Long Form	-port *port*		✓	✓	✓
Environment Variable	UCMDPORT *port*	✓	✓	✓	
Configuration File Keyword	port *port*	✓	✓	✓	✓
STRUCM Parameter	PORT(*port*)	✓			

Values

port is the TCP port IP on the remote computer.

The format of *port* can be either:

- Number (for example, 7887)
- Service name (for example, ubroker)

Default is 7887.

RESTART - UCMD Manager configuration option

Description

The RESTART option specifies whether or not this execution of the UCMD Manager is a restart of a previous manager fault tolerant UCMD Manager.

(See Manager Fault Tolerance - Universal Command for details on the manager fault tolerant feature.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-R *option*		✓	✓	✓
Command Line, Long Form	-restart *option*		✓	✓	✓
Environment Variable	UCMDRESTART *option*	✓	✓	✓	
Configuration File Keyword	restart *option*	✓	✓	✓	✓
STRUCM Parameter	RESTART(**option*)	✓			

Values

option is the specification for whether or not this execution of UCMD Manager is a restart.

Valid values for *option* are:

- **yes**
  UCMD Manager is restarting an existing unit of work represented by a command ID.

  The COMMAND_ID and MANAGER_FAULT_TOLERANT options are required.
- **no**
  UCMD Manager is not restarting.
- **auto**
  UCMD Manager checks to see if the Universal Broker is executing an existing unit of work, identified by COMMAND_ID. If a matching command ID is found, the UCMD Manager attempts a restart. If a matching command ID is not found, the UCMD Manager does not attempt a restart.

  The COMMAND_ID and MANAGER_FAULT_TOLERANT options are required. However, **auto** cannot be used if the COMMAND_ID value is *****, which specifies that the UCMD Manager will generate a unique command ID for each run.

Default is no.

SAF_KEY_RING - UCMD Manager configuration option

Description

The SAF_KEY_RING option specifies the SAF (RACF is a SAF implementation) certificate key ring name that the Universal Command job should used for its certificate.

The key ring must be associated with the user profile with which the Universal Command job executes.

Note

SAF_KEY_RING is required if the SSL_IMPLEMENTATION option is set to **system**.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-saf_key_ring *name*				✓
Environment Variable	n/a				
Configuration File Keyword	saf_key_ring *name*				✓
STRUCM Parameter	n/a				

Values

name is the name of the SAF certificate key ring.

SAF_KEY_RING_LABEL - UCMD Manager configuration option

Description

The SAF_KEY_RING_LABEL option specifies the label of the certificate in the SAF (RACF is a SAF implementation) certificate key ring that the Universal Command job should use for its certificate.

(The key ring is specified by the SAF_KEY_RING option.)

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-saf_key_ring_label *label*				✓
Environment Variable	n/a				
Configuration File Keyword	saf_key_ring_label *label*				✓
STRUCM Parameter	n/a				

Values

label is the label of the SAF certificate key ring.

Default is default certificate in the key ring.

SCRIPT_FILE - UCMD Manager configuration option

Description

The SCRIPT_FILE option specifies a script file to execute on the remote computer.

The script file can be any script that is valid for the command shell on the remote computer's operating system. The maximum size of the script file is approximately 64,000 characters.

The entire script is read and sent to the remote system for execution.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-s *ddname* or *file*		✓	✓	✓
Command Line, Long Form	-script *ddname* or *file*		✓	✓	✓
Environment Variable	UCMDSCRIPT *file*	✓	✓	✓	
Configuration File Keyword	n/a				
STRUCM Parameter	SCRFILE(*file*) [SCRMBR(*member*)]	✓			

Values

IBM i	Script is read from *file* name. The file name can be qualified with a library; otherwise, the library list ***LIBL** is searched for the first occurrence of the file. A member name can be used for further qualification by specifying the **SCRMBR** parameter.
UNIX	Script is read from *file* name.
Windows	Script is read from *file* name.
z/OS	Script is read from the DD statement *ddname*.

SCRIPT_LINE_IGNORE_MASK - UCMD Manager configuration option

Description

The SCRIPT_LINE_IGNORE_MASK option, which is used in conjunction with the SCRIPT_FILE option, specifies a regular expression that is matched against each line of the script being sent to the remote system.

Any line that matches the mask will be ignored and not sent to the remote system as part of the script.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-script_line_ignore_mask *mask*				✓
Environment Variable	n/a				
Configuration File Keyword	script_line_ignore_mask *mask*				✓
STRUCM Parameter	n/a				

Values

mask is a regular expression used to filter lines from a script specified with the -s option. Script lines that match the regular expression will not be transferred to the server for processing.

Example

An example use of the SCRIPT_LINE_IGNORE_MASK option would be to ignore script lines containing ITWS variables.

The following configuration line would cause any script run by Universal Command Manager to ignore script lines containing the text //*%OPC:
script_line_ignore_mask ".*//*%OPC"

If the intention were to ignore lines only where the ITWS variable started the line, the following regular expression could be used that "anchors" the expression to the start of the line:
script_line_ignore_mask "^//*%OPC"

Note

The '^' character is hex value 0x5F. Depending on the terminal code page settings, a different keyboard character may be required to get the 0x5F value into the regular expression string. It may be necessary to turn HEX ON in ISPF and manually adjust the hex value.

If the mask contains spaces or characters that are used by the command line interpreter, (for example, '-', '/'), the mask value must be enclosed in parenthesis.

SCRIPT_OPTIONS - UCMD Manager configuration option

Description

The SCRIPT_OPTIONS option specifies command line options to pass to the script file.

Note

⚠ SCRIPT_OPTIONS is valid only if the SCRIPT_FILE option specifies a script file.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-o *options*		✓	✓	✓
Command Line, Long Form	-options *options*		✓	✓	✓
Environment Variable	UCMDOPTIONS=*options*	✓	✓	✓	
Configuration File Keyword	n/a				
STRUCM Parameter	OPTIONS(*options*)	✓			

Values

options is the command line options to pass to the script file.

The remote command shell processes meta-characters (variable evaluation **$** or **%** and file name expansions such as *****) as normal on the remote computer.

IBM i	If *options* contains spaces or shell meta-characters, it must be enclosed in single (**'**) or double (**"**) quotation marks.
UNIX	If *options* contains spaces or shell meta-characters, it must be enclosed in single (**'**) or double (**"**) quotation marks. If an enclosing character is part of the option, prefix the character with the command line escape character, back slash (\).
Windows	If *options* contain spaces, it must be enclosed in double (**"**) quotation marks. If a quotation mark is part of the value, prefix it with the Windows escape character, back slash (\\).
z/OS	If *options* contains spaces, it must be enclosed in single (**'**) or double (**"**) quotation marks. If an enclosing character is part of options, use two consecutive characters to produce one.

SCRIPT_TYPE - UCMD Manager configuration option

Description

The SCRIPT_TYPE option specifies the type of script specified by the SCRIPT_FILE option.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-script_type *type*		✓	✓	✓
Environment Variable	UCMDSCRIPTTYPE=*type*	✓	✓	✓	
Configuration File Keyword	script_type *type*	✓	✓	✓	✓
STRUCM Parameter	SCRTYPE(*type*)	✓			

Values

type is the type of script.

(*type* is not case-sensitive.)

Note

⚠ Currently, only UCMD Servers for HP NonStop, UNIX, and Windows allow values other than their default values (see SCRIPT_TYPE).

Except for the service script type (which supports Universal Command Agent for SOA), UCMD Server for Windows interprets the value as a file extension. The extension must have a registered application associated with it; otherwise, UCMD Server will not accept it.

SERVER_OPTIONS - UCMD Manager configuration option

Description

The SERVER_OPTIONS option specifies options to override UCMD Server options.

Note

Not all UCMD Server options can be overridden. See Universal Command Server Configuration Options for information on which options can be overridden.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-S *options*		✓	✓	✓
Command Line, Long Form	-server *options*		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCM Parameter	SERVER(*options*)	✓			

Values

options is the options to override the UCMD Server options.

The options must be specified in the UCMD Server's command line option format.

IBM i	The options must be enclosed in single (**'**) quotation marks. For example: `-S '  -joblog error'`
UNIX	The options must be enclosed in single (**'**) or double (**"**) quotation marks, with at least one space between the first enclosing character and the first option name. The space is required to avoid a command line specification error due to how command options are parsed. For example: `-S '  -joblog error'`
Windows	The options must be enclosed in double (**"**) quotation marks, with at least one space between the first enclosing character and the first option name. The space is required to avoid a command line specification error due to how command options are parsed. For example: `-S "  -joblog error"`

z/OS	The options must be enclosed in single (`'`) or double (`"`) quotation marks, with at least one space between the first enclosing character and the first option name. The space is required to avoid a command line specification error due to how command options are parsed. For example: `-S '  -joblog error'`

SERVER_STOP_CONDITIONS - UCMD Manager configuration option

Description

The SERVER_STOP_CONDITIONS option specifies one or more exit codes of the executing UCMD Manager that should trigger the locally running Universal Broker to cancel the corresponding UCMD Server for the exited UCMD Manager.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-server_stop_conditions *codes*				✓
Environment Variable	UCMDSERVERSTOPCONDITIONS=*codes*				✓
Configuration File Keyword	server_stop_conditions *codes*				✓
STRUCM Parameter	n/a				

Values

codes is an exit code, or a comma-separated list of exit codes, that should cause the UCMD Server to be cancelled.

z/OS ABEND codes are specified in two different formats:

- System ABEND code
  Starts with S followed by a 3-character hexadecimal value.
- User ABEND code
  Starts with U followed by a 4-character decimal value.

For example, when a job is terminated with the CANCEL console command, the job ends with a system ABEND code of S222.

There is no default.

SIO_DATA_AUTHENTICATION - UCMD Manager configuration option

Description

The SIO_DATA_AUTHENTICATION option specifies whether or not the standard file data sent over the network should be authenticated when using the **UNVv2** protocol.

Generating a checksum value for each data block performs authentication. The checksum value is sent with the data block. The receiver generates a second checksum value for the data block. If the two checksum values are not equal, the authentication fails. Failed authentication closes the network connection.

The checksum is generated with the MD5 Message Digest Algorithm by RSA Data Security, Inc.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-a *option*		✓	✓	✓
Command Line, Long Form	-authenticate *option*		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCM Parameter	n/a				

Values

option is the specification for whether or not the data is authenticated.

Valid values for option are:

- **yes**
  Data authentication is required. The standard file data transfers are authenticated regardless of UCMD Server's authentication option.
- **no**
  Data authentication is not required for the standard file. It is still possible for UCMD Server to request data authentication.

Default is no.

SIO_DATA_COMPRESSION - UCMD Manager configuration option

Description

The SIO_DATA_COMPRESSION option specifies whether or not the standard file data transmitted across the network should be compressed.

It also can specify, optionally, the compression method to use.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-k *option[UCMD:,method]*		✓	✓	✓
Command Line, Long Form	-compress *option[UCMD:,method]*		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCM Parameter	n/a				

Values

option is the specification for whether or not data should be compressed.

Valid values for *option* are:

- **yes**
  Data compression is required. All network data transfers for the standard file are compressed regardless of UCMD Server's DATA_COMPRESSIOI option.
- **no**
  Data compression is not required. It still is possible for UCMD Server to request data compression {via its DATA_COMPRESSION option).

Default is no.

method is the compression method to be used

Valid values for *method* are:

- **zlib**
  Data will be compressed using ZLIB compression algorithm. This method usually results in a very high compression rate, but tends to be somewhat CPU-intensive. ZLIB is recommended in environments where controlling a process's CPU usage is not necessarily a priority.
- **hasp**
  Data will be compressed using the HASP compression algorithm. This method is less CPU-intensive than the ZLIB method, and is recommended in environments where controlling CPU usage is a priority. With HASP, the compression rate - while still very good - tends to be a little less than what is possible with the ZLIB method.

Default is zlib.

SIO_DATA_ENCRYPTION - UCMD Manager configuration option

Description

The SIO_DATA_ENCRYPTION option specifies whether or not the standard file data sent over the network should be encrypted.

Encryption protects the privacy of the data. **UNVv2** data encryption uses one of several encryption algorithms, such as the Data Encryption Standard (DES) algorithm.

SSL/TLS data encryption uses one of the SSL/TLS cipher suites specified with the DATA_SSL_CIPHER_LIST option.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-e *option*		✓	✓	✓
Command Line, Long Form	-encrypt *option*		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCM Parameter	n/a				

Values

option is the specification for whether or not is encrypted.

Valid values for *option* are:

- **yes**
  Data encryption is required. The standard file data transfers are encrypted regardless of the UCMD Server DATA_COMPRESSION option.
- **no**
  Data encryption is not required. It is still possible for UCMD Server to request data encryption (via its DATA_COMPRESSION option).

Default is no.

SIO_LOCAL_CODE_PAGE - UCMD Manager configuration option

Description

The SIO_LOCAL_CODE_PAGE option specifies the character code page used to translate text data received and transmitted over the network on the local system.

The local system is the system on which the UCMD Manager executes.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-t *codepage*		✓	✓	✓
Command Line, Long Form	-codepage *codepage*		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCM stdin Parameter STRUCM stdout Parameter STRUCM stderr Parameter	SINCPG *codepage* SOTCPG *codepage* SERCPG *codepage*	✓			

Values

codepage is the character code page to be used.

codepage references a Universal Translate Table (UTT) file provided with the product (see UTT Files). The code page value UTF-8, however, does not have a corresponding UTT file. UTF-8 support does not require a UTT file.

Note

UTF-8 support was added to UCMD Manager and UCMD Server at Version 3.2.0, Level 4.

See Character Code Pages for a complete list of character code pages provided by Stonebranch Inc. for use with Universal Agent.

Default

The default is different for different operating systems:

- ISO8859-1 (8-bit ASCII): ASCII-based operating systems
- IBM1047 (EBCDIC): EBCDIC-based operating system

SIO_LOCAL_FILE - UCMD Manager configuration option

Description

The SIO_LOCAL_FILE option specifies the location, instead of the default location, to which standard file data should be written or from where it should be read.

IBM i	SIO_LOCAL_FILE specifies the local file to be used for the standard files of the STRUCM command. The STRUCM standard files are redirected to and from the standard files of the remote command.
UNIX	SIO_LOCAL_FILE specifies that the standard file data should be written to or read from the specified file name instead of the standard output, standard error, and standard input of UCMD Manager. If no file name is provided, and the option is applied to standard input, it is treated as no standard input and a standard input end-of-file indicator is sent to the server immediately.
Windows	SIO_LOCAL_FILE specifies that the standard file data should be written to or read from the specified local file instead of the default file.
z/OS	SIO_LOCAL_FILE specifies that the standard file data should be written to or read from the specified ddname instead of the default ddnames.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-F *ddname* or *file*		✓	✓	✓
Command Line, Long Form	-localfile *ddname* or *file*		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCM stdin Parameter STRUCM stdout Parameter STRUCM stderr Parameter	SINFILE(*input_file*) [SINMBR(*member*)] SOTFILE(*output_file*) [SOTMBR(*member*)] SERFILE(*output_file*) [SERMBR(*member*)]	✓			

Values

IBM i	*input_file* is the file from where the data is read. Valid values for *input_file* are: • **NONE* No standard input file is allocated to *stdin*. If the remote command attempts to read from *stdin*, it will receive an end\-of\-file indicator. • **TERM* Standard input is allocated to the workstation for interactive jobs and to file *QINLINE* for batch jobs. If executed as an interactive job, an *ILE* session terminal is displayed from which input can be entered at the terminal. • *file_name* A file name to read as standard input. The file name can be qualified by a library name. If not, the library list *\*LIBL* is searched for the first occurrence of the file name. A member name can be used for further qualification by specifying the *SINMBR* parameter. *output_file* is the file to which the data is read. Valid values for *output_file* are: • **TERM* Standard output or error is allocated to the workstation for interactive jobs and to file **QPRINT** for batch jobs. If executed as an interactive job, an **ILE** session terminal is displayed from which output is viewed at the terminal. • *file_name* Standard output or error is written to a file. The file name can be qualified by a library name. If not, the library list ***LIBL** is searched for the first occurrence of the file name. If the file is not found, it is created as a physical source file with a record length of 266. If ***LIBL** is specified or implied, the file is created in **QGPL**. A member name can be used for further qualification by specifying the **SOTMBR** or **SERMBR** parameter.
UNIX	*file* is the file, instead of the default file, to which the data is written or from where it is read.
Windows	*file* is the file, instead of the default file, to which the data is written or from where it is read.

z/OS	*ddname* is the ddname, instead of the default ddname, to which the data is written or from where it is read.

Default

z/OS

Default values are:

- UNVOUT for standard out
- UNVERR for standard error
- UNVIN for standard input

SIO_MODE - UCMD Manager configuration option

Description

The SIO_MODE option specifies whether transferred data is treated as text or binary.

It also can specify, optionally, the translation method for a text data transfer. The translation method specifies how the text translation is performed.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-m *mode[,method]*		✓	✓	✓
Command Line, Long Form	-mode *mode[,method]*		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				

STRUCM stdin Parameter
STRUCM stdout Parameter
STRUCM stderr Parameter
	SINMODE(**mode*)
	SOTMODE(**mode*)
	SERMODE(**mode*)
		✓			

Values

mode is the specification for how transferred data is treated.

Valid values for *mode* are:

- **text**
  Treat the data as text. The data is translated to and from local character code pages.
  An example of text data is reports. Reports contain character data that must be translated from one code page to another if they are to be read on the receiving system.
- **binary**
  Treat the data as binary. The data is not translated in any way.
  An example of binary data is backup files. Backup files are not viewed on the receiving system. They contain non-printable characters that are used for its internal representation.

Default is text.

IBM i

An asterisk (*****) must be entered with *mode*, as shown above, in the selected parameter. If an asterisk is not entered, an error message will show. Do not enter an asterisk in the command line.

method is the method to use for translation *TEXT* data.

(Specify the *method* after the *mode*, separated by a comma and no spaces.)

Valid values for *method* are:

- **ucs**
  Text translation is performed by converting local codepages to the Universal Character Set (Unicode) format and back again.
  This method requires:
  - Less time and network resources to establish network connections between the UCMD Manager and remote server.
  - More time and CPU resources to perform the text translation.
- **direct**
  Text translation is performed by converting directly from the local codepage to the remote codepage.
  This method requires:
  - More time and network resources to establish network connections between the UCMD Manager and remote server.
  - Less time and CPU resources to perform the text translation.

Note

For transferring large amounts of text data, the **direct** method is recommended.

Default is direct.

SIO_REMOTE_CODE_PAGE - UCMD Manager configuration option

Description

The SIO_REMOTE_CODE_PAGE option specifies the character code page that is used to translate text data received and transmitted over the network on the remote system.

The remote system is the system on which the UCMD Server executes.

IBM i

If this option is not used, no code page is specified; the UCMD Server will use its configured code page.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-T *codepage*		✓	✓	✓
Command Line, Long Form	-remotecodepage *codepage*		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				

STRUCM stdin Parameter
STRUCM stdout Parameter
STRUCM stderr Parameter
	SINRCP(*codepage*)
	SOTRCP(*codepage*)
	SERRCP(*codepage*)
		✓			

Values

codepage is the character code page to be used.

codepage references a Universal Translate Table (UTT) file provided with the product (see UTT Files). The code page value UTF-8, however, does not have a corresponding UTT file. UTF-8 support does not require a UTT file.

Note

UTF-8 support was added to UCMD Manager and UCMD Server at Version 3.2.0, Level 4.

See Character Code Pages for a complete list of character code pages provided by Stonebranch Inc. for use with Universal Agent.

SIO_TRAILING_SPACES - UCMD Manager configuration option

Description

The SIO_TRAILING_SPACES option specifies how fixed format records are read from the UNVIN ddname.

Note

SIO_TRAILING_SPACES is only for standard input, text mode option for z/OS.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-trailingspaces *option*				✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCM Parameter	n/a				

Values

option is the specification for how records are read.

Valid values for *option* are:

- **yes**
  Complete record is read (including trailing spaces).
- **no**
  Record is truncated after the last non-space character.

Default is no.

SSL_IMPLEMENTATION - UCMD Manager configuration option

Description

The SSL_IMPLEMENTATION option specifies the SSL/TLS implementation to be used for network communications.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-ssl_implementation *option*				✓
Environment Variable	UCMDSSLIMPLEMENTATION=*option*				✓
Configuration File Keyword	ssl_implementation *option*				✓
STRUCM Parameter	n/a				

Values

option is the SSL/TLS implementation to be used.

Valid values for option are:

- **openssl**
  OpenSSL SSL library is used for the SSL/TLS protocol.
- **system**
  z/OS System SSL library is used for the SSL/TLS protocol. The z/OS System SSL library has installation and configuration prerequisites. (See the Universal Agent 7.0.x Installation, Upgrade, and Applying Maintenance for a description of the prerequisites before using System SSL.)

Default is openssl.

