



stonebranch

Universal Agent Utilities 7.1.x

Reference Guide

© 2021 by Stonebranch, Inc. All Rights Reserved.

1. Universal Agent Utilities 7.1.x Reference Guide	7
1.1 Universal Agent Utilities Configuration	8
1.2 Universal Certificate	10
1.2.1 Universal Certificate Usage	11
1.2.2 Universal Certificate for zOS	13
1.2.3 Universal Certificate for UNIX and Windows	16
1.2.4 Universal Certificate Configuration Options	18
1.2.4.1 CA - Universal Certificate configuration option	20
1.2.4.2 CA_CERT_FILE - Universal Certificate configuration option	21
1.2.4.3 CA_CERT_FORMAT - Universal Certificate configuration option	22
1.2.4.4 CERT_DB - Universal Certificate configuration option	23
1.2.4.5 CERT_FILE - Universal Certificate configuration option	24
1.2.4.6 CERT_FORMAT - Universal Certificate configuration option	25
1.2.4.7 CERTPBE - Universal Certificate configuration option	26
1.2.4.8 CODE_PAGE - Universal Certificate configuration option	27
1.2.4.9 COMMAND_FILE_ENCRYPTED - Universal Certificate configuration option	28
1.2.4.10 COMMAND_FILE_PLAIN - Universal Certificate configuration option	29
1.2.4.11 COMMON_NAME - Universal Certificate configuration option	30
1.2.4.12 COUNTRY - Universal Certificate configuration option	31
1.2.4.13 CREATE - Universal Certificate configuration option	32
1.2.4.14 CRL_FILE - Universal Certificate configuration option	33
1.2.4.15 CRL_FORMAT - Universal Certificate configuration option	34
1.2.4.16 DNS_NAME - Universal Certificate configuration option	35
1.2.4.17 EMAIL_ADDRESS - Universal Certificate configuration option	36
1.2.4.18 ENCRYPTION_KEY - Universal Certificate configuration option	37
1.2.4.19 HELP - Universal Certificate configuration option	38
1.2.4.20 IP_ADDRESS - Universal Certificate configuration option	39
1.2.4.21 KEY_SIZE - Universal Certificate configuration option	40
1.2.4.22 KEYPBE - Universal Certificate configuration option	41
1.2.4.23 KEYSTORE_PATH - Universal Certificate configuration option	42
1.2.4.24 LOCALITY - Universal Certificate configuration option	43
1.2.4.25 MESSAGE_LEVEL - Universal Certificate configuration option	44
1.2.4.26 NEXT_UPDATE_DAYS - Universal Certificate configuration option	45
1.2.4.27 NEXT_UPDATE_HOURS - Universal Certificate configuration option	46
1.2.4.28 NLS_DIRECTORY - Universal Certificate configuration option	47
1.2.4.29 NOT_AFTER_DATE - Universal Certificate configuration option	48
1.2.4.30 NOT_BEFORE_DATE - Universal Certificate configuration option	49
1.2.4.31 ORGANIZATION - Universal Certificate configuration option	50
1.2.4.32 ORGANIZATIONAL_UNIT - Universal Certificate configuration option	51