STDERR_FILE_SPEC - UCMD Manager configuration option

Description

The STDERR_FILE_SPEC option specifies that the Standard File options (those options beginning with the characters SIO) following this option apply to the **stderr** file.

The first option following a STDERR_FILE_SPEC option that is not a Standard File option terminates the list of Standard File option specifications for the **stderr** file.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-E		✓	✓	✓
Command Line, Long Form	-stderr		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCM Parameter	n/a				

Values

(There are no values to be specified for this option.)

STDIN_FILE_SPEC - UCMD Manager configuration option

Description

The STDIN_FILE_SPEC option specifies that the Standard File options (those options beginning with the characters SIO) following this option apply to the **stdin** file.

The first option following a STDIN_FILE_SPEC option that is not a Standard File option terminates the list of Standard File option specifications for the **stdir** file.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-I		✓	✓	✓
Command Line, Long Form	-stdin		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCM Parameter	n/a				

Values

(There are no values to be specified for this option.)

STDOUT_FILE_SPEC - UCMD Manager configuration option

Description

The STDOUT_FILE_SPEC option specifies that the Standard File options (those options beginning with the characters SIO) following this option apply to the **stdout** file.

The first option following a STDOUT_FILE_SPEC option that is not a Standard File option terminates the list of Standard File option specifications for the **stdout** file.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-O		✓	✓	✓
Command Line, Long Form	-stdout		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCM Parameter	n/a				

Values

(There are no values to be specified for this option.)

SYSTEM_ID - UCMD Manager configuration option

Description

The SYSTEM_ID option identifies the local Universal Broker with which the UCMD Manager must register before the Manager performs any request.

Each Universal Broker running on a system is configured with a system identifier that uniquely identifies the Broker.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-system_id *id*				✓
Environment Variable	UCMDSYSTEMID=*id*				✓
Configuration File Keyword	n/a				
STRUCM Parameter	n/a				

Values

id is the system identifier of the local Universal Broker.

Refer to the local Universal Broker administrator for the appropriate system ID to use.

TRACE_FILE_LINES - UCMD Manager configuration option

Description

The TRACE_FILE_LINES option specifies the maximum number of lines to write to the trace file.

(The average size of a trace file line is 50 characters.)

A trace file is generated when the MESSAGE_LEVEL option is set to a value of **trace**. The trace file will wrap around when the maximum number of lines has been reached and start writing trace entries after the trace header lines.

z/OS

The trace file is written to the **UNVTRACE** ddname.

TRACE_FILE_LINES has no effect if the **UNVTRACE** ddname has allocated a JES SYSOUT file.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-tracefilelines *lines*		✓	✓	✓
Environment Variable	UCMDTRACEFILELINES=*lines*	✓	✓	✓	
Configuration File Keyword	trace_file_lines *lines*	✓	✓	✓	✓
STRUCM Parameter	TRCLINES(*lines*)	✓			

Values

lines is the maximum number of lines to write to the trace file.

Default is 500,000,000.

TRACE_TABLE - UCMD Manager configuration option

Description

The TRACE_TABLE option specifies the size of a wrap-around trace table maintained in memory.

The trace table is written to a file / data set when the program ends under the conditions specified in this option. Tracing is activated, and a trace file is generated, when the MESSAGE_LEVEL option is set to **trace**.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-trace_table *size, cond*		✓	✓	✓
Environment Variable	UCMDTRACETABLE=*size,condition*	✓	✓	✓	
Configuration File Keyword	trace_table *size, cond*	✓	✓	✓	✓
STRUCM Parameter	TRCTBL(*size,cond*)	✓			

Values

size is the size (in bytes) of the table.

The size can be suffixed with either of the following characters:

- M: Indicates that the size is specified in megabytes
- K: Indicates that the size is specified in kilobytes

For example, 50M indicates that 50 X 1,048,576 bytes of memory is allocated for the trace table.

Default is 0 (trace table is not used).

condition is the condition under which the trace table is written.

Valid values for *condition* are:

- **error**
  Write the trace table if the program ends with a non-zero exit code.
- **always**
  Write the trace table when the program ends regardless of the exit code.
- **never**
  Never write the trace table.

Default is never.

UENCRYPTED_CODEPAGE - UCMD Manager configuration option

Description

The UENCRYPTED_CODEPAGE option specifies the character code page that Universal Command Manager uses to translate characters within a command file that has been encrypted with the Universal Encrypt utility.

The UENCRYPTED_CODEPAGE_PATH option specifies the path to the directory containing this code page.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-uencrypted_codepage *codepage*		✓	✓	✓
Environment Variable	UCMDUENCRYPTEDCODEPAGE=*codepage*		✓	✓	
Configuration File Keyword	n/a				
STRUCM Parameter	ECMFILECP(*codepage*)	✓			

Values

codepage is the character code page that will be used to translate the contents of an encrypted command file specified by the COMMAND_FILE_ENCRYPTED option.

codepage references a Universal Translate Table (UTT) file provided with the product. UTT files are used to translate between Unicode and the local single-byte code page. (All UTT files end with an extension of **.utt**.)

Note

UTF-8 is not a supported *codepage* value for UENCRYPTED_CODEPAGE. UTF-8 codepage is valid only for standard I/O text file translation. Consequently, it can be specified only with the SIO_LOCAL_CODE_PAGE and SIO_REMOTE_CODE_PAGE options.

See Character Code Pages for a complete list of character code pages provided by Stonebranch Inc. for use with Universal Agent.

Default

Default is different for different operating systems:

- ISO8859-1 (8-bit ASCII): ASCII-based operating systems
- IBM1047 (EBCDIC): EBCDIC-based operating system

UENCRYPTED_CODEPAGE_PATH - UCMD Manager configuration option

Description

The UENCRYPTED_CODEPAGE_PATH option identifies the location of the code page specified by the UENCRYPTED_CODEPAGE option. UENCRYPTED_CODEPAGE controls translation of command files encrypted with the Universal Encrypt utility.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-uencrypted_codepage_path *path*		✓	✓	
Environment Variable	UCMDUENCRYPTEDCODEPAGEPATH=*path*		✓	✓	
Configuration File Keyword	n/a				
STRUCM Parameter	n/a				

Values

path is the location of the code page specified by the UENCRYPTED_CODEPAGE option.

UNIX

Default is /opt/universal/nls.

Windows

Default is ..\nls.

USER_ID - UCMD Manager configuration option

Description

The USER_ID option specifies the user identifier that is used to sign on to the remote computer.

The remote UCMD Server determines whether this option is required.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-u *user*		✓	✓	✓
Command Line, Long Form	-userid *user*		✓	✓	✓
Environment Variable	UCMDUSERID=*user*	✓	✓	✓	
Configuration File Keyword	userid *user*	✓	✓	✓	✓
STRUCM Parameter	USERID(*user*)	✓			

Values

user is the user identifier that is used to sign on to the remote computer.

Note

user must be a valid user identifier on the remote computer.

IBM i

If STRUCM is executing as an interactive job, the workstation user is prompted for a user ID.

If STRUCM is executing in batch, it fails.

USER_PASSWORD - UCMD Manager configuration option

Description

The USER_PASSWORD option specifies the password for the user identifier that is specified in the USER_ID option.

The password is always encrypted when sent over the network, regardless of how encryption is configured on the UCMD Manager and UCMD Server.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-w *password*		✓	✓	✓
Command Line, Long Form	-pwd *password*		✓	✓	✓
Environment Variable	UCMDPWD=*password*	✓	✓	✓	
Configuration File Keyword	password *password*	✓	✓	✓	✓
STRUCM Parameter	PWD(*password*)	✓			

Values

pwd is the password for the user identifier.

It must be a valid password for the user identifier on the remote computer.

IBM i

If STRUCM is executing as an interactive job, the workstation user is prompted for a password. If STRUCM is executing in batch, it fails.

When the password-level system value (**QPWDLVL**) is set to two (2) or three (3), a passphrase consisting of non-alphabetic characters and mixed-case characters is enabled. Universal Command only allows upper case characters for passwords and passphrases.

Also, characters may be incorrectly translated due to reverse representations under 037 and 1047 CCSIDs:

- hat (circumflex): logical not
- left bracket: Y acute
- right bracket: diaeresis (umlaut)

The hex/decimal exchanges are:

- 5F/95: B0/176
- AD/173: BA/186
- BD/189: BB/187

VERIFY_HOST_NAME - UCMD Manager configuration option

Description

The VERIFY_HOST_NAME option specifies whether or not the Universal Broker's X.509 certificate identity is verified.

Verification consists of verifying that the certificate is issued by a trusted CA. The CA_CERTIFICATES option specifies which CA certificates are considered trusted.

The identity is verified by matching the value specified by VERIFY_HOST_NAME to the Universal Broker's certificate host value.

The following certificate fields are matched in the order listed:

1. X.509 **v3 dNSName** field of the **subjectAltName** extension value
2. X.509 **commonName** attribute of the **subject** field's Distinguished Name (DN) value
3. X.509 v3 **iPAddress** field of the**subjectAltName** extension value

One of these fields must match for identification to be considered successful. If either verification or identification fails, the session is rejected and the UCMD Manager terminates.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-verify_host_name *option*		✓	✓	✓
Environment Variable	UCMDVERIFYHOSTNAME=*option*	✓	✓	✓	
Configuration File Keyword	verify_host_name *option*	✓	✓	✓	✓
STRUCM Parameter	VFYHSTNM(*option*)	✓			

Values

option is the specification for whether or not the X.509 certificate identity is verified.

Valid values for *option* are:

- **no**
  Certificate identity is not verified.
- **yes**
  Certificate identity is verified using the host name specified by the REMOTE_HOST option.
- *hostname*_
  Certificate identity is verified using *hostname*. The value *hostname* can be a DNS host name or an IP address.

Default is no.

VERIFY_SERIAL_NUMBER - UCMD Manager configuration option

Description

The VERIFY_SERIAL _NUMBER option specifies a serial number which must be matched by the serial number of a verified Universal Broker X.509 certificate.

Certificate verification consists of verifying that the certificate is issued by a trusted CA. The CA_CERTIFICATES option specifies which CA certificates are considered trusted.

If either the certificate is not verified or the serial numbers do not match, the session is rejected and the UCMD Manager terminates.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-verify_serial_number *number*		✓	✓	✓
Environment Variable	UCMDVERIFYSERIAL NUMBER=*number*	✓	✓	✓	
Configuration File Keyword	verify_serial_number *number*	✓	✓	✓	✓
STRUCM Parameter	VFYSERNUM(*number*)	✓			

Values

number is the serial number to be matched by the X.509 certificate serial number.

number can be specified in a hexadecimal format by prefixing it with *0x* or *0X*. For example, the value *0x016A392E7F* would be considered a hexadecimal format.

VERSION - UCMD Manager configuration option

Description

The VERSION option writes the program version information and copyright.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-v		✓	✓	✓
Command Line, Long Form	-version		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCM Parameter	VERSION(***option*)	✓			

Values

(There are no values to be specified for this option, except for IBM i.)

IBM i

Valid values for *option* are:

- **yes**
  Write program version information and copyright.
- **no**
  Do not write program version information and copyright.

Default is no.

Universal Command Server for z/OS

Overview

The UCMD Server is responsible for:

1. Accepting a request from a UCMD Manager
2. Establishing network connections with the Manager
3. Creating another process to execute the Manager's command
4. Transmitting output and input files between the Server and Manager
5. Returning the command's exit code to the Manager

The Broker, at the Manager's request, starts the Server. The Server processes the request from the Manager and then exits. A new Server process is created for each Manager request. There may be zero or more Servers running simultaneously, processing requests from different Managers.

Environment

The UCMD Server for z/OS runs as a z/OS UNIX System Services (z/OS USS) background process started by the Universal Broker. The address space name is **UCMSRV**. Its user identifier is inherited from the Universal Broker address space.

All components dependent upon Universal Broker, such as UCMD Server, inherit the message language from Universal Broker. All messages issued by components are sent to the Universal Broker for writing.

Detailed Information

The following pages provide detailed information for the Universal Command Server for z/OS:

- Universal Command Server for z/OS - Commands
- Universal Command Server for z/OS - Component Definition
- Universal Command Server for z/OS - Configuration
- Universal Command Server for z/OS - UACL

Universal Command Server for zOS - Commands

Overview

There are three types of work that a z/OS Universal Command Server can execute:

1. z/OS USS commands and scripts
2. Started Tasks
3. Command References

In all cases, the work executes in its own address space with its own user identity. No Universal Agent programs share the address space with the unit of work started by the Server.

z/OS UNIX System Services Command

The UCMD Server's default command type is the z/OS USS shell. This can be customized with the COMMAND_TYPE configuration option. USS shell commands are executed in a USS process within its own address space.

A UCMD Manager requests the execution of a USS command by specifying a COMMAND_TYPE of **shell**. USS scripts are requested by specifying a SCRIPT_TYPE of **shell**.

The environmental attributes of the user process are described in the following sections.

User Identification

UCMD Server can operate with user security active or inactive, based on the USER_SECURITY configuration option.

- With user security active, the UCMD Server requires the UCMD Manager to supply a valid z/OS user ID and a password. The user process executes with the user ID and the primary and secondary group IDs of the user. The user profile must have a properly defined OMVS segment.

- With user security inactive, the Server does not require the Manager to supply a valid user ID. The user process executes with the user ID of the Server. The Server inherits its user ID from the Broker started task, which is a superuser account (UID 0). The superuser account provides a lot of access to the operating system that a user process typically does not require. Setting security inactive is not recommended because of the level of access it permits the user process.

Working Directory

The working directory of a user process depends on whether user security is active or inactive:

- With user security active, a user process's working directory is the home directory of the user ID as defined in the user profile's OMVS segment HOME parameter value.

- With user security inactive, a user process's working directory is the working directory as defined by the Universal Broker's user profile OMVS segment HOME parameter value. All user processes executed will use the same directory. Care should be taken to avoid name clashes and other consequences of multiple processes sharing a working directory.

Command Shell

The UCMD Manager LOGIN option and the UCMD Server LOGIN option determine what command shell is used.

For non-login environments, the default is shell **/bin/sh**. The shell used for non-login environments is configurable with the SHELL option.

For login environments, the shell defined in the user ID's OMVS segment with the SHELL option is used. The shell environment is created as if the user logged on interactively. For example, the shell's **.profile** is used to initialize the environment.

The non-login environment is similar to the environment that the **cron** scheduler provides. User resource files, such as **.profile**, are not utilized.

The application scripts being executed and your local system management policies should be used to determine which method is best.

If user security is inactive, the default shell **/bin/sh** always is used independent of the SHELL option.

Environment Variables

Environment variables are inherited from the Universal Command Server, which in turn inherits them from the Universal Broker. If security is active, certain variables are modified to match the user environment: HOME, LOGNAME, USER, PWD, and SHELL. Their values are updated to reflect the values for the new environment.

The following variables are added if not found in the environment: HOME, USER, SHELL, and UCMDENV. The UCMDENV variable is set to a value of *1*. It can be used within scripts to determine if Universal Command has invoked them.

The UCMD Manager LOGIN option and the UCMD Server LOGIN option have an impact on the environment variables defined. For login environments, the user's shell is invoked as a login shell, which, in turn, uses the shell profile file in the user's home directory. So any environment variables set in the profile file also will be defined.

If user security is inactive, no changes are made to the environment variables.

Started Tasks

The Universal Command Server has the ability to execute z/OS started tasks. Started tasks have some advantages over USS commands. They execute z /OS programs using standard JCL. The JCL must be predefined in a system procedure library.

UCMD Managers refer to the started task by name and optionally provide an input file and JCL overrides. A Manager requests the execution of a started task by specifying a COMMAND_TYPE of **stc**.

Started task requests are processed by the Universal Command Server Command Processor for Started Tasks (UCMSCPST). The Command Processor (CP) is executed by the Server as a USS process within its own address space.

The STC CP execution environment is the same as the USS command environment described in #z/OS UNIX System Services Command.

Extended MCS Console

The started task is started with the START system command through an extended MCS console. Refer to the IBM *MVS System Commands* manual for a complete description of the START command.

The extended MCS console is established with the following attributes:

Extended MCS Attribute	Value
Command Authority	System commands (SYS)
Console Key (used in DISPLAY C command)	STNBRNCH
Console Name	UNVSnnnn, where nnnn is 0000 - 9999.
Command Scope	Current system
Message Scope	Current system
Override User Profile OPERPARM	Yes

Extended MCS consoles can be protected so that only permitted users have the authority to issue commands. The RACF OPERCMDS class is used to establish user security for extended MCS consoles.

Refer to the IBM *MVS Planning: Operations* and the *Security Server RACF SecurityAdministrator Guide* manuals for complete details.

START System Command

The UCMD Manager provides the START command parameters. The STC CP adds parameter STDIN with a value of a cataloged dynamically allocated data set that contains the standard input from the Manager.

The syntax of the START command is as follows:

```
S manager-cmd,STDIN=stdin-dataset
```

The **manager-cmd** value is the command value provided by the UCMD Manager. The **stdin-dataset** value is the dynamically allocated data set that contains the Manager's standard input data.

As an example, the following Manager command, executed from a Windows system:

```
ucmd -c "prdtask,opt=abc" -cmd_type stc -u ts0023 ...
```

results in a START command as follows:

```
S PRDTASK,OPT=ABC,STDIN=TS0023.UCM.C08AD835.STDIN
```

Access to UCMD Manager started task requests and the associated command value can be protected with Universal Access Control Lists. See Universal Command Server for zOS - UACL for complete details on protecting request types.

Standard Input

A Manager can provide an input file to the started task via the UCMD Manager's standard input file. The Manager's standard input file is first spooled to a cataloged data set. The fully qualified data set name is passed to the started task as JCL procedure parameter STDIN.

The dynamically allocated stdin data set is allocated with a name formatted as: ***hlq*****.UCM.C_cid_.STDIN**

In this format:

- ***hlq*** - High-level qualifier is one of the following:
    - User ID with which the STC is executed.
    - Value of the configuration option STDIN_HLQ.
- **cid** - Component ID of the STC CP. The value is the last seven of eight digits of the component ID in a hexadecimal format.

Standard input data sets dynamically allocated by the UCMD Server are deleted after the STC completes execution.

The UCMD Server's default stdin data set attributes are set with the DEFAULT_STDIN_ALLOC configuration option. The default values are DSORG=PS, RECFM=VB, LRECL=1024, UNIT=SYSDA, SPACE=(CYL,(5,5),RLSE). The UCMD Manager, optionally, can provide data set attributes using the Manager SERVER_OPTIONS value specifying the Server STDIN_ALLOC option described below.

Instead of the Manager providing a standard input file, the Manager may provide the name of an existing data set allocated on the Server system. That is accomplished with a Manager SERVER_OPTIONS value specifying the Server STDIN_ALLOC option described below.

Standard Output and Error

The JES SYSOUT produced by the STC can be returned to the UCMD Manager as standard output and standard error. The STC JESLOG data (JESMSGLG, JESJCL, and JESYSMSG data) is returned as standard error. All step SYSOUT data is returned as standard output.

The STC CP will retrieve SYSOUT data after the STC completes execution. The SYSOUT must be spooled to the JES class specified by the UCMD Server JES_SELECT_CLAS option. Additionally, the SYSOUT data must be held. Released SYSOUT is not retrieved.

Each SYSOUT file is retrieve and written to the appropriate standard I/O file. Message UNV2435I prefixes each SYSOUT file. The message lists the ddname, step, procstep, and spool data set name of each SYSOUT file. The maximum number of records returned per SYSOUT file is controlled with the UCMD Server JES_MAX_LINES_READ configuration option.

After the SYSOUT files are retrieved, their disposition is controlled by the JES_DELETE_SPOOL_FILE and JES_SELECT_CLAS Server options.

JCL Requirements

The started task JCL can specify a job or a procedure. Job JCL must come from either the IEFJOBS or IEFPDSI ddnames of the master JCL. Procedure JCL comes from either the IEFPDSI or JES procedure libraries.

In determining on whether to use job or procedure JCL, refer to the IBM *MVS JCLReference* manual for a description of the advantages and disadvantages.

The first step of the started task must execute the Universal Started Task Support program, UCMSS000. The STC Support program establishes the user ID for the address space and performs necessary communication with the STC CP.

The following figure illustrates a started task procedure JCL.

```
//UCMREQ   PROC
//*
//UCMSS000 EXEC PGM=UCMSS000
//STEPLIB  DD   DISP=SHR,DSN=UNV.SUNVLOAD
//STDIN    DD   DISP=SHR,DSN=&STDIN
//SYSUDUMP DD   SYSOUT=H
//*
//S1       EXEC PGM=ABC123
//SYSOUT   DD  SYSOUT=A,HOLD=YES
//SYSPRINT DD  SYSOUT=A,HOLD=YES
//SYSIN    DD  DISP=SHR,DSN=&STDIN
```

The JCL executes two steps:

1. The first step executes the Universal Command Started Task Support program.
2. The second step executes program ABC123. (This second step - and any subsequent steps - can be any z/OS program.)

Note

The UCMSS00 step includes a STDIN ddname that uses the STDIN procedure variable. This is a JCL convention only to help eliminate one particular source of JCL errors when the source JCL is a procedure (not applicable for job JCL). Any procedure parameter (for example, STDIN) specified on the START command must be reference within the JCL. If it is not, a JCL error is the result. By using the STDIN JCL parameter in the first step, JCL errors caused by not using the parameter are eliminated. The UCMSS000 program does not attempt to use the STDIN ddname in any way.

The UCMSS000 program accepts one input parameter on the PARM keyword of the EXEC statement. The parameter SWUSR controls whether or not the address space user ID is switched or not. The format of the parameter is:

SWUSR={YES|NO}

A value of **YES** specifies the user ID is switched. A value of **NO** specifies the user ID is not switched. The default is **YES**.

Command References

A command reference is a file on a Universal Command (UCMD) Server system that contains a pre-defined command or script.

The UCMD Manager requests execution of a command reference by specifying:

- Name of the command reference, in the COMMAND option.
- **cmdref** (command reference) value, in the COMMAND_TYPE option.

The UCMD Server searches the system for its directory of command references, as specified in the UCMD Server CMD_REFERENCE_DIRECTORY option, and executes the command or script in the command reference.

The UCMD Manager does not provide a command or script; everything is defined within the command reference. This provides the ability to define and control precisely what is executed by the UCMD Server.

Optionally, UCMD Managers provide an input file (via standard input) and options.

Command references are defined as PDS members. The command reference PDS is allocated to the UNVCREF ddname of the Broker started task.

For example, the following UCMD Manager command can be used from Windows or UNIX to request execution of the command reference **cref100** and pass it options **opt1** and **opt2**:

```
ucmd -c "cref100 opt1,opt2" -cmd_type cmdref ...
```

z/OS command references can define any valid command type, such as USS shell commands and scripts and started task commands.

For a complete discussion of Command References, see Command References.

USS Command Reference Example

The following command reference executes a **ucopy** command to read a file.

```
# Command reference to read a file.
#

-format cmd
-type   shell


<eof>



ucopy /opt/application/file.txt
```

STC Command Reference Example

The following command reference starts started task SCHEDINT.

```
# Command reference to scheduler interface.
#

-format cmd
-type   stc

<eof>



SCHEDINT,OPT=ABC
```

Universal Command Server for zOS - Component Definition

Component Definition

All Universal Agent components managed by Universal Broker have a component definition. The component definition is a text file of options containing component-specific information required by Universal Broker.

The syntax of a component definition file is the same as a configuration file.

The UCMD Server for z/OS component definition is located in the component definition library **UNVCOMP** allocated to the Universal Broker ddname **UNVCOMP**. The UCMD Server component definition member is **UCMCMP00**.

Component Definition Options

The following table identifies all of the options that comprise the UCMD Server for z/OS component definition. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AUTOMATICALLY_START	Specification for whether or not UCMD Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCMD Server
CONFIGURATION_FILE	Member name of the UCMD Server configuration file in the **UNVCONF** library allocated to the Broker ddname **UNVCONF**
RUNNING_MAXIMUM	Maximum number of UCMD Servers that can run simultaneously
START_COMMAND	Member name of the UCMD Server program
WORKING_DIRECTORY	HMS directory used as the working directory of the UCMD Server

Universal Command Server for zOS - Configuration

Overview

Universal Command Server configuration consists of defining runtime and default values via configuration options.

Manager Override

A UCMD Manager can specify certain UCMD Server configuration options when it makes its request for command execution to the UCMD Server. The UCMD Manager command line option **-server** (SERVER_OPTIONS) is used to specify UCMD Server options.

Which options are available for manager override depend on the UCMD Server platform and release. UCMD Server configuration options specify a manager override option only if applicable (see Universal Command Server Configuration Options). If override is not specified, no UCMD Manager override is available.

The UCMD Manager is not notified of override errors. The UCMD Server logs the error and continues processing the request.

Configuration File

The configuration file provides the simplest method of specifying configuration options whose values you do not want changed with each command invocation.

The UCMD Server configuration file name is specified in the Universal Command Server component definition. The default name is **UCSCFG00**. The name refers to a member in the PDS allocated to the Universal Broker ddname **UNVCONF**.

Note

⚠ For any changes to the UCMD Server configuration file to become active, a Universal Broker refresh is required, or the Universal Broker started task must be restarted.

Configuration Options

The following table identifies all UCMD Server for z/OS configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
ALLOW_SPOOLING	Specification for whether or not spooling is permitted.
CODE_PAGE	Code page used for text translation.
COMMAND_TYPE	Default command type.
DATA_AUTHENTICATION	Specification for whether or not data integrity checks are performed on all standard I/O files.
DATA_COMPRESSION	Specification for whether or not data is compressed on all standard I/O files.
DATA_ENCRYPTION	Specification for whether or not data is encrypted on all standard I/O files.
DATA_SSL_CIPHER_LIST	SSL/TLS cipher list for the control sessions.
DEFAULT_STDIN_ALLOC	Default STC standard input data allocation parameters.
EVENT_GENERATION	Events to be generated and processed as persistent events.
JES_DELETE_SPOOL_FILE	Specification for whether or not selected STC SYSOUT is deleted.
JES_MAX_LINES_READ	Maximum number of records read from selected STC SYSOUT.
JES_REQUEUE_CLAS	JES class in which selected STC SYSOUT is re-queued.

JES_SELECT_CLAS	JES class from which STC SYSOUT is selected.
JOB_RETENTION	Number of seconds that a disconnected server remains active after user process completes.
KEEPALIVE_INTERVAL	Specifies if and how often a keepalive message is sent.
LOGIN	Specifications for setup and login environment.
MESSAGE_LEVEL	Level of messages written.
MIN_SSL_PROTOCOL	Minimum SSL/TLS protocol level that will be negotiated and used for communications channels.
MSG_SUPPRESSION_LIST	List of message IDs representing Universal messages to be suppressed.
NETWORK_FAULT_TOLERANT	Specification for whether or not the server accepts the network fault tolerant protocol.
SCRIPT_TYPE	Script type of the user job being run.
SHELL	Default shell interpreter.
STDIN_ALLOC	STC standard input data set allocation parameters.
STDIN_HLQ	STC standard input data high-level qualifier.
STDIO_TIMEOUT	Wait time for standard I/O to close before the server process exits.
TMP_DIRECTORY	Name of the directory used for temporary files.
TRACE_FILE_LINES	Maximum number of lines to write to a trace file.
TRACE_TABLE	Memory trace table specification.
UCMSCPST_PATH	Absolute path to an external link that resides on the z/OS UNIX file system. The link points to the UCMD Server STC command processor, UCMSCPST.
USER_SECURITY	Specification for whether or not user authentication is active.

Universal Command Server for zOS - UACL

Overview

Universal Command Server uses the Universal Access Control List (UACL) file as an extra layer of security. The UACL file contains Universal Command Server entries that contain Access Control List (ACL) rules that permit or deny access to the Server.

See Universal Access Control List for details on the Universal Access Control List feature.

UACL Entries

The syntax of a UACL entry file is the same as the Universal Command configuration file.

The following table identifies all Universal Command Server for z/OS UACL entries. Each **UACL Entry** is a link to detailed information about that option.

UACL Entry Name	Description
UCMD_ACCESS	Allows or denies access to Universal Command Server services
UCMD_REQUEST	Allows or denies access to Universal Command Server services based on client identification and request type

UACL Entry Precedence

Deny or Allow Access

The UCMD_ACCESS rules are searched first for an entry that matches the client request. If a UCMD_ACCESS entry is found and the rule denies access to the UCMD Manager, the search stops and the UCMD Manager request is denied.

If a UCMD_ACCESS entry is not found or a UCMD_ACCESS rule allows access, the UCMD_REQUEST entries are searched. If a UCMD_REQUEST entry is found, its rule determines whether the UCMD Manager request is denied or allowed.

If no rules are found, the UCMD Manager request is allowed.

Authenticate or No Authenticate Access

The UCMD_ACCESS entries are searched followed by the UCMD_REQUEST entries.

- If a UCMD_REQUEST entry is found, it sets the authentication requirement.
- If a UCMD_REQUEST entry is not found and an UCMD_ACCESS entry is found, the UCMD_ACCESS rule sets the authentication requirement.
- If no rules are found, the UCMD Manager request requires authentication.

The following table identifies the UACL entry precedence rules described above.

ucmd_access	ucmd_access	ucmd_request	ucmd_request	Result
Allow/Deny	Auth/Noauth	Allow/Deny	Auth/Noauth	
NO-MATCH	NO-MATCH	NO-MATCH	NO-MATCH	ALLOW, AUTH
DENY	n/a	n/a	n/a	DENY
ALLOW	AUTH	NO-MATCH	NO-MATCH	ALLOW, AUTH
ALLOW	AUTH	ALLOW	AUTH	ALLOW, AUTH
ALLOW	AUTH	ALLOW	NOAUTH	ALLOW, NOAUTH

ALLOW	AUTH	DENY	n/a	DENY
ALLOW	NOAUTH	NO-MATCH	NO-MATCH	ALLOW, NOAUTH
ALLOW	NOAUTH	ALLOW	AUTH	ALLOW, AUTH
ALLOW	NOAUTH	ALLOW	NOAUTH	ALLOW, NOAUTH

Universal Command Server for Windows

Overview

The UCMD Server is responsible for:

1. Accepting a request from a UCMD Manager
2. Establishing network connections with the Manager
3. Creating another process to execute the Manager's command
4. Transmitting output and input files between the Server and Manager
5. Returning the command's exit code to the Manager

The Broker, at the Manager's request, starts the Server. The Server processes the request from the Manager and then exits. A new Server process is created for each Manager request. There may be zero or more Servers running simultaneously, processing requests from different Managers.

Server Environment

The UCMD Server for Windows runs as a background process. It does not interact with a console or desktop.

All components dependent upon Universal Broker (such as UCMD Server) inherit their message language from the Broker. All messages issued by components are sent to the Universal Broker for writing.

Detailed Information

The following pages provide detailed information for the Universal Command Server for Windows:

- Universal Command Server for Windows - Commands
- Universal Command Server for Windows - Component Definition
- Universal Command Server for Windows - Configuration
- Universal Command Server for Windows - UACL

Universal Command Server for Windows - Commands

Overview

There are two types of work that a UCMD Server can execute:

1. Commands and Scripts
2. Command References

In all cases, the work executes in its own address space with its own user identity. No Universal Agent programs share the address space with the unit of work started by the UCMD Server.

Command Environment

The user command executes in its own process. No Universal Agent programs share the process with the user command. The process environment consists of several attributes that are described in this section.

User Identification

UCMD Server can operate with user security active or inactive, as specified by the USER_SECURITY configuration option.

- With user security active, the UCMD Server requires the UCMD Manager to supply a valid local system user account or a domain account and a password. The user command executes with the user account identified by the supplied user ID.

  UCMD Managers specify a domain user account as **DOMAIN\USER**.

- With user security inactive, the UCMD Server does not require the UCMD Manager to supply a valid user ID. The user command executes with the user account of the UCMD Server. The user account of the UCMD Server is inherited from the Universal Broker.

  The Universal Broker service can be configured to run as the Local System account or as an Administrative account with substantial authority. When the UCMD Server is configured to run with security inactive, the user process will inherit a lot of privileges which it may not need. Setting security inactive is not recommended because of the level of access it permits the user process.

Working Directory

UCMD Server can operate with user security active or inactive, as specified by the USER_SECURITY configuration option.

- With user security active, a user command's working directory is a subdirectory of the Universal Command Home directory, which defaults to **`\ProgramFiles\Universal\UcmdHome`**. The name of the subdirectory is the user ID with which the command executes. For example, if user HOGIN executes a command via Universal Command, the commands working directory is **`\ProgramFiles\Universal\UcmdHome\HOGIN`**.

  If the working directory is not defined when the user command executes, the UCMD Server creates the directory before it executes the user command.

- With user security inactive, a user command's working directory is the UCMD Server's working directory. All user commands executed use the same directory. Care should be taken to avoid name clashes or other consequences of multiple processes sharing a working directory.

Command Shell

The default command interpreter used to execute commands and scripts is **CMD.EXE**. This commonly is referred to as the DOS command processor. The path to the **CMD.EXE** program is obtained from the **COMPSPEC** environment variable, or if **COMPSPEC** is not defined, the path is derived from the **WINDIR** environment variable as **`%WINDIR\system32\cmd.exe`**. If that fails, the server exits with an error.

Manager-supplied script files are processed as batch files (extension .BAT) by default. The file type (that is, the extension) can be changed with the SCRIPT_TYPE UCMD Server configuration option or the SCRIPT_TYPE UCMD Manager option.

Environment Variables

UCMD Server inherits its environment variables from the Universal Broker. In turn, the user command inherits its environment variables from the UCMD Server. The UCMD Server does not add, delete, or edit any environment variables.

Command References

A command reference is a file on a Universal Command (UCMD) Server system that contains a pre-defined command or script.

The UCMD Manager requests execution of a command reference by specifying:

- Name of the command reference, in the COMMAND option.
- **cmdref** (command reference) value, in the COMMAND_TYPE option.

The UCMD Server searches the system for its directory of command references, as specified in the UCMD Server CMD_REFERENCE_DIRECTORY option, and executes the command or script in the command reference.

The UCMD Manager does not provide a command or script; everything is defined within the command reference. This provides the ability to define and control precisely what is executed by the UCMD Server.

A command reference also can specify that the command or script accepts user-provided options / command line arguments from the UCMD Manager.

For example, the following UCMD Manager command can be used from Windows or UNIX to request execution of the command reference **cref100** and pass it options **opt1** and **opt2**:

```
ucmd -c "cref100 opt1,opt2" -cmd_type cmdref ...
```

Command references can define any valid command type, such as commands and scripts.

Command Reference Example

The following command reference executes a **ucopy** command to read a file.

```
# Command reference to read a file.
#

-format cmd
-type   shell


<eof>



ucopy \application\file.txt
```

Universal Command Server for Windows - Component Definition

Component Definition

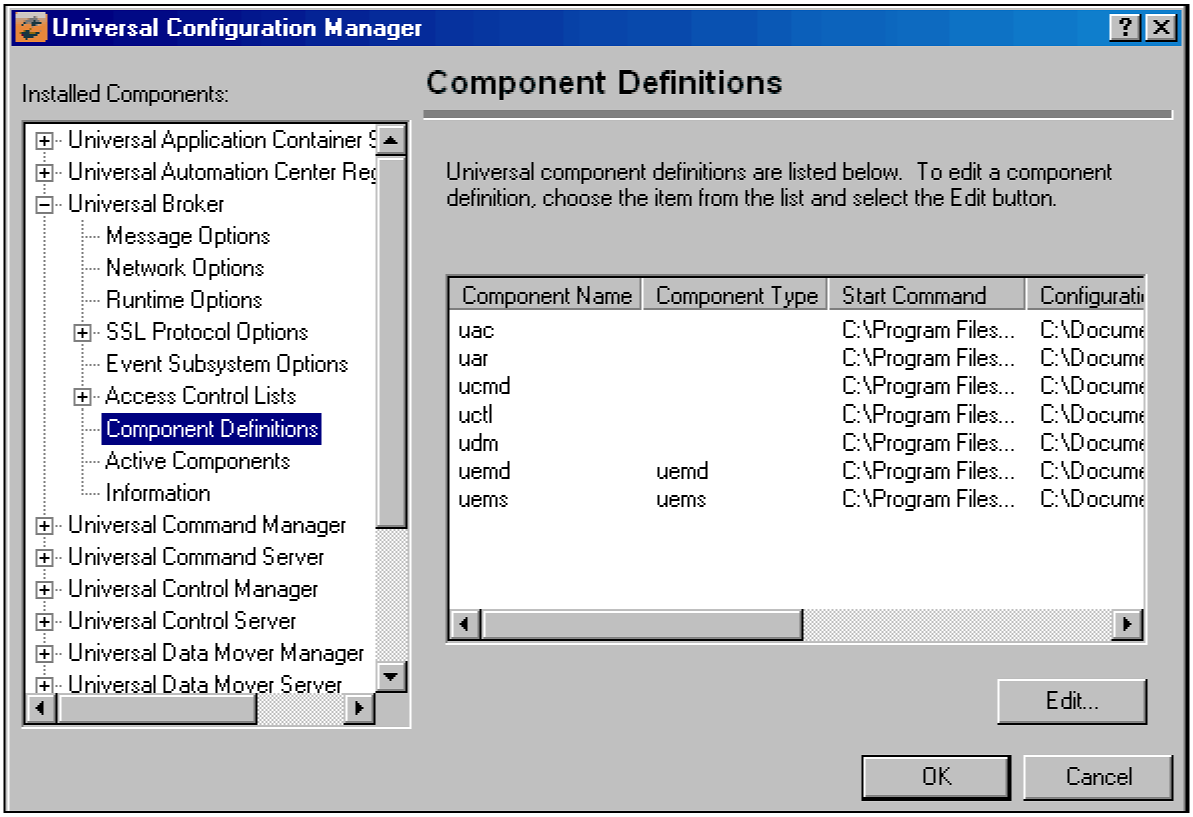
All Universal Agent components managed by Universal Broker have a component definition. The component definition is a text file of options containing component-specific information required by Universal Broker.

The syntax of a component definition file is the same as a configuration file.

Although component definition files can be edited with any text editor (for example, Notepad), the Universal Configuration Manager application is the recommended way to edit component definitions for Windows.

Note

The component definitions for all Universal Agent components are identified in the Component Definitions property page of the Universal Broker (as shown below).



Universal component definitions are listed below. To edit a component definition, choose the item from the list and select the Edit button.

Component Name	Component Type	Start Command	Configurati
uac		C:\Program Files...	C:\Docume
uar		C:\Program Files...	C:\Docume
ucmd		C:\Program Files...	C:\Docume
uctl		C:\Program Files...	C:\Docume
udm		C:\Program Files...	C:\Docume
uemd	uemd	C:\Program Files...	C:\Docume
uems	uems	C:\Program Files...	C:\Docume

Component Definition Options

The following table identifies all of the options that comprise the UCMD Server for Windows component definition. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AUTOMATICALLY_START	Specification for whether or not UCMD Server starts automatically when Universal Broker is started.

COMPONENT_NAME	Name by which the clients know the UCMD Server
CONFIGURATION_FILE	Name of the UCMD Server configuration file
RUNNING_MAXIMUM	Maximum number of UCMD Servers that can run simultaneously
START_COMMAND	Full path name of the UCMD Server program
WORKING_DIRECTORY	Directory used as the working directory of the UCMD Server

Universal Command Server for Windows - Configuration

Configuration

UCMD Server configuration consists of defining runtime and default values. This section describes the UCMD Server configuration options.

Manager Override

A UCMD Manager can specify certain UCMD Server configuration options when it makes its request for command execution to the UCMD Server. The UCMD Manager command line option **-server** (SERVER_OPTIONS) is used to specify UCMD Server options.

Which options are available for manager override depend on the UCMD Server platform and release. UCMD Server configuration options specify a manager override option (see Universal Command Server Configuration Options) only if applicable. If override is not specified, no UCMD Manager override is available.

The UCMD Manager is not notified of override errors. The UCMD Server logs the error and continues processing the request.

Configuration File

The configuration file provides the simplest method of specifying configuration options whose values you do not want changed with each command invocation.

The UCMD Server configuration file name is specified in the UCMD Server component definition. The default name is **ucmds.conf**.

Although configuration files can be edited with any text editor (for example, Notepad), the Universal Configuration Manager application, accessible via the Control Panel, is the recommended way to set configuration options.

The Universal Configuration Manager provides a graphical interface and context-sensitive help, and helps protect the integrity of the configuration file by validating all changes to configuration option values.

Note

⚠ For any changes made directly to the UCMD Server configuration file to become active, a Universal Broker refresh is required, or the Universal Broker service must be restarted. Changes made by the Universal Configuration Manager do not require any additional action for the options to become active.

Configuration Options

The following table identifies all UCMD Server for Windows configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
ALLOW_SPOOLING	Specification for whether or not spooling in permitted.
ASSIGN_PROCESS_TO_JOB	Specification for whether or not UCMD Server assigns child processes to a single Windows job object.
CMD_REFERENCE_DIRECTORY	Command reference directory.
CODE_PAGE	Code page used for text translation.
COMMAND_TYPE	Default command type.
DATA_AUTHENTICATION	Specification for whether or not data integrity checks are performed on all standard I/O files.
DATA_COMPRESSION	Specification for whether or not data is compressed on all standard I/O files.
DATA_ENCRYPTION	Specification for whether or not data is encrypted on all standard I/O files.
DATA_SSL_CIPHER_LIST	SSL/TLS cipher list for the control sessions.