1.2.4.33 PRINT - Universal Certificate configuration option	52
1.2.4.34 PRIVATE_KEY_FILE - Universal Certificate configuration option	53
1.2.4.35 PRIVATE_KEY_FORMAT - Universal Certificate configuration option	54
1.2.4.36 PRIVATE_KEY_INFFILE - Universal Certificate configuration option	55
1.2.4.37 PRIVATE_KEY_PWD - Universal Certificate configuration option	56
1.2.4.38 REQUEST_FILE - Universal Certificate configuration option	57
1.2.4.39 REQUEST_FORMAT - Universal Certificate configuration option	58
1.2.4.40 REVOKE - Universal Certificate configuration option	59
1.2.4.41 REVOKE_REASON - Universal Certificate configuration option	60
1.2.4.42 SERIAL_NUMBER - Universal Certificate configuration option	61
1.2.4.43 SIGNATURE_ALGORITHM - Universal Certificate configuration option	62
1.2.4.44 STATE - Universal Certificate configuration option	63
1.2.4.45 TRANSPORT_FILE - Universal Certificate configuration option	64
1.2.4.46 TRANSPORT_FILE_PWD - Universal Certificate configuration option	65
1.2.4.47 VERIFY - Universal Certificate configuration option	66
1.2.4.48 VERSION - Universal Certificate configuration option	67
1.3 Universal Control	68
1.3.1 Universal Control Manager for zOS	69
1.3.2 Universal Control Manager for Windows	74
1.3.3 Universal Control Manager for UNIX	78
1.3.4 Universal Control Manager for IBM i	82
1.3.5 Universal Control Manager Configuration Options	86
1.3.5.1 ACTIVITY_MONITORING - UCTL Manager configuration option	88
1.3.5.2 BIF_DIRECTORY - UCTL Manager configuration option	89
1.3.5.3 CA_CERTIFICATES - UCTL Manager configuration option	90
1.3.5.4 CERTIFICATE - UCTL Manager configuration option	91
1.3.5.5 CERTIFICATE_EXPIRATION_NOTICE - UCTL Manager configuration option	92
1.3.5.6 CERTIFICATE_REVOCATION_LIST - UCTL Manager configuration option	93
1.3.5.7 CODE_PAGE - UCTL Manager configuration option	94
1.3.5.8 COMMAND_FILE_ENCRYPTED - UCTL Manager configuration option	95
1.3.5.9 COMMAND_FILE_PLAIN - UCTL Manager configuration option	96
1.3.5.10 COMMAND_ID - UCTL Manager configuration option	97
1.3.5.11 CTL_SSL_CIPHER_LIST - UCTL Manager configuration option	98
1.3.5.12 ENCRYPTION_KEY - UCTL Manager configuration option	99
1.3.5.13 EVENT_GENERATION - UCTL Manager configuration option	100
1.3.5.14 HELP - UCTL Manager configuration option	101
1.3.5.15 HOSTNAME_RETRY_COUNT - UCTL Manager configuration option	102
1.3.5.16 INSTALLATION_DIRECTORY - UCTL Manager configuration option	103
1.3.5.17 MESSAGE_LANGUAGE - UCTL Manager configuration option	104
1.3.5.18 MESSAGE_LEVEL - UCTL Manager configuration option	105