ELEVATE_USER_TOKEN	Specification for whether or not the UCMD Server assigns an elevated user token to the child process that it executes. An elevated user token is one that is not subject to User Account Control (UAC) restrictions.
EVENT_GENERATION	Events to be generated and processed as persistent events.
INSTALLATION_DIRECTORY	Base directory in which UCMD Server is installed.
INTERACT_WITH_DESKTOP	Specification for whether or not the desktop of the current interactive logon session is accessible to the user process.
JOB_RETENTION	Number of seconds a disconnected server remains active after user process completes.
KEEPALIVE_INTERVAL	Specification for if and how often a keepalive message is sent.
LOGIN	Login information.
LOGON_METHOD	Method of how users are logged onto the system.
MESSAGE_LEVEL	Level of messages written.
MIN_SSL_PROTOCOL	Minimum SSL/TLS protocol level that will be negotiated and used for communications channels.
MSG_SUPPRESSION_LIST	List of message IDs representing Universal messages to be suppressed.
NLS_DIRECTORY	Location of UMC and UTT files.
SCRIPT_TYPE	Script type of the user job being run.
SPOOL_DIRECTORY	Location of spool files.
STDIO_TIMEOUT	Specifies the time in seconds to wait for Standard I/O to close before the server process exits.
TMP_DIRECTORY	Name of the directory used for temporary files.
TRACE_DIRECTORY	Location of trace files.
TRACE_FILE_LINES	Maximum number of lines to write to a trace file.
TRACE_TABLE	Memory trace table specification.
USER_SECURITY	Specification for whether or not user authentication is active.

Universal Command Server for Windows - UACL

Universal Access Control List

Universal Command Server uses the Universal Access Control List (UACL) file as an extra layer of security. The UACL file contains entries for the Universal Command Server. These entries contain Access Control List (ACL) rules that permit or deny access to the Universal Command Server.

See Universal Access Control List for details on the Universal Access Control List feature.

UACL Entries

The syntax of a UACL file is the same as the Universal Command configuration file.

The following table identifies all Universal Command for Windows UACL entries. Each **UACL Entry Name** is a link to detailed information about that option.

UACL Entry Name	Description
UCMD_ACCESS	Allows or denies access to Universal Command Server services
UCMD_REQUEST	Allows or denies access to Universal Command Server services based on client identification and request type

Universal Command Server for UNIX

Overview

The UCMD Server is responsible for:

1. Accepting a request from a UCMD Manager
2. Establishing network connections with the Manager
3. Creating another process to execute the Manager's command
4. Transmitting output and input files between the Server and Manager
5. Returning the command's exit code to the Manager

The Broker, at the Manager's request, starts the Server. The Server processes the request from the Manager and then exits. A new Server process is created for each Manager request. There may be zero or more Servers running simultaneously, processing requests from different Managers.

Server Environment

The UCMD Server for UNIX runs as a background process. It does not interact with a console.

All components dependent upon Universal Broker (such as UCMD Server) inherit the message language from the Universal Broker. All messages issued by components are sent to the Universal Broker for writing.

Detailed Information

The following pages provide detailed information for the Universal Command Server for UNIX:

- Universal Command Server for UNIX - Commands
- Universal Command Server for UNIX - Component Definition
- Universal Command Server for UNIX - Configuration
- Universal Command Server for UNIX - UACL

Universal Command Server for UNIX - Commands

Overview

There are two types of work that a UCMD Server can execute:

1. Commands and Scripts
2. Command References

In all cases, the work executes in its own address space with its own user identity. No Universal Agent programs share the address space with the unit of work started by the UCMD Server.

User Identification

UCMD Server can operate with user security active or inactive, as specified by the USER_SECURITY configuration option.

- With user security active, the UCMD Server requires the UCMD Manager to supply a valid user ID for the local system and a password. The user command executes with the user ID and the primary and secondary group ID's of the user.
- With user security inactive, the UCMD Server does not require the UCMD Manager to supply a valid user ID. The user command executes with the user account of the UCMD Server. The user account of the UCMD Server is the superuser account (UID 0).
  The superuser account provides a lot of access to the operating system that a user process typically does not require. Setting security inactive is not recommended because of the level of access it permits the user process.

Working Directory

The working directory of a user command depends on whether user security is active or inactive:

- With user security active, a user command's working directory is the home directory of the user ID as defined in the **`/etc/passwd`** file.
- With user security inactive, a user command's working directory is the UCMD Server's working directory. All user commands executed use the same directory. Care should be taken to avoid name clashes and other consequences of multiple processes sharing a working directory.

Command Shell

The UCMD Manager LOGIN option and UCMD Server LOGIN option specify what shell is used.

For non-login environments, the default is shell **/bin/sh**. The shell used for non-login environments is configurable with the SHELL option.

For login environments, the shell associated with the user ID found in the **/etc/passwd** file is used. The shell environment is created as if the user logged on interactively. For example, the **korn** shell's **.profile** is used to initialize the environment.

The non-login environment is similar to the environment the **cron** scheduler provides. User resource files, such as **.profile**, are not utilized.

The application scripts being executed and your local system management policies should be used to determine which method is best.

If user security is inactive, the default shell **/bin/sh** always is used independent of the LOGIN option.

Environment Variables

Environment variables are inherited from the server, which in turn inherits them from the broker.

If security is active, certain variables are modified to match the user environment. They are HOME, LOGNAME, USER, PWD, and SHELL. Their values are updated to reflect the values for the new environment.

The following variables are added if not found in the environment: HOME, USER, SHELL, and UCMDENV. The UCMDENV variable is set to a value of *1*. It can be used within scripts to determine if Universal Command has invoked them.

The UCMD Manager LOGIN option and UCMD Server LOGIN option have an impact on the environment variables defined. For login environments, the user's shell is invoked as a login shell, which, in turn, uses the shell **.profile** file in the user's home directory. Therefore, any environment variables set in the **.profile** file also will be defined. UCMD Server inherits its environment variables from the Universal Broker. In turn, the user command inherits its environment variables from the UCMD Server.

If user security is inactive, no changes are made to the environment variables.

Command References

A command reference is a file on a Universal Command (UCMD) Server system that contains a pre-defined command or script.

The UCMD Manager requests execution of a command reference by specifying:

- Name of the command reference, in the COMMAND option.
- **cmdref** (command reference) value, in the COMMAND_TYPE option.

The UCMD Server searches the system for its directory of command references, as specified in the UCMD Server CMD_REFERENCE_DIRECTORY option, and executes the command or script in the command reference.

The UCMD Manager does not provide a command or script; everything is defined within the command reference. This provides the ability to define and control precisely what is executed by the UCMD Server.

A command reference also can specify that the command or script accepts user-provided options / command line arguments from the UCMD Manager.

For example, the following UCMD Manager command can be used from Windows or UNIX to request execution of the command reference **cref100** and pass it options **opt1** and **opt2**:

```
ucmd -c "cref100 opt1,opt2" -cmd_type cmdref ...
```

Command references can define any valid command type, such as commands and scripts.

For a complete discussion of Command References, see Command References.

Command Reference Example

The following command reference executes a **ucopy** command to read a file.

```
# Command reference to read a file.
#

-format cmd
-type   shell


<eof>



ucopy /application/file.txt
```

Universal Command Server for UNIX - Component Definition

Overview

All Universal Agent components managed by Universal Broker have a component definition. The component definition is a text file of options containing component-specific information required by Universal Broker.

The syntax of a component definition file is the same as a configuration file.

The UCMD Server for UNIX component definition is located in the component definition directory of the Universal Broker.

Component Definition Options

The following table identifies all of the options that comprise the UCMD Server for UNIX component definition. Each Option Name is a link to detailed information about that option

Option Name	Description
AUTOMATICALLY_START	Specification for whether or not UCMD Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCMD Server
CONFIGURATION_FILE	Name of the UCMD Server configuration file
RUNNING_MAXIMUM	Maximum number of UCMD Servers that can run simultaneously
START_COMMAND	Full path name of the UCMD Server program
WORKING_DIRECTORY	Directory used as the working directory of the UCMD Server

Universal Command Server for UNIX - Configuration

Configuration

Universal Command Server configuration consists of defining runtime and default values via Server configuration options.

See Configuration Management for details on Universal Agent configuration methods.

Manager Override

A UCMD Manager can specify certain UCMD Server configuration options when it makes its request for command execution to the UCMD Server. The UCMD Manager command line option **-server** (SERVER_OPTIONS) is used to specify UCMD Server options.

Which options are available for manager override depend on the UCMD Server platform and release. UCMD Server configuration options specify a manager override option (see Universal Command Server Configuration Options) only if applicable. If override is not specified, no UCMD Manager override is available.

The UCMD Manager is not notified of override errors. The UCMD Server logs the error and continues processing the request.

Configuration File

The configuration file provides the simplest method of specifying configuration options whose values you do not want changed with each command invocation.

The UCMD Server configuration file name is specified in the Universal Command Server component definition. The default name is **ucmds.conf**. See the component definition file to determine the directory in which it is located. This file can be edited manually with any text editor.

Note

For any changes made directly to the UCMD Server configuration file to become active, a Universal Broker refresh is required, or the Universal Broker daemon must be restarted.

Configuration Options

The following table identifies all UCMD Server for UNIX configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
ALLOW_SPOOLING	Specification for whether or not spooling in permitted.
CMD_REFERENCE_DIRECTORY	Command reference directory.
CODE_PAGE	Code page used for text translation.
COMMAND_TYPE	Default command type.
DATA_AUTHENTICATION	Specification for whether or not data integrity checks are performed on all standard I/O files.
DATA_COMPRESSION	Specification for whether or not data is compressed on all standard I/O files.
DATA_ENCRYPTION	Specification for whether or not data is encrypted on all standard I/O files.
DATA_SSL_CIPHER_LIST	SSL/TLS cipher list for the control sessions.
EVENT_GENERATION	Events to be generated and processed as persistent events.
INSTALLATION_DIRECTORY	Base directory in which UCMD Server is installed.
JOB_RETENTION	Number of seconds a disconnected server remains active after user process completes.

KEEPALIVE_INTERVAL	Specification for if and how often a keepalive message is sent.
LOGIN	Login information.
MESSAGE_LEVEL	Level of messages written.
MIN_SSL_PROTOCOL	Minimum SSL/TLS protocol level that will be negotiated and used for communications channels.
MSG_SUPPRESSION_LIST	List of message IDs representing Universal messages to be suppressed.
NETWORK_FAULT_TOLERANT	Specification for whether or not the server accepts the network fault tolerant protocol.
NLS_DIRECTORY	Location of UMC and UTT files.
SCRIPT_TYPE	Script type of the user job being run.
SHELL	Default shell interpreter.
SPOOL_DIRECTORY	Location of spool files.
STDIO_TIMEOUT	Specifies the time in seconds to wait for Standard I/O to close before the server process exits.
TMP_DIRECTORY	Name of the directory used for temporary files.
TRACE_DIRECTORY	Location of trace files.
TRACE_FILE_LINES	Maximum number of lines to write to a trace file.
TRACE_TABLE	Memory trace table specification.
USER_SECURITY	Specification for whether or not user authentication is active.

Universal Command Server for UNIX - UACL

Overview

Universal Command Server uses the Universal Access Control List (UACL) file as an extra layer of security. The UACL file contains Universal Command Server entries that contain Access Control List (ACL) rules that permit or deny access to the Server.

See Universal Access Control List for details on the Universal Access Control List feature.

UACL Entries

The syntax of a UACL entry file is the same as the Universal Command configuration file.

The following table identifies all Universal Command for UNIX UACL entries. Each **UACL Entry Name** is a link to detailed information about that entry.

UACL Entry Name	Description
UCMD_ACCESS	Allows or denies access to Universal Command Server services.
UCMD_REQUEST	Allows or denies access to Universal Command Server services based on client identification and request type.

UACL Entry Precedence

Deny or Allow Access

The **ucmd_access** rules are searched first for an entry that matches the client request. If an **ucmd_access** entry is found and the rule denies access to the Manager, the search stops and the Manager request is denied.

If no **ucmd_access** entry is found or an **ucmd_access** rule allows access, the **ucmd_request** entries are searched. If an **ucmd_request** entry is found, its rule determines whether the Manager request is denied or allowed.

If no rules are found, the Manager request is allowed.

Authenticate or No Authenticate Access

The **ucmd_access** entries are searched followed by the **ucmd_request** entries.

- If an **ucmd_request** entry is found, it sets the authentication requirement.
- If no **ucmd_request** entry is found and an **ucmd_access** entry is found, the **ucmd_access** rule sets the authentication requirement.
- If no rules are found, the Manager request requires authentication.

The following table identifies the UACL entry precedence rules described above.

ucmd_access	ucmd_access	ucmd_request	ucmd_request	Result
Allow/Deny	Auth/Noauth	Allow/Deny	Auth/Noauth	
NO-MATCH	NO-MATCH	NO-MATCH	NO-MATCH	ALLOW, AUTH
DENY	n/a	n/a	n/a	DENY
ALLOW	AUTH	NO-MATCH	NO-MATCH	ALLOW, AUTH
ALLOW	AUTH	ALLOW	AUTH	ALLOW, AUTH
ALLOW	AUTH	ALLOW	NOAUTH	ALLOW, NOAUTH

ALLOW	AUTH	DENY	n/a	DENY
ALLOW	NOAUTH	NO-MATCH	NO-MATCH	ALLOW, NOAUTH
ALLOW	NOAUTH	ALLOW	AUTH	ALLOW, AUTH
ALLOW	NOAUTH	ALLOW	NOAUTH	ALLOW, NOAUTH

Universal Command Server for IBM i

Currently, IBM i runs Workload Automation 5.1.1.   These pages provide information for that version.

Overview

The UCMD Server is responsible for:

1. Accepting a request from a UCMD Manager
2. Establishing network connections with the Manager
3. Creating another process to execute the Manager's command
4. Transmitting output and input files between the Server and Manager
5. Returning the command's exit code to the Manager

The Broker, at the Manager's request, starts the Server. The Server processes the request from the Manager and then exits. A new Server process is created for each Manager request. There may be zero or more Servers running simultaneously, processing requests from different Managers.

Server Environment

The Universal Command Server runs under the **UNVUBR511** subsystem's pre-start job UCMSRV. When the Broker receives a request for a Universal Command component, it passes the request to the UCMSRV program running under the UCMSRV pre-start job.

All components dependent upon Universal Broker (such as Universal Command Server) inherit the message language from the Broker. All messages issued by components are sent to the Universal Broker for writing.

Detailed Information

The following pages provide detailed information for the Universal Command Server for IBM i:

- Universal Command Server for IBM i - Commands
- Universal Command Server for IBM i - Component Definition
- Universal Command Server for IBM i - Configuration
- Universal Command Server for IBM i - UACL

Universal Command Server for IBM i - Commands

User Command Environment

The user request is initiated by the Universal Command Server Initiator program (**UCMSINIT**) running under the **UCMSINIT** pre-start job via the **UCMSRV** program running under the Universal Command Server (**UCMSRV**) job.

The UCMSINIT program:

1. Performs environment setup required to execute user commands or scripts.
2. Redirects the job log as requested.
3. Changes the user profile if user security is active.
4. Initiates the user request.
5. Monitors the user's request as it runs.
6. Catches any exceptions that occur.

Following completion of the users request, the **UCMSRV** program:

1. Processes the **UCMSINIT** job logs as required.
2. Returns the **UCMSINIT** job logs to the Universal Command Manager and, optionally, keeps a job log copy on the local iSeries server, as requested.
3. Cleans up the environment.

The **UCMSRV** program also handles fault tolerant requests from the Universal Command Manager.

By default, the **UCMSINIT** job log is returned to the Universal Command Manager via the standard error output stream (stderr). The spool file output produced by the executed commands is written to the user's default spool output queue. The spool files produced by the commands are not returned to the Universal Command Manager. They are left in the spool. If you would like the command output returned to the Universal Command Manager as well as the job log, see Universal Submit Job for an execution method that returns all command spool files as well as the job log.

Both the **UCMSINIT** and **UCMSRV** pre-start jobs are defined in the **UNVUBR511** subsystem. By default, there are always at least three each of the **UCMSINIT** and UCMSRV pre-start jobs running under the **UNVUBR511** subsystem.

Initiator (UCMSINIT) Exit Points

The Initiator (**UCMSINIT**) calls two user exits:

1. **UCMSJOBI** is called once for job initialization.
2. **UCMSJOBT** is called once for job termination.

The CL source code is provided in **UNVPRD511/UNVCLSRC**.

The CL source files are compiled and bound with the following command:

```
CRTBNDCL PGM(UNVPRD511/exitname)
               SRCFILE(UNVPRD511/UNVCLSRC)
               SRCMBR(exitname)
```

Change the **exitname** to the name of the exit to be compiled and bound.

UCMSJOBI

The **UCMSJOBI** exit is called before any user command is executed. This exit can be used to customize the job's environment to meet local requirements. It executes under the user profile requested by the Manager. If the exit issues unhandled messages with a severity greater than or equal to the value of the UCMD Server END_SEVERITY option, the job will terminate without executing any user commands.

UCMSJOBI sets the current library to the current library specified in the user profile under which the job runs. However, if the UCMD Server LOGIN option is enabled, there may be a conflict between the current library set by **UCMSJOBI** and by the LOGIN functionality. To avoid this conflict, a new **UCMSJOBILOGIN** parameter specifies that when LOGIN is active, **UCMSJOBI** no longer sets the current library.

If the 3.2.0 (or later) version of **UCMSJOBI** is used to replace **UCMSJOBI** on a 3.1.1 system, an exception will occur because of the new **LOGIN** parameter.

Note

⚠

If **UCMSJOBI** from a previous release (3.1.1) of UCMD Server for IBM i is copied in place of the 3.2.0 version, the program will run with the potential conflict.

UCMSJOBT

The **UCMSJOBT** exit is called after all user commands have completed. The current exit code is passed in as a parameter. The exit executes under the user profile requested by the UCMD Manager. The exit will always be called once **UCMSJOBI** returns successfully. If **UCMSJOBI** issues an unhandled message that causes job termination, **UCMSJOBT** will not be called.

User Command Exit Code

The exit code returned to the UCMD Manager indicates the success or failure of the user-requested command. The exit code is returned to the UCMD Manager by the UCMD Server (UCMSRV) based on the exit code that it receives from the Initiator (**UCMSINIT**). The Initiator sets its exit code based on the highest severity of the IBM i messages propagated to it from user commands or scripts. The Initiator traps and handles all *ESCAPE, *NOTIFY, *STATUS messages and function checks.

UCMSINIT continues executing user commands as long as the highest message severity is less than the severity specified by the UCMD Server END_SEVERITY option.

In the event of an error not associated with the user-requested command, **UCMSINIT** returns exit code 99. If the error occurs following set up for returning the **UCMSINIT** job log to the user, the job log is returned as usual. Otherwise, no job log is returned and the user must check the output queues for a job log associated with the failure.

Depending on the job's logging settings and the severity of the error, no job jog may be saved. If an exception results in job termination, the returned exit code will be based on the numeric portion of the message identifier. The last four hexadecimal digits are used for the return code, with the most common being 9901 (decimal 39169). However, partial truncation, to the lower three digits, also occurs for managers running on some platforms.

If a job ends as a result of the **ENDJOB** command, whether issued directly or indirectly, the exit code will be the special value 199.

User Identification

UCMD Server can operate with user security active or inactive, based on the user security configuration option.

- With user security active, the UCMD Server requires the UCMD Manager to supply a valid user ID and password for the local system. The user command executes with the user profile of the received user ID.
- With user security inactive, the UCMD Server does not require the UCMD Manager to supply a valid user ID. The user command executes with the user profile of the UCMD Server. The user profile of the UCMD Server is inherited from the Universal Broker. The inherited user profile is **UNVUBR511**; as installed, this profile provides a very high level of authority including *ALLOBJ, *SPLCTL, and *JOBCTL.

Current Library and Working Directory

The current library and working directory of a user command depends on whether user security is active or inactive:

- With user security active, the user's current library and working directory is the home directory of the user profile specified in the UCMD Manager.
- With user security inactive, the current library and the working directory are those for the user profile associated with the service program. The default user profile defined and associated with the service program at installation is **UNVUBR511**.

Note

⚠

The default value used for the current library in the **UNVUBR511** user profile is **UNVTMP511**. Care should be taken to avoid name clashes and other consequences of multiple processes sharing a common current library and working directory.

User Commands

UCMD Server accepts four forms of commands from a UCMD Manager.

1. Single CL command
2. Single REXX line
3. CL command file
4. REXX EXEC file

Single CL Command

The remote UCMD Manager specifies a CL command using the COMMAND (**-cmd**) option.

The CL command must be of a type that can be executed by the QCMDEXC API. This is indicated by the command description in the CL Reference manual by the keyword Exec in the upper right corner of the command's syntax diagram. Limit **-cmd** option length to 1000 bytes.

Single REXX Line

The remote UCMD Manager specifies a single REXX line using the COMMAND (**-cmd**) option with the COMMAND_TYPE (**-cmd_type**) option of *rexx*.

REXX and any associated commands must be of a type that can be executed by the QCMDEXC API as described in Single CL Command above. Limit COMMAND option length to 1000 bytes.

Multiple statements contained in the single line command must be separated by semicolons as described in the REXX manuals. The first statement does no require a REXX comment.

For example, the following command sends the text "'Change current library to ABC" to standard output and changes the current library:

```
ucmd -c "say 'Change current library to ABC'; \"CHGCURLIB CURLIB(ABC) \"" -i as400 -u qsysopr -w qsysopr -
cmd_type rexx
```

A user may use a simple REXX program in this context to setup and execute programs on the IBM i.

REXX provides the benefit of using standard output (STDOUT) and standard input (STDIN) files as part of their environment. The SAY command writes to STDOUT and the PULL command reads from STDIN.

STDOUT from REXX on the IBM i is redirected back to STDIN of the Command Manager and REXX STDIN on the IBM i is redirected from the STDOUT of the Command Manager. CL command files do not use STDOUT or STDIN directly.

CL Command File

The remote UCMD Manager specifies a CL command file using the SCRIPT_FILE (-script) option.

The command file contains a sequence of CL commands to be executed in sequential order. The commands are executed from first to last or until a command generates a message with a severity greater than or equal to the UCMD Server END_SEVERITY option.

The CL commands are limited to the same set of CL commands described in Single CL commands above.

Each command is executed within the same job environment. This is similar to a batch job execution, but // CL commands cannot be used.

Blank lines and CL comment lines are ignored in the command file.

CL line continuation characters (+ and -) can be used in the command file and are processed accordingly.

The first line cannot be a comment line containing the word REXX.

REXX EXEC File

The remote UCMD Manager specifies a REXX EXEC file by using the SCRIPT_FILE (-script) option.

This same UCMD Manager option is used to specify a CL command file. To distinguish between a REXX EXEC and a CL command file, the first line of the file containing the REXX EXEC must be a comment line containing the word REXX. The case of the letters does not matter.

For example, the following line is sufficient to indicate a REXX EXEC file:

```
/* REXX */
```

REXX EXECs have the benefit of using standard output (STDOUT) and standard input (STDIN) files as part of their environment. The SAY command writes to STDOUT and the PULL command reads from STDIN.

The STDOUT file is redirected back to the STDOUT of the UCMD Manager and the STDIN is redirected from the STDIN of the UCMD Manager. CL command files do not directly use STDOUT and STDIN.

Command References

A command reference is a file on a Universal Command (UCMD) Server system that contains a pre-defined command or script.

The UCMD Manager requests execution of a command reference by specifying:

- Name of the command reference, in the COMMAND option.
- **cmdref** (command reference) value, in the COMMAND_TYPE option.

The UCMD Server searches the system for its directory of command references, as specified in the UCMD Server CMD_REFERENCE_DIRECTORY option, and executes the command or script in the command reference.

The UCMD Manager does not provide a command or script; everything is defined within the command reference. This provides the ability to define and control precisely what is executed by the UCMD Server.

Optionally, UCMD Managers provide an input file (via standard input) and options.

Command references that execute IBM i commands or REXX scripts are defined as files in the UNVCMDREF library. Each file can contain only one member. For security reasons, the library name is set to UNVCMDREF; this name cannot be changed or redefined.

For example, the following UCMD Manager command can be used from Windows or UNIX to request execution of the command reference **cref100** and pass it options **opt1** and **opt2**:

```
ucmd -c "cref100 opt1,opt2" -cmd_type cmdref ...
```

IBM i command references can define command types **cmd** and **rexx**.

For a complete discussion of Command References, see Command References.

Command Reference Example

The following command reference contains a command to display a library catalog.

To use this **cmdref**, invoke UCMD Manager using:

```
ucmd -c "cmdref_cmd" -cmd_type cmdref -u xxxx -w zzzz -i as400
```

In this case, the user (**xxxx**) has authority to call the IBM i system object **QWCRJBST** and **cmdref_cmd** is the name of the command reference file on the IBM i.

```
# -- File named cmdref_cmd in library UNVCMDREF --
# Use USBMJOB to execute the DSPLIB command.
#
-format cmd
-type cmd

<eof>


usbmjob cmd(dsplib qsysopr)
```

The next command reference contains a series of four commands which are executed in sequence from top to bottom.

Invoke these commands from system **as4test** using:

```
ucmd -c "cref_test" -cmd_type cmdref -u xxxx -w zzzz -i as4test
```

In this case, the user (**xxxx**) has authority to call the IBM i system object QWCRJBST and **cref_test** is the name of the command reference file on the IBM i.

```
# -- File named cref_test in library UCVCMDREF --
# Execute a series of commands.  The output of the first two will remain
# on the output queue associated with the job's printer.  The output of
# the second two (by means of USBMJOB) will be directed to standard output
# and sent to the system running the Universal Command Manager.
#
-format script
-type cmd

<eof>



DSPUSRPRF USRPRF(QUSER)
dspcurdir
usbmjob cmd(dsplib quser)
usbmjob cmd(dsplibl)
```

REXX Command Reference Example

The following command reference sends the message and the library catalog to standard output. The job logs are sent to standard error.

To invoke this **cmdref** on system denver, use:

```
ucmd -c "rexx_test" -cmd_type cmdref -u xxxx -w zzzz -i denver
```

In this case, the user (**xxxx**) has authority to call the IBM i system object **QWCRJBST** and **rexx_test** is the name of the command reference file on the IBM i.

Once again, the user, designated by the **-u** option, requires access to the object **QWCRJBST**. Without this access, **usbmjob** will fail.

```
# -- File named rexx_test in library UCVCMDREF --
# Use USBMJOB to execute the DSPLIB command.
#
-format script
-type rexx

<eof>


say "Submitting job to display library qsysopr"
'usbmjob cmd(dsplib qsysopr)'
```

Universal Command Server for IBM i - Component Definition

Overview

All Universal Agent components managed by Universal Broker have a component definition. The component definition is a text file of options containing component-specific information required by Universal Broker.

The syntax of a component definition file is the same as a configuration file.

UCMD Server for IBM i component definition is located in the component definition file of the Universal Broker. The default location for Universal Broker component definition files is **UNVPRD511/UNVCOMP**. The UCMD Server component member is **UCMD**.

Component Definition Options

The following table identifies all of the options that comprise the UCMD for IBM i component definition. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AUTOMATICALLY_START	Specification for whether or not UCMD Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCMD Server
CONFIGURATION_FILE	Name of the UCMD Server configuration file
RUNNING_MAXIMUM	Maximum number of UCMD Servers that can run simultaneously
START_COMMAND	Full path name of the UCMD Server program
WORKING_DIRECTORY	Directory used as the working directory of the UCMD Server

Universal Command Server for IBM i - Configuration

Overview

UCMD Server configuration consists of defining runtime and default values. This section describes the Server configuration options.

Manager Override

A UCMD Manager can specify certain UCMD Server configuration options when it makes its request for command execution to the UCMD Server. The UCMD Manager command line option -server is used to specify UCMD Server options.

Which options are available for UCMD Manager override depend on the UCMD Server platform and release. The configuration options listed below describe the UCMD Manager override option only if applicable. If the option is not listed, than no UCMD Manager override is available.

The UCMD Manager is not notified of override errors. The UCMD Server logs the error and continues processing the request.

Configuration File

The configuration file provides the simplest method of specifying configuration options whose values you do not want changed with each command invocation.

Configuration options are specified in the UCMD Server configuration file. The configuration file name is specified in the UCMD Server component definition. The default file name is **UNVPRD511/UNVCONF(UCMDS)**. This file can be edited manually with any text editor (for example, Notepad or Source Edit Utility (SEU)).

Configuration Options

The following table identifies all UCMD Server for IBM i configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
ALLOW_SPOOLING	Specification for whether or not spooling is permitted.
CODE_PAGE	Code page used for text translation.
COMMAND_TYPE	Default command type.
DATA_AUTHENTICATION	Specification for whether or not data integrity checks are performed on all standard I/O files.
DATA_COMPRESSION	Specification for whether or not data is compressed on all standard I/O files.
DATA_ENCRYPTION	Specification for whether or not data is encrypted on all standard I/O files.
DATA_SSL_CIPHER_LIST	SSL/TLS cipher list for the control sessions.
END_SEVERITY	Message severity that terminates the initiator.
EVENT_GENERATION	Events to be generated and processed as persistent events.
JOB_LOG	Job log processing.
JOBLOG_COPY_KEEP	Specification for whether or not copies of the joblog from the UCMSINIT job and (optionally) from the job started with USBMJOB are written to a spool file.
JOB_RETENTION	Number of seconds that a disconnected server remains active after user process completes.

KEEPALIVE_INTERVAL	Frequency of how often a keepalive message is sent.
LOGIN	Specification for whether or not to set up a login environment.
MESSAGE_LEVEL	Level of messages written.
MIN_SSL_PROTOCOL	Minimum SSL/TLS protocol level that will be negotiated and used for communications channels. Note This option was introduced to IBM i in version 5.1.1.0.
NETWORK_FAULT_TOLERANT	Specification for whether or not the server accepts the network fault tolerant protocol.
SCRIPT_TYPE	Script type of the user job being run.
STDIO_TIMEOUT	Length of time to wait for standard I/O to close before the server process exits.
TRACE_FILE_LINES	Maximum number of lines to write to a trace file.
TRACE_TABLE	Memory trace table specification.
USE_USER_ACCOUNTING_CODE	Specification for whether or not the IBM i user profile under which a process is run is to be used as the source for the job accounting code
USER_SECURITY	Specification for whether or not user authentication is active.

Universal Command Server for IBM i - UACL

Overview

Universal Command Server uses the Universal Access Control List (UACL) file as an extra layer of security. The UACL file contains Universal Command Server entries that contain Access Control List (ACL) rules that permit or deny access to the Server.

See Universal Access Control List for details on the Universal Access Control List feature.

UACL Entries

The syntax of a UACL entry file is the same as the Universal Command configuration file.

The following table identifies all Universal Command for IBM i UACL entries. Each **UACL Entry Name** is a link to detailed information about that entry.

UACL Entry Name	Description
UCMD_ACCESS	Allows or denies access to Universal Command Server services
UCMD_REQUEST	Allows or denies access to Universal Command Server services based on client identification and request type

UACL Entry Precedence

Deny or Allow Access

The **ucmd_access** rules are searched first for an entry that matches the client request. If an **ucmd_access** entry is found and the rule denies access to the Manager, the search stops and the Manager request is denied.

- If no **ucmd_access** entry is found or an **ucmd_access** rule allows access, the **ucmd_request** entries are searched. If an **ucmd_request** entry is found, its rule determines whether the Manager request is denied or allowed.
- If no rules are found, the Manager request is allowed.

Authenticate or No Authenticate Access

The **ucmd_access** entries are searched followed by the **ucmd_request** entries.

- If a **ucmd_request** entry is found, it sets the authentication requirement.
- If no **ucmd_request** entry is found and a **ucmd_access** entry is found, the **ucmd_access** rule sets the authentication requirement.
- If no rules are found, the Manager request requires authentication.

The following table identifies the UACL entry precedence rules described above.

ucmd_access	ucmd_access	ucmd_request	ucmd_request	Result
Allow/Deny	Auth/Noauth	Allow/Deny	Auth/Noauth	
NO-MATCH	NO-MATCH	NO-MATCH	NO-MATCH	ALLOW, AUTH
DENY	n/a	n/a	n/a	DENY
ALLOW	AUTH	NO-MATCH	NO-MATCH	ALLOW, AUTH
ALLOW	AUTH	ALLOW	AUTH	ALLOW, AUTH
ALLOW	AUTH	ALLOW	NOAUTH	ALLOW, NOAUTH
ALLOW	AUTH	DENY	n/a	DENY

ALLOW	NOAUTH	NO-MATCH	NO-MATCH	ALLOW, NOAUTH
ALLOW	NOAUTH	ALLOW	AUTH	ALLOW, AUTH
ALLOW	NOAUTH	ALLOW	NOAUTH	ALLOW, NOAUTH

Universal Command Server Configuration Options

Universal Command Server Configuration Options

This page provides links to detailed information on the configuration options available for use with the Universal Command Server.

The options are listed alphabetically, without regard to any specific operating system.

Configuration Options Information

For each configuration option, this page provides the following information.

Description

Describes the option and how it is used.

Usage

Provides a table of the following information:

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	<Format / Value>				
Manager Override	<Format / Value>				

Method

Identifies the different methods used to specify Universal Command Server configuration options:

- Configuration File Keyword
- Manager Override

Note

Each option can be specified using one or more methods.

Syntax

Identifies the syntax of each method that can be used to specify the option:

- Format: Specific characters that identify the option.
- Value: Type of value(s) to be supplied for this method.

Note

If a Method is not valid for specifying the option, the Syntax field contains **n/a**.

(Operating System)

Identifies the operating systems for which each method of specifying the option is valid:

- IBM i
- HP NonStop
- UNIX
- Windows
- z/OS

Values

Identifies all possible values for the specified value type.

Defaults are identified in **bold type**.

<Additional Information>

Identifies any additional information specific to the option.

Configuration Options List

The following table identifies the Universal Command Server configuration options.

Option Name	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
ALLOW_SPOOLING	Specification for whether or not spooling is permitted.
ASSIGN_PROCESS_TO_JOB	Specification for whether or not UCMD Server assigns child processes to a single Windows job object.
CMD_REFERENCE_DIRECTORY	Command reference directory.
CODE_PAGE	Code page used for text translation.
COMMAND_TYPE	Default command type.
DATA_AUTHENTICATION	Specification for whether or not data integrity checks are performed on all standard I/O files.
DATA_COMPRESSION	Specification for whether or not data is compressed on all standard I/O files.
DATA_ENCRYPTION	Specification for whether or not data is encrypted on all standard I/O files.
DATA_SSL_CIPHER_LIST	SSL/TLS cipher list for the control sessions.
DEFAULT_STDIN_ALLOC	Default STC standard input data allocation attributes.
ELEVATE_USER_TOKEN	Specification for whether or not the UCMD Server assigns an elevated user token to the child process that it executes. An elevated user token is one that is not subject to User Account Control (UAC) restrictions.
ENCRYPT_CONTROL_SESSION	Specification for whether or not the control session must be encrypted.
END_SEVERITY	Message severity that terminates the initiator.
EVENT_GENERATION	Events to be generated as persistent events.
INSTALLATION_DIRECTORY	Base directory in which UCMD Server is installed.
INTERACT_WITH_DESKTOP	Specification for whether or not the desktop of the current interactive logon session is accessible to the user process.
JES_DELETE_SPOOL_FILE	Specification for whether or not selected STC SYSOUT is deleted.
JES_MAX_LINES_READ	Maximum number of records read from selected STC SYSOUT.
JES_REQUEUE_CLAS	JES class in which selected STC SYSOUT is re-queued.
JES_SELECT_CLAS	JES class from which STC SYSOUT is selected.
JOB_LOG	Job log processing.
JOBLOG_COPY_KEEP	Specification for whether or not copies of the joblog from the UCMSINIT job and (optionally) from the job started with USBMJOB are written to a spool file.
JOB_RETENTION	Number of seconds a disconnected server remains active after user process completes.
KEEPALIVE_INTERVAL	Specification for whether or not a keepalive message is sent, and if so, how often.
LOGIN	Specification for whether or not to set up a login environment.
LOGON_METHOD	Specification for how users are logged onto the system.
MESSAGE_LEVEL	Level of messages written.

MIN_SSL_PROTOCOL	Minimum SSL/TLS protocol level that will be negotiated and used for communications channels. Note This option was introduced to IBM i in version 5.1.1.0.
MSG_SUPPRESSION_LIST	List of message IDs representing Universal messages to be suppressed.
NETWORK_FAULT_TOLERANT	Specification for whether or not the server accepts the network fault tolerant protocol.
NLS_DIRECTORY	Location of UMC and UTT files.
PRIORITY	Execution priority of the user job being run.
SCRIPT_TYPE	Default script type.
SHELL	Default shell interpreter.
SPOOL_DIRECTORY	Spool file directory.
STDIN_ALLOC	STC standard input data set allocation parameters.
STDIN_HLQ	STC standard input data high?level qualifier.
STDIO_TIMEOUT	Time to wait for Standard I/O to close before the server process exits.
TMP_DIRECTORY	Name of directory used for temporary files.
TRACE_DIRECTORY	Trace file directory.
TRACE_FILE_LINES	Maximum number of lines to write to a trace file.
TRACE_TABLE	Memory trace table specification.
UCMSCPST_PATH	Absolute path to an external link that resides on the z/OS UNIX file system. The link points to the UCMD Server STC command processor, UCMSCPST.
USE_USER_ACCOUNTING_CODE	Specification for whether or not the IBM i user profile under which a process is run is to be used as the source for the job accounting code.
USER_SECURITY	Specification for whether or not user authentication is active.

ACTIVITY_MONITORING - UCMD Server configuration option

Description

The ACTIVITY_MONITORING option specifies whether or not product activity monitoring events are generated.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	activity_monitoring *option*	✓	✓	✓	✓
Manager Override	n/a				

Values

option is the specification for whether or not product activity monitoring events are generated.

Valid values for *option* are:

- **yes**
  Activate monitoring events.
- **no**
  Deactivate monitoring events.

Default is yes.

ALLOW_SPOOLING - UCMD Server configuration option

Description

The ALLOW_SPOOLING option specifies whether or not the UCMD Server supports spooling.

Since spooling requires disk space to be allocated, some Servers may want to conserve disk resources.

z/OS

Spool data is written to the HFS database allocated to the Broker UNVSPOOL ddname.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	allow_spooling *option*	✓	✓	✓	✓
Manager Override	n/a				

Values

option is the specification for whether or not spooling is allowed.

Valid values for *option* are:

- **yes**
  Spooling is permitted.
- **no**
  Spooling is not permitted.

Default is no.

ASSIGN_PROCESS_TO_JOB - UCMD Server configuration option

Description

The ASSIGN_PROCESS_TO_JOB option controls the startup and shutdown behavior of UCMD Server processes executed on Windows.

- If ASSIGN_PROCESS_TO_JOB is set to **yes**, UCMD Server assigns all of its child processes to a system resource known as a job object.
- If ASSIGN_PROCESS_TO_JOB is set to **no**, child processes are not assigned to a job object, and no relationship among parent / child processes is maintained.

ASSIGN_PROCESS_TO_JOB simplifies process management by forcing all child processes to end whenever a user terminates UCMD Server. Prior to UCMD Server 3.2.0.0, the only child process guaranteed to terminate along with UCMD Server was its immediate, well-known child process, **cmd.exe**. Consequently, its child process (that is, the command or script specified from UCMD Manager) could continue to run even after cancelling UCMD Server. When cancelling a UCMD Server that assigns its well-known child process to a job, the entire process tree ends.

If the command or script specified from UCMD Manager must continue to run even after cancelling a UCMD Server, simply set ASSIGN_PROCESS_TO_JOB to **no**. In that case, process termination behaves exactly as it did prior to UCMD Server 3.2.0.0.

Note

The affect of ASSIGN_PROCESS_TO_JOB on UCMD Server's behavior is limited to Server cancellation. All other UCMD Server behavior – including, but not limited to, standard I/O redirection, network and host fault tolerance, and detection of process completion – is unchanged.

Although ASSIGN_PROCESS_TO_JOB is available only for a Windows UCMD Server, a UCMD Manager (3.2.0.0 or later) on any operating system can override it for any single UCMD Server instance.

A Stonebranch Tip

Assigning a UCMD Server child process to a job means that Windows will automatically assign all processes that child spawns to that job.

If any of those child processes creates its own job object with the expectation that it is not part of any other job, some unexpected behavior may occur.

In these situations, either set ASSIGN_PROCESS_TO_JOB to **no** or change the application's behavior.

For more information, refer to Windows Platform SDK documentation for the CreateProcess, CreateProcessAsUser, and IsProcessInJob functions, and the CREATE_BREAKAWAY_FROM_JOB and JOB_OBJECT_LIMIT_BREAKAWAY_OK options.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	assign_process_to_job *option*			✓	
Manager Override	-assign_process_to_job *option*	✓	✓	✓	✓

Values

option is the specification for whether or not UCMD Server child processes are assigned to a job.

Valid values for *option* are:

- **yes**
  Create a job object and assign all UCMD Server child processes to it.
- **no**
  Do not create a job object.

Default is yes.

CMD_REFERENCE_DIRECTORY - UCMD Server configuration option

Description

The CMD_REFERENCE_DIRECTORY option specifies the name of the directory that contains command reference files.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	cmd_reference_directory *option*	✓	✓	✓	
Manager Override	n/a				

Values

option is the name of the directory.

Full path names are recommended.

Default

IBM i	Default is **`UNVCMDREF`**.
UNIX	Default is **`/var/opt/universal/cmdref`**.
Windows	Default is **`C:\Program Files\Universal\cmdref`**.