1.3.5.19	MIN_SSL_PROTOCOL - UCTL Manager configuration option	106
1.3.5.20	MSG_SUPPRESSION_LIST - UCTL Manager configuration option	107
1.3.5.21	NETWORK_DELAY - UCTL Manager configuration option	108
1.3.5.22	NLS_DIRECTORY - UCTL Manager configuration option	109
1.3.5.23	OUTBOUND_IP - UCTL Manager configuration option	110
1.3.5.24	PLF_DIRECTORY - UCTL Manager configuration option	111
1.3.5.25	PRIVATE_KEY - UCTL Manager configuration option	112
1.3.5.26	PRIVATE_KEY_PWD - UCTL Manager configuration option	113
1.3.5.27	REFRESH_CMD - UCTL Manager configuration option	114
1.3.5.28	REMOTE_HOST - UCTL Manager configuration option	115
1.3.5.29	REMOTE_PORT - UCTL Manager configuration option	116
1.3.5.30	SAF_KEY_RING - UCTL Manager configuration option	117
1.3.5.31	SAF_KEY_RING_LABEL - UCTL Manager configuration option	118
1.3.5.32	SSL_IMPLEMENTATION - UCTL Manager configuration option	119
1.3.5.33	START_CMD - UCTL Manager configuration option	120
1.3.5.34	STOP_CMD - UCTL Manager configuration option	121
1.3.5.35	SYSTEM_ID - UCTL Manager configuration option	122
1.3.5.36	USER_ID - UCTL Manager configuration option	123
1.3.5.37	USER_PASSWORD - UCTL Manager configuration option	124
1.3.5.38	VERIFY_HOST_NAME - UCTL Manager configuration option	125
1.3.5.39	VERIFY_SERIAL_NUMBER - UCTL Manager configuration option	126
1.3.5.40	VERSION - UCTL Manager configuration option	127
1.3.6	Universal Control Server for zOS	128
1.3.6.1	Universal Control Server for zOS - Component Definition	129
1.3.6.2	Universal Control Server for zOS - Configuration	130
1.3.6.3	Universal Control Server for zOS - Universal Access Control List	131
1.3.7	Universal Control Server for Windows	132
1.3.7.1	Universal Control Server for Windows - Component Definition	133
1.3.7.2	Universal Control Server for Windows - Configuration	135
1.3.7.3	Universal Control Server for Windows - Universal Access Control List	136
1.3.8	Universal Control Server for UNIX	137
1.3.8.1	Universal Control Server for UNIX - Component Definition	138
1.3.8.2	Universal Control Server for UNIX - Configuration	139
1.3.8.3	Universal Control Server for UNIX - Universal Access Control List	140
1.3.9	Universal Control Server for IBM i	141
1.3.9.1	Universal Control Server for IBM i - Component Definition	142
1.3.9.2	Universal Control Server for IBM i - Configuration	143
1.3.9.3	Universal Control Server for IBM i - Universal Access Control List	144
1.3.10	Universal Control Server Configuration Options	145
1.3.10.1	ACTIVITY_MONITORING - UCTL Server configuration option	146
1.3.10.2	CODE_PAGE - UCTL Server configuration option	147
1.3.10.3	EVENT_GENERATION - UCTL Server configuration option	148
1.3.10.4	INSTALLATION_DIRECTORY - UCTL Server configuration option	149
1.3.10.5	LOGON_METHOD - UCTL Server configuration option	150
1.3.10.6	MESSAGE_LEVEL - UCTL Server configuration option	151
1.3.10.7	MSG_SUPPRESSION_LIST - UCTL Server configuration option	152
1.3.10.8	NLS_DIRECTORY - UCTL Server configuration option	153
1.3.10.9	TMP_DIRECTORY - UCTL Server configuration option	154
1.3.10.10	TRACE_DIRECTORY - UCTL Server configuration option	155
1.3.10.11	USER_SECURITY - UCTL Server configuration option	156
1.3.11	Universal Control Component Definition Options	157
1.3.11.1	AUTOMATICALLY_START - UCTL Component Definition option	159
1.3.11.2	COMPONENT_NAME - UCTL Component Definition option	160
1.3.11.3	CONFIGURATION_FILE - UCTL Component Definition option	161
1.3.11.4	RUNNING_MAXIMUM - UCTL Component Definition option	162
1.3.11.5	START_COMMAND - UCTL Component Definition option	163
1.3.11.6	WORKING_DIRECTORY - UCTL Component Definition option	164
1.3.12	Universal Control UACL Entries	165
1.3.12.1	UCTL_ACCESS - UCTL UACL entry	167
1.3.12.2	UCTL_REQUEST - UCTL UACL entry	168
1.4	Universal Copy	170
1.4.1	Universal Copy for Windows and UNIX	171
1.4.2	Universal Copy for IBM i	173
1.4.3	Universal Copy Configuration Options	175
1.4.3.1	CPY_MODE - UCOPY configuration option	176
1.4.3.2	FRMFILE - UCOPY configuration option	177
1.4.3.3	FRMFILES - UCOPY configuration option	178
1.4.3.4	FRMMBR - UCOPY configuration option	179
1.4.3.5	HELP - UCOPY configuration option	180
1.4.3.6	INPUTMODE - UCOPY configuration option	181
1.4.3.7	MESSAGE_LEVEL - UCOPY configuration option	182
1.4.3.8	MODE - UCOPY configuration option	183
1.4.3.9	OUTPUT - UCOPY configuration option	184
1.4.3.10	OUTPUTMODE - UCOPY configuration option	185
1.4.3.11	REPLACE - UCOPY configuration option	186
1.4.3.12	TOFILE - UCOPY configuration option	187
1.4.3.13	TOMBR - UCOPY configuration option	188
1.4.3.14	TRANSACTIONAL - UCOPY configuration option	189
1.4.3.15	VERSION - UCOPY configuration option	190