CODE_PAGE - UCMD Server configuration option

Description

The CODE_PAGE option specifies the character code page that is used to translate text data received and transmitted over the network.

The Universal Translate Table (UTT) files are used to translate between Unicode and the local single-byte code page.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	codepage *codepage*	✓	✓	✓	✓
Manager Override	-codepage *codepage*	✓	✓	✓	✓

Value

codepage is the character code page that is used to translate data.

codepage references a Universal Translate Table (UTT) file provided with the product (see UTT Files). UTT files are used to translate between Unicode and the local single-byte code page. (All UTT files end with an extension of **.utt**.)

Note

UTF-8 is not a supported *codepage* value for CODE_PAGE. UTF-8 codepage is valid only for standard I/O text file translation. Consequently, it can be specified only with the UCMD Manager SIO_LOCAL_CODE_PAGE and SIO_REMOTE_CODE_PAGE options.

See Character Code Pages for a complete list of code pages provided by Stonebranch Inc. for use with Universal Agent.

Default

Default code page is different for different operating systems:

- ISO8859-1 (8-bit ASCII): ASCII-based operating systems
- IBM1047 (EBCDIC): EBCDIC-based operating system

COMMAND_TYPE - UCMD Server configuration option

Description

The COMMAND_TYPE option specifies the default command type if one is not specified by a UCMD Manager COMMAND_TYPE option.

(The UCMD Manager COMMAND_TYPE option specifies the type of command provided by the UCMD Manager COMMAND option.)

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	cmd_type *type*	✓	✓	✓	✓
Manager Override	-cmd_type *type*	✓	✓	✓	✓

Value

type is the command type.

Valid values for *type* are:

Command Type	IBM i	UNIX	Windows	z/OS
cmd	✓			
cmdref	✓	✓	✓	✓
shell		✓	✓	✓
rexx	✓			
stc				✓

Default

IBM i	Default is **cmd**.
UNIX	Default is **shell**.
Windows	Default is **shell**.
z/OS	Default is **shell**.

DATA_AUTHENTICATION - UCMD Server configuration option

Description

The DATA_AUTHENTICATION option specifies whether or not all data sent over the network is authenticated when using the **UNVv2** protocol.

Generating a checksum value for each data block performs authentication. The checksum value is sent with the data block. The receiver generates a second checksum value for the data block. If the checksum values are not equal, the authentication fails. Failed authentication closes the network connection.

The checksum is generated with the MD5 Message Digest Algorithm by RSA Data Security, Inc.

DATA_AUTHENTICATION does not have any effect on the SSL/TLS protocol. See the DATA_SSL_CIPHER_LIST option for SSL/TLS data authentication.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	authenticate *option*	✓	✓	✓	✓
Manager Override	n/a				

Value

option is the specification for whether or not data is authenticated.

Valid values for *option* are:

- **yes**
  Data authentication is required for the **UNVv2** protocol. All network data transfers are authenticated regardless of UCMD Manager's DATA_AUTHENTICATION option.
- **no**
  Data authentication is not required. However, the UCMD Manager still can request data authentication via its DATA_ATUTHENTICATION option.

Default is no.

DATA_COMPRESSION - UCMD Server configuration option

Description

The DATA_COMPRESSION option specifies whether or not data standard I/O file transmissions across the network should be compressed.

Optionally, it also can specify the compression method to use.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	compress *option[UCMD:,method]*	✓	✓	✓	✓
Manager Override	n/a				

Values

option is either of the following values:

- **yes**
  Data compression is required. All data in standard I/O file transmissions is compressed regardless of the UCMD Manager DATA_COMPRESSION option value.
- **no**
  Data compression is not required. However, data compression still can be requested via the UCMD Manager DATA_COMPRESSION option.

Default = no.

method is either of the following values:

- **zlib**
  Data is compressed using ZLIB compression algorithm. This method usually results in a very high compression rate, but tends to be somewhat CPU-intensive. It is recommended in environments where controlling a process's CPU usage is not necessarily a priority.
- **hasp**
  Data is compressed using the HASP compression algorithm. This method is less CPU-intensive than the ZLIB method. It is recommended in environments where controlling CPU usage is a priority. With HASP, the compression rate, while still very good, tends to be a little less than what is possible with the ZLIB.

Default is zlib.

DATA_ENCRYPTION - UCMD Server configuration option

Description

The DATA_ENCRYPTION option specifies whether or not all data sent over the network is encrypted when using the **UNVv2** protocol.

Encryption protects the privacy of the data. **UNVv2** data encryption uses one of several encryption algorithms, such as the Data Encryption Standard (DES) algorithm.

DATA_ENCRYPTION does not have any effect on SSL/TLS protocol. See the DATA_SSL_CIPHER_LIST option for SSL/TLS encryption.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	encrypt *option*	✓	✓	✓	✓
Manager Override	n/a				

Values

option is the specification for whether or not data is encrypted.

Valid values for *option* are:

- **yes**
  Data encryption is required for the **UNVv2** protocol. All network data transfers are encrypted regardless of UCMD Manager's DATA_ENCRYPTION option.
- **no**
  Data encryption is not required. However, UCMD Manager still can request data encryption via its DATA_ENCRYPTION option.

Default is no.

DATA_SSL_CIPHER_LIST - UCMD Server configuration option

Description

The DATA_SSL_CIPHER_LIST option specifies one or more SSL/TLS cipher suites that are acceptable to use for network communications on the data session, which is used for standard I/O file transmission.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	data_ssl_cipher_list *cipherlist*	✓	✓	✓	✓
Manager Override	n/a				

Values

cipherlist is a comma-separated list of SSL/TLS cipher suites. The following table identifies the list of SSL/TLS cipher suites supported for this option.

The list is in default order, with the most preferred suite first and the least preferred suite last.

Cipher Suite	**Description**
AES256-GCM-SHA384	256-bit AES encryption in Galois Counter Mode, SHA-2 384-bit message digest.
AES256-SHA	256-bit AES encryption with SHA-1 message digest.
AES128-GCM-SHA256	128-bit AES encryption in Galois Counter Mode, SHA-2 256-bit message digest.
AES128-SHA	128-bit AES encryption with SHA-1 message digest.
RC4-SHA	128-bit RC4 encryption with SHA-1 message digest.
RC4-MD5	128-bit RC4 encryption with MD5 message digest.
DES-CBC3-SHA	128-bit Triple-DES encryption with SHA-1 message digest.
DES-CBC-SHA	128-bit DES encryption with SHA-1 message digest. Note ⚠ As of Universal Agent 6.7.0.0, DES-CBC-SHA is supported only on HP-UX. Additionally, any Agents on HP-UX that accept connections from, or attempt connections to, Agents on other platforms must be configured with at least one currently supported cipher suite besides DES-CBC-SHA. Therefore, those HP-UX Agents cannot be configured only with DES-CBC-SHA in their list of cipher suites.
NULL-SHA256	No encryption and SHA-2 256-bit message digest.
NULL-SHA	No encryption and SHA-1 message digest.
NULL-MD5	No encryption and MD5 message digest.

Note

⚠

To configure the UCMD Server to accept only encrypted data sessions, do not include the NULL-* ciphers in the list.

DEFAULT_STDIN_ALLOC - UCMD Server configuration option

Description

The DEFAULT_STDIN_ALLOC option specifies DCB allocation attributes for the standard input data set that is allocated for a started task request.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	default_stdin_alloc *options*				✓
Manager Override	n/a				

Values

options is a comma-separated list of DCB allocation attributes, in a format similar to JCL parameters.

The following table identifies the attributes that are allowed.

Attribute	Description
LRECL	Logical record length
BLKSIZE	Block size
SPACE	Space unit, primary space, secondary space, and release option
UNIT	Unit type or group
VOLSER	Volume serial number
DSORG	Data set organization
RECFM	Record format
DSN	Data set name of existing data set
DISP	Disposition status of existing data set

Default is DSORG=PS,RECFM=VB,LRECL=1024,UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE).

ELEVATE_USER_TOKEN - UCMD Server configuration option

Description

The ELEVATE_USER_TOKEN option allows a process to execute with the highest privileges available to it; the UCMD server will execute the process with an elevated user token (that is, one not subject to User Account Control (UAC) restrictions).

Starting with Windows Vista, the Windows User Account Control (UAC) feature allows some privileged operations (for example, a process that takes ownership of a file) to execute only after receiving confirmation to do so. Windows obtains this confirmation when a user responds to a UAC prompt, thereby giving the application permission to proceed. This response elevates the user's access token to a fully-privileged token.

The non-interactive nature of a UCMD Server child processes prevents it from issuing the UAC elevation prompt, so any process that would normally require elevation may fail.

The ELEVATE_USER_TOKEN option solves this problem by instructing UCMD Server to obtain a user's elevated token and use it to execute the child process. This gives the process all privileges available to the user, not just those permitted by UAC.

Note
⚠
Setting this option will not provide the user with any additional privileges. It will simply enable all of the privileges that the user already has been granted.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	elevate_user_token *option*			✓	
Manager Override	elevate_user_token *option*			✓	

Values

option specifies whether or not UCMD Server child processes execute with an elevated user token.

Valid values for *option* are:

- **yes**
  Obtain an elevated user token and use it to execute the child process.
- **no**
  Execute the child process with a default user token. If that is not an elevated user token, some operations subject to UAC restrictions may fail.

Note
⚠
Some accounts - such as the built-in Administrator account - may already be fully elevated. Processes executed with this account may not need to set this option.

Default is no.

ENCRYPT_CONTROL_SESSION - UCMD Server configuration option

Description

The ENCRYPT_CONTROL_SESSION option specifies whether or not the control session must be encrypted.

Application data and commands, including user identifiers and passwords, are transmitted over the network on the control session. Stdio data are transmitted over the stdio sessions. The default protocol SSL/TLS always uses encryption for the control session.

A UCMD Manager can disable SSL/TLS by reverting to the Universal Products (UNVv2) protocol without encryption and data integrity checks.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	encrypt_control_session *option*	✓	✓	✓	✓
Manager Override	n/a				

Values

option is the specification for whether or not the control session is encrypted.

Valid values for *option* are:

- **yes**
  Encryption is required (SSL/TLS and UNVv2 protocols, with encryption, can be used).
- **no**
  Encryption is not required (SSL/TLS and UNVv2 protocols, with or without encryption, can be used).

Default is yes.

END_SEVERITY - UCMD Server configuration option

Description

The END_SEVERITY option specifies the message severity level that must be reached in order for the Initiator job to terminate.

The Initiator job traps messages received from the user commands and records the maximum severity message. When the maximum severity is equal to or greater than the value specified by the END_SEVERITY option, the Initiator terminates without executing any additional user commands.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	end_severity *level*	✓			
Manager Override	-endseverity *level*	✓			

Values

level is the message security level.

Valid values for *level* are positive integers.

Default is 30.

EVENT_GENERATION - UCMD Server configuration option

Description

The EVENT_GENERATION option specifies which types of events are to be generated and processed as persistent events by the Universal Event Subsystem (UES).

A persistent event record is saved in a Universal Enterprise Controller (UEC) database, the UES database (***uec.evm.db***), for long-term storage.

For a list of all event types for all Universal Agent components, see Event Definition Details.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	event_generation *types*		✓	✓	✓
Manager Override	n/a				

Values

type specifies a comma-separated list of event types. It allows for all or a subset of all potential event message types to be selected.

Event type ranges can be specified by separating the lower and upper range values with a dash (**-**) character.

Event types can be selected for inclusion or exclusion:

- Exclusion operator is **X** or **x**.
- An asterisk (*) represents all event types.

Examples

100,101,102	Generate event types 100, 101, and 102.
100-102	Generate event types 100 through 102.
100-102,200	Generate event types 100 through 102 and 200.
*	Generate all event types.
*,X100	Generate all event types except for 100.
X*	Generate no event types.
*,X200-250,X300	Generate all event types except for 200 through 250 and 300.

Default is X* (no event types).

INSTALLATION_DIRECTORY - UCMD Server configuration option

Description

The INSTALLATION_DIRECTORY option specifies the location in which UCMD Server is installed.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	installation_directory *directory*		✓	✓	
Manager Override	n/a				

Values

directory is the location in which UCMD Server is installed.

UNIX	*directory* is the UCMD Server installation file. **Default is `/opt/universal/ucmdsrv`.**
Windows	*directory* is the UCMD Server installation file. **Default is `C:\Program Files\Universal\ucmdsrv`.**

INTERACT_WITH_DESKTOP - UCMD Server configuration option

Description

The INTERACT_WITH_DESKTOP option specifies whether or not user processes are allowed to interact with the current console logon session.

INTERACT_WITH_DESKTOP is applicable only when the LOGON_METHOD option is set to **interactive**. (If LOGON_METHOD is set to **batch**, the established security context already disallows all interaction with the desktop.)

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	interact_with_desktop *option*			✓	
Manager Override	n/a				

Values

option is the specification for whether or not user processes can interact with the current logon session.

Valid values for *option* are:

- **yes**
  User processes run in a context that permits the current interactive console session to interact with them. This interaction can go both ways, as the process may invoke system functions that access desktop elements (for example: Windows, menus, and buttons) associated with the session. This is considered a security risk, in that it creates an opportunity for a malicious process to hijack the desktop. If the security context of the interactive session is higher than that of the process, the process could invoke code using an elevated security context.
- **no**
  User processes run in a context that is isolated from the current interactive logon session. Unless user processes requiring user interaction are executed, this is the recommended value.

Default is no.

JES_DELETE_SPOOL_FILE - UCMD Server configuration option

Description

The JES_DELETE_SPOOL_FILE option specifies whether or not the UCMD Server deletes selected started task SYSOUT files after redirecting them to the UCMD Manager.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	jes_delete_spoolfile *option*				✓
Manager Override	n/a				

Values

option is the specification for whether or not the UCMD Server deletes the SYSOUT files.

Valid values for *option* are:

- **yes**
  Selected JES spool files are deleted.
- **no**
  Selected JES spool files are not deleted.

Default is yes.

JES_MAX_LINES_READ - UCMD Server configuration option

Description

The JES_MAX_LINES_READ option specifies the maximum number of lines, or records, of a selected JES SYSOUT file that is read and transferred to the UCMD Manager.

When this maximum number is reached, that SYSOUT file is closed and processing continues with the next available SYSOUT file. No message is written indicating that the limit has been exceeded.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	jes_max_lines_read *maximum*				✓
Manager Override	n/a				

Values

maximum is the maximum number of lines read and transferred.

Valid values for *maximum* are any integer.

A value of 0 indicates that there is no maximum.

Default is 100000.

JES_REQUEUE_CLAS - UCMD Server configuration option

Description

The JES_REQUEUE_CLAS option specifies the JES class to which that selected JES SYSOUT files are re-queued.

If JES_REQUEUE_CLAS is not specified, the SYSOUT file remains in its current JES class

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	jes_requeue_clas *class*				✓
Manager Override	n/a				

Values

class is the JES class to which files are re-queued.

JES_SELECT_CLAS - UCMD Server configuration option

Description

The JES_SELECT_CLAS option specifies the JES class from which started task JES-held SYSOUT files are selected for transfer back to the UCMD Manager.

Note

The SYSOUT files must be held.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	jes_select_clas *class*				✓
Manager Override	n/a				

Values

class is the JES class from which files are selected.

Default is A.

JOB_LOG - UCMD Server configuration option

Description

The JOB_LOG option specifies how the UCMD Server processes the job log of the Initiator job.

JOB_LOG specifies under what conditions the job log is returned to the **stderr** file of the UCMD Manager. If the job log is returned, it is done so after the Initiator job ends.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	joblog *option*	✓			
Manager Override	-joblog *option*	✓			

Values

option is the specification for how the UCMD Server processes the job log.

Valid values for *option* are:

- **always**
  The job log is always returned to the UCMD Manager on the **stderr** file.
- **error**
  The job log is returned only if the Initiator job's exit code is greater than or equal to the END_SEVERITY option value.
- **never**
  The job log is never returned; it is written to a spool file under the control of the **UCMSINIT** job description LOG value.

Default is always.

Note

If the value is **always** or **error**, the **UCMSINIT** job description must not restrict the production of the job log with the LOG value. A job log either is returned to the UCMD Manager or it is spooled, but not both.

JOB_RETENTION - UCMD Server configuration option

Description

The JOB_RETENTION option specifies the default number of seconds that a disconnected UCMD Server, running with network fault tolerance, waits for a reconnection request from the UCMD Manager after a user process has completed.

This situation could occur if the network connection between the UCMD Manager and UCMD Server is lost. Since the UCMD Server is using network fault tolerance, it waits for the UCMD Manager to reestablish the network connections.

If the user process ends while the UCMD Server and UCMD Manager are disconnected, JOB_RETENTION specifies how long the UCMD Server should wait for the UCMD Manager to reconnect before the UCMD Server ends. If the UCMD Server does end, the UCMD Manager cannot reconnect to it; it must restart.

Note

The UCMD Manager JOB_RETENTION option can override this option.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	job_retention *seconds*	✓	✓	✓	✓
Manager Override	n/a				

Values

option is the number of seconds to wait.

Default is 172,800 seconds (2 days).

JOBLOG_COPY_KEEP - UCMD Server configuration option

Description

The JOBLOG_COPY_KEEP option controls whether or not copies of the joblog from the **UCMSINIT** job and (optionally) from the job started with **USBMJOB** are written to a spool file.

Prior to release 5.1.0, Universal Command Server for IBM i spooled a copy of the job log to the **QEZJOBLOG** output queue via specialized printer files. **USBMJOB** used printer file **USBMJPRTF** and the Universal Command Server used printer file **UCMSVRPRTF**.

As of release 5.1.0, Universal Command Server for IBM i produces system-generated job logs that are stored in the **QEZJOBLOG** output queue, or other system designated job log output queue, using the system job log print file, **QPJOBLOG**.

For the Universal Command joblog, the **UCMSINIT** job description is used. For the user process joblog from **UJOBINIT**, the job description is dependent on the user profile and the submit job parameters for **USBMJOB**.

JOBLOG_COPY_KEEP has no impact on whether or not the joblogs are returned to the Universal Command manager.

Job Options Affecting Job Log Output

IBM i system option **JOBMSGQFL** can affect output spooled to the job log spool file. When **JOBMSGQFL** is set to ***PRTWRAP**, job log messages in the job log message queue are written to the job log spool file before being overwritten in the queue. These messages, once written to the spool file, cannot be read.

Therefore, with JOBLOG_COPY_KEEP = yes, the job log sent via stderr with **JOBMSGQFL** set to ***PRTWRAP** will be the same as when **JOBMSGQFL** is set to ***WRAP**. Stated differently, for job logs returned via stderr with JOBLOG_COPY_KEEP = yes, **JOBMSGQFL** set to ***PRTWRAP** is equivalent to **JOBMSGQFL** set to ***WRAP**.

The **JOBMSGQFL** setting does not affect messages written to the job log spool file with JOBLOG_COPY_KEEP = yes. All job log message are written to the job log spool file just as if Universal Command were not part of the process.

The **JOBMSGQFL** default setting is ***NOWRAP**. When set to ***NOWRAP**, the operating system ends the job when the job log message queue is full.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	joblog_copy_keep *option*	✓			
Manager Override	n/a				

Values

option is the specification for whether or not copies of the job logs are written to a spool file.

Valid values for *option* are:

- **yes**
  Job logs are written to a spool file.
- **no**
  Job logs are not written to a spool file.

Default is no.

Note

The JOB_LOG option does impact the JOBLOG_COPY_KEEP option for the Universal Command job log from **UCMSINIT**.

For example, if the JOB_LOG value is **never**, Universal Command does not touch the **UCMSINIT** joblog. Instead, as with previous versions of Universal Command, the joblog goes to the spool based on the **UCMSINIT** job description.

Basically, JOBLOG_COPY_KEEP impacts the **UCMSINIT** joblog only if JOB_LOG produces a joblog for Universal Command to process. The user process joblog from **UJOBINIT** is always produced when **USBMJOB** is executed, so it will always be processed based on JOBLOG_COPY_KEEP.

KEEPALIVE_INTERVAL - UCMD Server configuration option

Description

The KEEPALIVE_INTERVAL option specifies the frequency with which a Keep-Alive message is sent to the UCMD Manager.

The Keep-Alive message is used to verify the presence of a connection between the UCMD Server and UCMD Manager during periods of network inactivity.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	keep_alive_interval *frequency*	✓	✓	✓	✓
Manager Override	n/a				

Values

frequency is the frequency (in seconds) with which a Keep-Alive message is sent.

Default is 120.

LOGIN - UCMD Server configuration option

Description

The LOGIN option specifies different login information for different operating systems.

UNIX and z/OS	The LOGIN option specifies whether or not the shell program that is used to create the user process is invoked as a login shell. A login shell will read and execute commands from the system profile and the user profile. Which profile files are read and executed depends on the type of shell. A non-login shell will not read and execute commands from the profiles. This matches the environment of a process scheduled with **cron**.
AIX	The AIX platform provides a **/etc/environment** file within which global environmental variables can be exported for all users of the machine. Universal Command adds the environment variables defined in this file to the user's login environment. The order in which it is processed by Universal Command is slightly different than the AIX login process. The AIX login process reads and processes the following files in the order listed: 1. /etc/profile 2. /etc/environment 3. $HOME/.profile 4. $HOME/.env The LOGIN option directs Universal Command to use the login shell to execute the user command. The AIX login shell does not process the **/etc/environment** file, so Universal Command processes the files in the following order: 1. /etc/environment 2. /etc/profile 3. $HOME/.profile 4. $HOME/.env
Windows	The LOGIN option specifies whether or not the user's profile and environment block are loaded by Universal Command and made available to the user process. For purposes of this discussion, a user's profile are those settings listed in the Windows registry under the **HKEY_CURRENT_USER** key when that user is logged on. This also is known as the user's registry hive. A user's environment block contains the environment variables that are defined for the user when that user is logged on.

IBM i	The LOGIN option specifies whether or not UCMD Server:
	• Executes its initial program, which is specified via the **CRTUSRPRF** and **CHGUSRPRF** commands. This program normally runs only when the user logs in from a terminal, making it an interactive session.
	• Sets the following job attributes to those specified in the target user profile:
	  ◦ ASP group information
	  ◦ Coded character set ID
	  ◦ Country or region ID
	  ◦ Current library
	  ◦ Character identifier control
	  ◦ Home directory
	  ◦ Initial library list
	  ◦ Job accounting code
	  ◦ Language ID
	  ◦ Locale
	  ◦ Output queue name
	  ◦ Output queue priority
	  ◦ Print text
	  ◦ Printer device name
	  ◦ Sort sequence table
	  ◦ Status message handling
	Note
	LOGIN is used only if the USER_SECURITY option is set to **default**. If USER_SECURITY is set to **none**, the initial program inherits the attributes of the invoking job. Thus, setting job attributes would have no consequence. Also, running the initial program from a general profile, if one is specified, may produce unwanted results for some customers. If job customization is desired, the **UCMSJOBI** exit program should be used.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	login *option*	✓	✓	✓	✓
Manager Override	n/a				

Values

option is the specification for whether or not to:

IBM i	Execute the initial program specified via the **CRTUSRPRF** and **CHGUSRPRF** commands and sets the job attributes.
UNIX	Invoke the login shell.
Windows	Load the user's registry hive and environment block.
z/OS	Invoke the login shell.

Valid values for *option* are:

IBM i	• **yes** Execute the initial program. • **no** Do not execute the initial program.
UNIX	• **yes** Invoke the shell as a login shell. • **no** Do not invoke the shell as a login shell.
Windows	• **yes** Load the user's registry hive and environment block. • **no** Do not load the user's registry hive and environment block.
z/OS	• **yes** Invoke the shell as a login shell. • **no** Do not invoke the shell as a login shell.

Default is no.

LOGON_METHOD - UCMD Server configuration option

Description

The LOGON_METHOD option specifies the user's log on method.

If the UCMD Server is configured for user security (see the USER_SECURITY option), the log on method determines how the user is logged onto the Windows system.

If security is inactive, LOGON_METHOD is ignored.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	logon *option*			✓	
Manager Override	n/a				

Values

option is the user's log on method.

Valid values for *option* are:

- **batch**
  Windows log on type is **batch**. A batch log on prevents the command from interacting with the desktop. The user ID logging on as a batch user requires the Windows User Right "Log on as a batch job." If the user does not have this right, the log on action will fail.
- **interactive**
  Windows log on type is **interactive**. An interactive log on permits the command to interact with the desktop. No additional rights are required for a user to log on as an interactive user.

Default is interactive.

MESSAGE_LEVEL - UCMD Server configuration option

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	message_level *level*	✓	✓	✓	✓
Manager Override	n/a				

Values

level indicates either of the following level of messages:

- **trace**
  Writes trace messages used for diagnostic purposes (see Trace Files).

  Note
  ⚠ Use **trace** only as directed by Stonebranch, Inc. Customer Support.
- **audit**
  Writes audit, informational, warning, and error messages.
- **info**
  Writes informational, warning, and error messages.
- **warn**
  Writes warning and error messages.
- **error**
  Writes error messages only.

Default is info.

Trace Files

IBM i	The trace file name is **CURLIB/UNVTRCUCMS(Sxxxyyyyyy)**, where: • **xxx** is the three least-significant hexadecimal digits of the UCMD Server process identifier. • **yyyyyy** is the six least-significant hexadecimal digits of the component identifier. The fault tolerant modes of Universal Command require this combination to produce unique, meaningful member identifiers. The default ***CURLIB** is **UNVTMP520**.
UNIX	The trace file name is **ucmsrv-N-S.trc**, where: • **N** is the component ID assigned to this instance of the UCMD Server by the Broker. • **S** is a sequence number. The trace file is created in the trace directory, as specified by the TRACE_DIRECTORY option, which defaults to **/var/opt/universal/trace**.
Windows	The trace file name is **ucmsrv-N.trc**, where: • **N** is the process ID of UCMD Server. It is created in the working directory of UCMD Server.

z/OS	There are two possible destinations of the trace data: 1. If ddname **UNVTRMDL** is defined in the UBROKER started task procedure, a sequential data set is created using the data set allocated to UNVTRMDL as a model. The dynamically allocated trace data set name is **#HLQ.UCM.Dyymmdd.Thhmmss.Cnnnnnns**, where: - **#HLQ** is the data set name allocated on the UNVTRMDL ddname. - **yymmdd** is the year, month, and day. - **hhmmss** is the hour, minute, second the data set was allocated. - **nnnnnn** is the last six digits of the Server's component ID in hexadecimal format. - **s** is the component ID's sequence number from 0 - F. Each time that a server is restarted, its sequence number is incremented. If a server is restarted more than 15 times, tracing is disabled. 2. If ddname **UNVTRMDL** is not defined in the UBROKER started task procedure, member name **Cnnnnnns** is created in the PDS or PDS/E allocated to the UNVTRACE ddname, where: - **nnnnnn** is the last six digits of the Server's component ID in hexadecimal format. - **s** is the component ID's sequence number from 0 - F. Each time that a server is restarted, its sequence number is incremented. If a server is restarted more than 15 times, tracing is disabled. Depending on the error condition being diagnosed, it is possible that the member name of the **UNVTRACE** PDS or PDS/E is not created. If this occurs, the **UNVTRMDL** ddname must be used to create a sequential data set name. The records written to PDS and PDS/E members cannot be wrapped, so the TRACE_FILE_LINES configuration option limit has no effect on the maximum number of trace records written to the member.

MIN_SSL_PROTOCOL - UCMD Server configuration option

Description

The MIN_SSL_PROTOCOL option specifies the minimum SSL/TLS protocol level that will be negotiated and used for communications channels.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	min_ssl_protocol *option*		✓	✓	✓
Manager Override	n/a				

Note

This option was introduced to IBM i in version 5.1.1.0.

Values

option is the specification for the minimum SSL/TLS protocol level that will be supported.

- **TLS1_0**
  Minimum SSL/TLS protocol is TLS 1.0.
- **TLS1_2**
  Minimum SSL/TLS protocol is TLS 1.2.

Default is TLS1_2.

MSG_SUPPRESSION_LIST - UCMD Server configuration option

Description

The MSG_SUPPRESSION_LIST option specifies a list of message IDs representing Universal messages to be suppressed.

The list consists of zero or more comma-separated Universal message ID numbers. For example:

- 193 - Suppress message UNV0193W only.
- 192,193 - Suppress message UNV0192W and UNV0193W.

Suppressed messages are not printed to logs or output, even if a condition arises that normally would produce the message(s).

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	msg_suppression_list *list*		✓	✓	✓
Manager Override	n/a				

Values

list is the list of message IDs representing Universal messages to be suppressed.

NETWORK_FAULT_TOLERANT - UCMD Server configuration option

Description

The NETWORK_FAULT_TOLERANT option specifies whether or not network fault tolerance should be activated.

Network fault tolerance enables UCMD Manager and UCMD Server to recover from network faults and continue executing without any loss of data.

Network fault tolerance is not negotiated between the UCMD Manager and UCMD Server. The UCMD Server must have this option activated in order for the UCMD Manager and UCMD Server to execute with the network fault tolerant protocol.

If the UCMD Server does have this option activated, this UCMD Manager NETWORK_FAULT_TOLERANT configuration option controls whether or not it is used.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	network_fault_tolerant *options*	✓	✓		✓
Manager Override	n/a				

Values

option is the user's log on method.

Valid values for *options* are:

- **yes**
  Fault tolerance is requested. If the UCMD Manager has fault tolerance activated, the UCMD Manager and UCMD Server will use a fault tolerant network protocol.
- **no**
  Fault tolerance is not requested. The fault tolerant protocol will not be used.

Default is yes.

NLS_DIRECTORY - UCMD Server configuration option

Description

The NLS_DIRECTORY option specifies the name of the directory where the UCMD Server message catalog and code page tables are located.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	nls_directory *directory*		✓	✓	
Manager Override	n/a				

Values

directory is the name of the directory where the catalog and tables are located.

Full path names are recommended.

Relative path names are relative to the **universal** installation directory.

Default

UNIX	Default is **`/opt/universal/nls`**.
Windows	Default is **`..\nls`**.

PRIORITY - UCMD Server configuration option

Description

The PRIORITY option specifies the execution priority of the user job being run.

In the case of a TACL job, a TACL shell is executed with the specified priority and the user job is started with a priority of one less than that of the TACL shell's priority.

If PRIORITY is not specified, the user job is executed at the same priority as the UCMD server process.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	priority *priority*				
Manager Override	-priority *priority*				

Values

priority is the execution priority.

SCRIPT_TYPE - UCMD Server configuration option

Description

The SCRIPT_TYPE option specifies the type of script specified by the UCMD Manager SCRIPT_FILE option.

The UCMD Server will process the SCRIPT_FILE script according to the value of this SCRIPT_TYPE option.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	script_type *option*	✓	✓	✓	✓
Manager Override	-script_type *option*	✓	✓	✓	

Values

option is the type of script file specified by the UCMD Manager SCRIPT_FILE option.

UNIX	**service**: Supports Universal Command Agent for SOA workload.
Windows	Except for **service** (see below), Windows script types are interpreted as file extensions. The script is executed by the program associated with the file extension. If no program association exists for the extension, the script will fail. • **service**: Supports Universal Command Agent for SOA workload.

Default

IBM i	**cmd**
UNIX	**shell**
Windows	**bat**
z/OS	**shell**

SHELL - UCMD Server configuration option

Description

The SHELL option specifies the command shell (UNIX or z/OS UNIX System Services) with which to execute the user shell commands.

The command shell must be specified with an argument that directs it to execute the given command and then exit.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	shell *file*		✓		✓
Manager Override	n/a				

Values

file is the name of the command shell.

Default is "/bin/sh -c".

SPOOL_DIRECTORY - UCMD Server configuration option

Description

The SPOOL_DIRECTORY option specifies the name of the directory where the UCMD Server spool files are located.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	spool_directory *directory*		✓	✓	
Manager Override	n/a				

Values

directory is the name of the UCMD Server spool files directory.

Relative path names are relative to the UCMD Server installation directory. Full path names are recommended.

Default

UNIX	Default is **`/var/opt/universal/spool`**.
Windows	Default is **`C: \Program Files\Universal\spool`**.

STDIN_ALLOC - UCMD Server configuration option

Description

The STDIN_ALLOC option specifies the DCB allocation attributes for the standard input data set allocated for a started task request.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	n/a				
Manager Override	-stdin_alloc *options*				✓

Values

options is a comma-separated list of DCB allocation attributes, in a format similar to JCL parameters.

See the DEFAULT_STDIN_ALLOC option for a list of the attributes that are allowed.

STDIN_ALLOC values are merged with the DEFAULT_STDIN_ALLOC values, which override the STDIN_ALLOC values.

STDIN_HLQ - UCMD Server configuration option

Description

The STDIN_HLQ option specifies the high-level qualifier used for dynamically allocating started task standard input data sets.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	stdin_hlq *hlq*				✓
Manager Override	n/a				

Values

hlq is the high-level qualifier.

Default is requested user ID.

STDIO_TIMEOUT - UCMD Server configuration option

Description

The STDIO_TIMEOUT option specifies the amount of time that the UCMD Server process will wait for standard I/O to be closed by child processes after the parent process has completed.

When this time has expired, the server process will exit.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	stdio_timeout *seconds*	✓	✓	✓	✓
Manager Override	-stdiotimeout *seconds*	✓	✓	✓	✓

Values

seconds is the amount of time (in seconds) that the UCMD Server will wait for standard I/O to be closed.

Note

seconds must be greater than *0* (zero).

Default is 31536000 (1 year).

Manager Override

The format (long and short forms) for the UCMD Manager override of this option is:

- **-server " -stdiotimeout seconds"**
- **-S " -stdiotimeout seconds"**

Note

The first quotation mark must be preceded and followed by a space.

TMP_DIRECTORY - UCMD Server configuration option

Description

The TMP_DIRECTORY option specifies the name of the directory that the UCMD Server uses for temporary files.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	tmp_directory *directory*		✓	✓	✓
Manager Override	n/a				

Values

directory is the name of the directory.

It should specify a fully qualified path name.

Default

UNIX	Default is **`/var/opt/universal/tmp`**.
Windows	Default is **`..\tmp`**.
z/OS	Default is **`/tmp`**.

TRACE_DIRECTORY - UCMD Server configuration option

Description

The TRACE_DIRECTORY option specifies the directory name that the UCMD Server uses for its trace files.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	trace_directory *directory*		✓	✓	
Manager Override	n/a				

Values

directory is the name of the trace file directory.

Relative path names are relative to the UCMD Server installation directory. Full path names are recommended.

Default

Windows	Default is **`C:\Program Files\Universal\UCmdSrv`**.
UNIX	Default is **`/var/opt/universal/trace`**.

TRACE_FILE_LINES - UCMD Server configuration option

Description

The TRACE_FILE_LINES option specifies the maximum number of lines to write to the trace file.

When this maximum number of lines has been reached, the trace file will wrap around and the UCMD Server will start writing trace entries after the trace header lines.

z/OS

Trace file wrapping is supported only with sequential data sets that have a fixed record format. Partitioned data sets or variable record formats are not supported.

Note

A trace file is generated when the MESSAGE_LEVEL option is set to a value of **trace**.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	trace_file_lines *number*	✓	✓	✓	✓
Manager Override	n/a				

Values

number is the maximum number of lines to write to the trace file.

The average size of a trace file line is 50 characters.

Default is the very large value of 5E8E8E800. If space is limited, set this to a smaller value.

TRACE_TABLE - UCMD Server configuration option

Description

The TRACE_TABLE option specifies the size of a wrap-around trace table maintained in memory.

The trace table is written to a file / data set when the program ends under the conditions specified in this option. Tracing is activated, and a trace file is generated, when the MESSAGE_LEVEL option is set to **trace**.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	trace_table *size, condition*	✓	✓	✓	✓
Manager Override	n/a				

Values

size is the size (in bytes) of the table.

The size can be suffixed with either of the following characters:

- **M** indicates that the size is specified in megabytes
- **K** indicates that the size is specified in kilobytes

For example, **50M** indicates that 50 x 1,048,576 bytes of memory is allocated for the trace table.

Note

A value of **0** indicates that the trace table is not used.

condition is the condition under which the trace table is written.

Possible values for *condition* are:

- **error**
  Write the trace table if the program ends with a non-zero exit code.
- **always**
  Write the trace table when the program ends regardless of the exit code.
- **never**
  Never write the trace table.

UCMSCPST_PATH - UCMD Server configuration option

Description

The UCMSCPST_PATH option specifies the absolute path to an external link that resides on the z/OS UNIX file system. The link points to the UCMD Server STC command processor, UCMSCPST.

The option is provided to support disabling the UID 0 requirement for the Universal Broker started task. Specifying this link allows UCMD Server to execute a started task without BPX.SUPERUSER and BPX.DAEMON privileges.

If a value is provided for this option, the external link must be created manually prior to any attempt to run a started task via UCMD Server. The link must point to the UCMSCPST load module and it must be owned by UID 0.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	ucmscpst_path *directory*				✓
Manager Override	n/a				

Values

directory is the fully-qualified path to an external link that resides on the z/OS UNIX file system.

If this option is not specified, UCMD Server will format a link name and attempt to create it in the location specified by the TMP_DIRECTORY UCMD Server configuration option.

If UCMD Server does not have BPX.SUPERUSER privileges, the attempt to create the link will fail.

USE_USER_ACCOUNTING_CODE - UCMD Server configuration option

Description

The USE_USER_ACCOUNTING_CODE option specifies whether or not the IBM i user profile under which a process is run is to be used as the source for the job accounting code.

Note

The LOGIN option also allows switching to the new accounting code and takes precedence over the USE_USER_ACCOUNTING_CODE option.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	use_user_accounting_code *option*	✓			
Manager Override	n/a				

Values

option is the specification for whether or not the IBM i user profile is to be used as the source for the job accounting code.

Valid values for *option* are:

- **yes**
  IBM i user profile is to be used as the source for the job accounting code.
    - If the USER_SECURITY option is set to **default**, the user profile under which the server runs is the user profile specified via the user and password received from the initiating UCMD manager.
    - If the USER_SECURITY option is set to **none**, the server runs under the user profile associated with Universal Broker job. The user profile and accounting code switch occurs in the submitted command initialization phase, **UCMSINIT**.
- **no**
  IBM i user profile is not to be used as the source for the job accounting code.

USER_SECURITY - UCMD Server configuration option

Description

The USER_SECURITY option specifies whether or not to user security and, if so, the security method.

- If user security is activated, the UCMD Server logs the user onto the system, and the command is run with the user's identity.
- If user security is not activated, the command runs with the same identity as the UCMD Server.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	security *option*	✓	✓	✓	✓
Manager Override	n/a				

Values

option is the specification (and method) for activating user security.

IBM i	• DEFAULT User-supplied user ID and password is authenticated against the user profile. • NONE No user security. Note If the UCMD Server runs with this option value, Stonebranch, Inc. highly recommends removing *ALLOBJ authority from the user profile UNVUBR520. Otherwise, all commands will execute with this authority.
UNIX	• DEFAULT Use UNIX default user authentication method, */etc/passwd* or */etc/shadow*. • INHERIT Universal Command Server will inherit the user account of the broker which started it. The user's process is started with the same user ID as that inherited from the Broker. • NONE Universal Command Server will inherit the user account of the broker which started it. The user's process is started with the same user ID as that inherited from the Broker. • PAM Use the Pluggable Authentication Modules (PAM) interface to provide user authentication. • PAM_SESSIONS (Linux only) Processes Pluggable Authentication Modules (PAM) session modules in addition to account and authentication modules. • TRUSTED Use HP Trust Security authentication. WARNING! If PAM_SESSIONS is selected, the system requires that at least one session is configured for PAM. Without a properly configured PAM session module, Universal Command fails to start. Check system logs, including the authentication log, for failure information.
Windows	• DEFAULT User\-supplied user ID and password is authenticated against the user profile. • INHERIT Universal Command Server will inherit the user account of the Broker which started it. The user's process is started with the same user ID as that inherited from the Broker. • NONE Universal Command Server will inherit the user account of the Broker which started it. The user's process is started with the same user ID as that inherited from the Broker.

z/OS	• DEFAULT Use z/OS SAF user authentication method. The user ID must have an OMVS segment. • INHERIT Universal Command Server will inherit the user account of the broker which started it. The user's process is started with the same user ID as that inherited from the Broker. • NONE Universal Command Server will inherit the user account of the broker which started it. The user's process is started with the same user ID as that inherited from the Broker.

Note

For UNIX, Windows, and z/OS: The INHERIT value replaces the NONE value. There currently are no plans to deprecate support for NONE, but new installs should use INHERIT instead of NONE.

Universal Command Component Definition Options

Overview

This page provides links to detailed information about the options that comprise Universal Command (UCMD) component definitions.

The options are listed alphabetically, without regard to any specific operating system.

Component Definition Options Information

For each component definition option, these pages provide the following information.

Description

Describes the option and how it is used.

Usage

Provides a table of the following information:

Method	Syntax	IBM i	UNIX	Windows	z/OS
Component Definition Keyword	<Format / Value>				

Method

Identifies the method used for specifying a Universal Command component definition option:

- Component Definition Keyword

Syntax

Identifies the syntax of the method used to specify the option:

- Format Specific characters that identify the option.
- Value Type of value(s) to be supplied for this method.

(Operating System)

Identifies the operating systems for which the method of specifying the option is valid:

- IBM i
- UNIX
- Windows
- z/OS

Values

Identifies all possible values for the specified value type.

Defaults are identified in **bold type**.

Component Definition Options List

The following table identifies all of the options that can comprise a Universal Command component definition.

Component	Description
AUTOMATICALLY_START	Specification for whether or not the UCMD Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCMD Server.
CONFIGURATION_FILE*	Name of the UCMD Server configuration file.
RUNNING_MAXIMUM	Maximum number of UCMD Servers that can run simultaneously.
START_COMMAND*	Program name of the UCMD Server.
WORKING_DIRECTORY*	Directory used as the working directory of the UCMD Server.
* These options are required in all component definitions.	

AUTOMATICALLY_START - UCMD Component Definition option

Description

The AUTOMATICALLY_START option specifies whether or not the UCMD Server starts automatically when the Universal Broker is started.

Note

AUTOMATICALLY_START is optional in a component definition.

Usage

Method	Parameter / Value	IBM i	UNIX	Windows	z/OS
Component Definition Keyword	auto_start *option*	✓	✓	✓	✓

Values

option is the specification for how the UCMD Server is started.

The only valid value for *option* is:

- **no**
  Universal Command Server is not started automatically when Universal Broker is started. It is started only on demand.

COMPONENT_NAME - UCMD Component Definition option

Description

The COMPONENT_NAME option specifies the name of the UCMD Server.

Component start requests refer to UCMD Server by this name.

Note

COMPONENT_NAME is optional in a component definition. If it is not specified, the file name is used as the component name.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Component Definition Keyword	component_name *name*	✓	✓	✓	✓

Values

name is the name of the Universal Command Server.

There is only one valid value for *name*:

- **ucmd**
  (This is the name of the Universal Command Server component definitions file / member.)

This name should not be changed.

CONFIGURATION_FILE - UCMD Component Definition option

Description

The CONFIGURATION_FILE option specifies the name of the UCMD Server configuration file.

Note

⚠ CONFIGURATION_FILE is required in a component definition.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Component Definition Keyword	configuration_file *member***or** configuration_file *file*	✓	✓	✓	✓

Values

member / *file* is the name of the configuration member / file.

IBM i	Configuration file name can be any valid file name. If the name is non-qualified, library list ***LIBL** is searched. The default file name is **UNVPRD520/UNVCONF(UCMDS)**.
UNIX	Full path name of the configuration file. The file name can be any valid file name. The installation default is **`/etc/universal/ucmds.conf`**.
Windows	Full path name of the configuration file. The file name can be any valid file name. The installation default is **`c:\Documents and Settings\All Users\Application Data\Universal\conf\ucmds.conf`**.
z/OS	Member name of the component configuration file in the **UNVCONF** library allocated to the Universal Broker ddname **UNVCONF**. The installation default is **UCSCFG00**.

RUNNING_MAXIMUM - UCMD Component Definition option

Description

The RUNNING_MAXIMUM option specifies the maximum number of UCMD Servers that can run simultaneously.

If this maximum number is reached, any command received to start a UCMD Server is rejected.

Note

⚠ RUNNING_MAXIMUM is optional in a component definition.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Component Definition Keyword	running_max *maximum*	✓	✓	✓	✓

Values

maximum is the maximum number of UCMD Servers that can run simultaneously.

Default is 100.

Note

⚠ If you specify 0 for *maximum*, the default (100) will be used. To use 0 for the maximum number of servers, specify -1 or less for *maximum*.

START_COMMAND - UCMD Component Definition option

Description

The START_COMMAND option specifies the full path name (member name for z/OS) of the UCMD Server program.

Optionally, START_COMMAND also can specify command line options.

Note

START_COMMAND is required in a component definition.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Component Definition Keyword	start_command *member***or** start_command *name[options]*	✓	✓	✓	✓

Values

member / *name* is the full path name of the UCMD Server program.

options is the optional list of command line options.

IBM i	*name* is the UCMD Server program. If the program name is non-qualified, the library list ***LIBL** is searched.
UNIX	*name* is the full path name of the UCMD Server program.
Windows	*name* is the full path name of the UCMD Server program. This name is defined at installation; it is not modifiable from the Universal Configuration Manager.
z/OS	*member* is the program object of the UCMD Server. The program object must be in the Universal Broker's search order for loading program objects. The default location is the SUNVLOAD library allocated to the Universal Broker's STEPLIB ddname. Alternatively, starting with Universal Broker 5.1.0.4, *member* can be the fully specified path of a USS external link to the UCMD Server program.

options is not a valid value.

WORKING_DIRECTORY - UCMD Component Definition option

Description

The WORKING_DIRECTORY option specifies the full path name used as the working directory of UCMD Server.

Note

⚠ WORKING_DIRECTORY is required in a component definition.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Component Definition Keyword	working_directory *directory*	✓	✓	✓	✓

Values

directory is the full path name of the working directory.

IBM i	working_directory serves as a required placeholder only.
UNIX	*directory* is the full path name of the directory Universal Command Server uses as its working directory.
Windows	*directory* is the full path name of the directory Universal Command Server uses as its working directory.
z/OS	*directory* is the HFS directory name that the Universal Command Server uses as its working directory.

Default = (.).

Note

⚠ Do not change this directory.

Universal Command UACL Entries

Universal Command UACL Entries

This page provides links to detailed information on the Universal Access Control List (UACL) entries available for use with Universal Command.

The UACL entries are listed alphabetically, without regard to any specific operating system.

UACL Entries Information

For each UACL entry, the following information is provided.

Description

Describes the UACL entry and how it is used.

Usage

Provides a table of the following information:

Method	Syntax	IBM i	UNIX	Windows	z/OS
UACL File Keyword	<Type / Rule>				

Method

Identifies the method used for specifying a UACL entry:

- UACL File Keyword

Syntax

Identifies the syntax of the method used for a UACL entry:

- Type: Universal Agent component to which the rule applies.
- Rule: Client's identity, request to which the entry pertains, and security attributes that the entry enforces.

(Operating System)

Identifies the operating systems for which the method of specifying the UACL entry is valid:

- IBM i
- UNIX
- Windows
- z/OS

Values

Identifies all possible values for the fields in a UACL entry rule.

Defaults are identified in **bold type**.

UACL Entries List

The following table identifies all Universal Command UACL Entries.

UACL Entry	Description
UCMD_ACCESS	Allows or denies access to Universal Command Server services. There are two forms to this entry: • **ucmd_access** • **ucmd_cert_access**
UCMD_REQUEST	Allows or denies access to Universal Command Server services based on client identification and request type.

UCMD_ACCESS - UCMD UACL Entry

Description

A UCMD_ACCESS UACL entry either allows or denies access to Universal Command Server services.

If access is permitted, UCMD_ACCESS also specifies whether or not user authentication is required.

There are two forms of the UCMD_ACCESS entry, based on the client identification method:

- **ucmd_access** is for IP-based client identification.
- **ucmd_cert_access** is for X.509 certificate-based client identification.

A **ucmd_access** UACL entry is matched if all of the following occur:

- Request comes from an IP address identified by *host*.
- Remote end is executing as user *remote_user*.
- Remote user is requesting to execute a command as local user *local_user*.

A **ucmd_cert_access** UACL entry is matched if both of the following occur:

- Request comes from a client with a certificate identifier of *certid*.
- Remote user is requesting to execute a command as local user *local_user*.

The first matching rule is used to control access.

See UACL Entries for details on *host*, *remote_user*, *local_user*, and *certid* specification syntax.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
UACL File Keyword	ucmd_access *host,remote_user,local_user,access,auth*	✓	✓	✓	✓
UACL File Keyword	ucmd_cert_access *certid,local_user,access,auth*	✓	✓	✓	✓

Values

Valid values for *access* are:

- **deny**
  Service is denied. A message is returned to the remote end. The connection is closed.
- **allow**
  Service is accepted and processed.

Valid values for *auth* are:

- **auth**
  Local user account must be authenticated. The Manager must provide a proper password for the account.
- **noauth**
  Local user account does not require user authentication. The Manager still must supply a password to satisfy command syntax rules, but it will not be verified. Any password value will suffice.

  Note

  noauth should be used with care. Turning off user authentication may violate your local security policies on the Server system.

  Windows

  noauth is not a valid value.

UCMD_REQUEST - UCMD UACL Entry

Description

A UCMD_REQUEST UACL entry allows or denies access to Universal Command Server services based on client identification and request type.

If access is permitted, the UCMD_REQUEST also specifies whether or not user authentication is required.

There are two forms of the UCMD_REQUEST entry based on the client identification method:

- **ucmd_request** form is for IP-based client identification.
- **ucmd_cert_request** is for X.509 certificate-based client identification.

A **ucmd_request** UACL entry is matched if all of the following occur:

- Request comes from an IP address identified by *host*.
- Remote end is executing as user *remote_user*.
- Remote user is requesting to execute a command as local user *local_user*.

A **ucmd_cert_request** UACL entry is matched if both of the following occur:

- Request comes from a client with a certificate identifier of **certid**.
- Remote user is requesting to execute a command as local user **local_user**.

The first matching rule is used to control access.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
UACL File Keyword	ucmd_request *host,remote_user,local_user,req_type,req_name,access,auth*	✓	✓	✓	✓
UACL File Keyword	ucmd_cert_request *certid,local_user,req_type,req_name,access,auth*	✓	✓	✓	✓

Values

req_type

req_type specifies the type of request that the Universal Command Manager is requesting.

Valid values for *req_type* are:

Value	Description	IBM i	Windows	UNIX	z/OS
cmd	IBM i command (may be user-defined).	✓			
rexx	Single line consisting entirely of REXX statements (maximum 1000 bytes).	✓			
cmdref	Request is for the execution of a command reference. • *req_name* is the command reference name, which is case insensitive. UNIX *req_name* is case sensitive.	✓	✓	✓	✓
shell	Request is for the execution of a shell command or shell script: • For a shell command, *req_name* is the name of the command. • For a shell script, *req_name* is empty.		✓	✓	✓
stc	Request is for the execution of a started task. • *req_name* is the started task command, which is case insensitive.				✓

req_name

req_name further qualifies the request by its specified name.

Valid values for *req_name* depend on the *req_type* value.

req_name includes any options provided by the Universal Command Manager.

For Example

```
ucmd -c "cmd3 o1 o2" -cmd_type cmdref ...
```

This command will not match the following UACL entry:

ucmd_request ALL,*,*,cmdref,cmd3,allow,auth

Since the options **o1** and **o2** are part of the request name field, the UACL entry must be written as:

ucmd_request "ALL,*,*,cmdref,cmd3*,allow,auth"

access

Valid values for *access* are:

- **deny**
  Service is denied. A message is returned to the remote end. The connection is closed.
- **allow**
  Service is accepted and processed.

auth

Valid values for *auth* are:

- **auth**
  Local user account must be authenticated. The Manager must provide a proper user ID and password.
- **noauth**
  Local user account does not require user authentication. The Manager still must supply a password to satisfy command syntax rules, but it will not be verified. Any password value will suffice.

  Note

  noauth should be used with care. Turning off user authentication may violate your local security policies on the Server system.

  Windows

  noauth is not a valid value.

See UACL Entries for details on *host*, *remote_user*, *local_user*, and *certid* specification syntax.

Universal Command - Command References

Overview

A command reference is a file residing on the system Universal Command (UCMD) Server installed that contains a pre-defined command or script to be executed at the request of a Universal Command Manager.

The UCMD Manager requests execution of a command reference by specifying:

- The name of the command reference with the COMMAND option.
- The command type as a command reference with the COMMAND_TYPE option.

The UCMD Server finds the command references in the configured command reference location, as specified in the UCMD Server CMD_REFERENCE_DIRECTORY option. If the command reference is found, the UCMD Server executes the command or script defined in the command reference.

The UCMD Manager does not provide its own command or script to execute. It only provides the name of a command reference. This provides the ability to precisely control what UCMD Managers can execute on the server. The UCMD Manager does not have the ability to specify the commands executed on the system. It can only reference a pre-defined command reference that has been defined by the UCMD administrator for the system.

A command reference also can specify that the command or script accepts options or command line arguments from the UCMD Manager.

Command Reference Sections

A command reference is a sequential file containing two sections:

1. Options Section
2. Command Section

The sections are separate with a logical end-of-file marker. The logical end-of-file market is the string **<eof>**.

Options Section

The Options section contains #Command Reference Options that define the command reference and specify operational characteristics. It also can contain comments.

Syntax Rules

Syntax rules for the Options section are:

1	Lines starting with a hash character (#) are comments.
2	Blank lines are ignored.
3	Options are specified with a Name followed by a Value.
4	Names are prefixed with a hyphen (-).
5	Names are case-insensitive.
6	Values are case-insensitive.
7	One or more spaces must separate the Name and Value.

8	Options section is ended with <eof> at the start of a line.

Command Reference Options

The following table provides a description and values for all command reference options.

Name	Description	Values
-format	Format of the Command Section. This option is required.	Valid values are: - **cmd** Command section contains a command. - **script** Command section contains a script. The script is defined in the same format as if it was being defined normally in a self-contained file.
-type	Type of command or script in the Command Section. This option is not required.	Values depend on the UCMD Server operating system. See the UCMD Server COMMAND_TYPE andSCRIPT_TYPE options for description of valid values.
-allow_options	Specification for whether or not the command or script accepts user-provided options or command line arguments from the UCMD Manager. This option is not required.	Valid values are: - **yes** Options are passed. - For commands, the UCMD Manager options are concatenated to the end of the command. - For scripts, the UCMD Manager options are passed to the script as command line arguments. - **no** Options are not passed. This is the default.

Command Section

The Command section contains the command or script (as specified by the **-format** command reference option) to be executed.

Commands

Commands are single commands that are executed based on the **-type** option.

The command may span multiple lines. No line continuation characters are required. All lines will be concatenated together to form one command line.

Scripts

Scripts are read verbatim and placed into a temporary file. The temporary file is passed to the script processor for the appropriate script type, as specified by the **-type** option.

Command Reference Example

The following figure is an example command reference that executes a Windows DIR command. The two sections are separated with a logical end-of-file marker, **<eof>**.

```
# Comments describing the command reference.
#
# Execute the DIR command.
#

-format         cmd
-type           shell
-allow_options  no

<eof>

DIR
```

Command Reference Request

A UCMD Manager requests the execution of a command reference just as it does for other command types: via the COMMAND_TYPE option. A COMMAND_TYPE value of `cmdref` specifies that the COMMAND option value is referring to a command reference.

Examples

The following examples illustrate how a UCMD Manager for UNIX would invoke a command reference.

Executing a command reference with no options

The following example executes a command reference named `cmd100`. The `-cmd` option specifies the name of the command reference file on the remote system and the `-cmd_type` option specifies that `-cmd` is referring to a command reference.

```
ucmd -cmd cmd100 -cmd_type cmdref ...
```

Executing a command reference with options

The following example executes a command reference named `cmd100`. The `-cmd` option specifies the name of the command reference file on the remote system and the `-cmd_type` option specifies that `-cmd` is referring to a command reference. Two command line options are passed to the command reference, `opt1` and `opt2`.

```
ucmd -cmd "cmd100 opt1 opt2" -cmd_type cmdref ...
```

The actual format of the options, `opt1` and `opt2`, depend on the command or script being executed in the command reference. In some cases, the options may be required to be comma separated and, other cases, space separated.

Command Line Options

The command line options used in these examples are:

Option	Description
-cmd	Command to execute on the remote system.
-cmd_type	Type of command specified by -cmd.

UACL Rules

Command references can be protected with Universal Access Control List (UACL) rules. Access can be denied or allowed based on the command reference name (see Universal Access Control List for details).

Universal Command Additional Information

Universal Command Additional Information

The following table identifies and provides links to additional information related to Universal Command.

Information	Description
SSL/TLS Cipher Suites	SSL/TLS cipher suites for use with Universal Command.
Character Code Pages	Character Code pages for use with Universal Command.
UTT Files	Universal Translate Table (UTT) files are used to translate between Unicode and the local single-byte code page.

Character Code Pages - UCMD

The following table identifies the character code pages provided by Stonebranch Inc. for use with Universal Agent on each supported operating system.

Code Page	CCSID	z /OS	UNIX	Windows	IBM i / HFS	IBM i / LIB	HP NonStop
IBM037	037	✓			✓	✓	
IBM273	273	✓			✓	✓	
IBM277	277	✓			✓	✓	
IBM278	278	✓			✓	✓	
IBM280	280	✓			✓	✓	
IBM284	284	✓			✓	✓	
IBM500	500	✓			✓	✓	
IBM875	875	✓					
IBM1025		✓					
IBM1047		✓			✓	✓	
IBM1140	1140	✓			✓	✓	
IBM1141	1141	✓			✓	✓	
IBM1142	1142	✓			✓	✓	
IBM1143	1143	✓			✓	✓	
IBM1144	1144	✓			✓	✓	
IBM1145	1145	✓			✓	✓	
IBM1146	1146	✓			✓	✓	
IBM1147	1147	✓			✓	✓	
IBM1148	1148	✓			✓	✓	
IBM4971	4971	✓					
ISO8859-1	819		✓	✓	✓		✓
ISO8859-2	912		✓	✓	✓		✓
ISO8859-3	913		✓	✓	✓		✓
ISO8859-4	914		✓	✓	✓		✓
ISO8859-5	915		✓	✓	✓		✓
ISO8859-6	1089		✓	✓	✓		✓
ISO8859-7	813		✓	✓	✓		✓
ISO8859-8	916		✓	✓	✓		✓
ISO8859-9	920		✓	✓	✓		✓
ISO8859-10			✓	✓	✓		✓
ISO8859-13	921		✓	✓	✓		✓
ISO8859-14			✓	✓	✓		✓
ISO8859-15	923		✓	✓	✓		✓
PC437	437			✓	✓		
PC737	737			✓	✓		
PC775	775			✓	✓		
PC850	850			✓	✓		
PC852	852			✓	✓		
PC855	855			✓	✓		
PC857	857			✓	✓		
PC860	860			✓	✓		

PC861	861			✓	✓		
PC862	862			✓	✓		
PC863	863			✓	✓		
PC864	864			✓	✓		
PC865	865			✓	✓		
PC866	866			✓	✓		
PC869	869			✓	✓		
PC874	874			✓	✓		
WIN1250	1250			✓	✓		
WIN1251	1251			✓	✓		
WIN1252	1252			✓	✓		
WIN1253	1253			✓	✓		
WIN1254	1254			✓	✓		
WIN1255	1255			✓	✓		
WIN1256	1256			✓	✓		
WIN1257	1257			✓	✓		
WIN1258	1258			✓	✓		

SSL/TLS Cipher Suites - UCMD

SSL/TLS Cipher Suites

The following table identifies all SSL/TLS cipher suites provided by Stonebranch, Inc. for use with Universal Command.

Cipher Suite Name	Description
AES256-GCM-SHA384	256-bit AES encryption in Galois Counter Mode, SHA-2 384-bit message digest.
AES256-SHA	256-bit AES encryption with SHA-1 message digest.
AES128-GCM-SHA256	128-bit AES encryption in Galois Counter Mode, SHA-2 256-bit message digest.
AES128-SHA	128-bit AES encryption with SHA-1 message digest.
RC4-SHA	128-bit RC4 encryption with SHA-1 message digest.
RC4-MD5	128-bit RC4 encryption with MD5 message digest.
DES-CBC3-SHA	128-bit Triple-DES encryption with SHA-1 message digest.
DES-CBC-SHA	128-bit DES encryption with SHA-1 message digest. Note As of Universal Agent 6.7.0.0, DES-CBC-SHA is supported only on HP-UX. Additionally, any Agents on HP-UX that accept connections from, or attempt connections to, Agents on other platforms must be configured with at least one currently supported cipher suite besides DES-CBC-SHA. Therefore, those HP-UX Agents cannot be configured only with DES-CBC-SHA in their list of cipher suites.
NULL-SHA256	No encryption and SHA-2 256-bit message digest.
NULL-SHA	No encryption and SHA-1 message digest.
NULL-MD5	No encryption and MD5 message digest.
NULL-NULL	No encryption, no data authentication, SSL/TLS is not used; instead, Universal V2 Protocol (**UNVv2**) is used.

UTT Files - UCMD

UTT Files

The following table identifies the Universal Translate Table (UTT) files that are used to translate between Unicode and the local single-byte code page.

Operating System	UTT File Location
IBM i	UTT files are located in the source physical file **UNVPRD510/UNVNLS**. *codepage* is the member name of the UTT file.
z/OS	UTT files are located in the library allocated to the **UNVNLS** ddname. *codepage* is the member name of the UTT file.
UNIX	UTT files are located in the **nls** subdirectory of the installation directory. *codepage* is the base file name of the UTT file. All UTT files end with an extension of **.utt**.
Windows	UTT files are located in the NLS subdirectory of the installation directory. *codepage* is the base file name of the UTT file. All UTT files end with an extension of **.utt**.
HP NonStop	UTT files are located in the **$SYSTEM.UNVNLS** subvolume. *codepage* is the base file name of the UTT file.