1.5 Universal Database Dump	191
1.5.1 Universal Database Dump for zOS	192
1.5.2 Universal Database Dump for Windows and UNIX	194
1.5.3 Universal Database Dump Configuration Options	195
1.5.3.1 DATABASE_FILE - UDBDUMP configuration option	196
1.5.3.2 DUMP_OPTIONS - UDBDUMP configuration option	197
1.6 Universal Database Load	198
1.6.1 Universal Database Load for zOS	199
1.6.2 Universal Database Load for Windows and UNIX	201
1.6.3 Universal Database Load Configuration Options	202
1.6.3.1 DATABASE_FILE - UDBLOAD configuration option	203
1.6.3.2 OVERWRITE - UDBLOAD configuration option	204
1.7 Universal Display Log File	205
1.7.1 Universal Display Log File Configuration Options	206
1.7.1.1 PRIMARY_FILE - UDSPLOGF configuration option	207
1.7.1.2 PRIMARY_MEMBER - UDSPLOGF configuration option	208
1.7.1.3 REMOVE_MEMBERS - UDSPLOGF configuration option	209
1.7.1.4 SECONDARY_FILE - UDSPLOGF configuration option	210
1.7.1.5 SECONDARY_MEMBER - UDSPLOGF configuration option	211
1.8 Universal Encrypt	212
1.8.1 Universal Encrypt for zOS	213
1.8.2 Universal Encrypt for Windows and UNIX	215
1.8.3 Universal Encrypt for IBM i	216
1.8.4 Universal Encrypt Configuration Options	217
1.8.4.1 AES - UENCRYPT configuration option	218
1.8.4.2 BIF_DIRECTORY - UENCRYPT configuration option	219
1.8.4.3 CODE_PAGE - UENCRYPT configuration option	220
1.8.4.4 ENCRYPTION_KEY - UENCRYPT configuration option	221
1.8.4.5 GENERATE_KEY - UENCRYPT configuration option	222
1.8.4.6 HELP - UENCRYPT configuration option	223
1.8.4.7 INPUT_FILE - UENCRYPT configuration option	224
1.8.4.8 KEYSTORE_PATH - UENCRYPT configuration option	225
1.8.4.9 NLS_DIRECTORY - UENCRYPT configuration option	226
1.8.4.10 OUTPUT_FILE - UENCRYPT configuration option	227
1.8.4.11 STORE_KEY - UENCRYPT configuration option	228
1.8.4.12 VERSION - UENCRYPT configuration option	229
1.9 Universal Event Log Dump	230
1.9.1 Universal Event Log Dump - Configuration	231
1.9.2 Universal Event Log Dump - Command Line Syntax	233
1.9.3 Universal Event Log Dump Configuration Options	234
1.9.3.1 BACKUP_LOG - UELD configuration option	235
1.9.3.2 CLEAR_LOG - UELD configuration option	236
1.9.3.3 END_TIME - UELD configuration option	237
1.9.3.4 HELP - UELD configuration option	238
1.9.3.5 INSTALLATION_DIRECTORY - UELD configuration option	239
1.9.3.6 LOG_DIRECTORY - UELD configuration option	240
1.9.3.7 LOG_TYPE - UELD configuration option	241
1.9.3.8 MESSAGE_DESTINATION - UELD configuration option	242
1.9.3.9 MESSAGE_LANGUAGE - UELD configuration option	243
1.9.3.10 MESSAGE_LEVEL - UELD configuration option	244
1.9.3.11 NLS_DIRECTORY - UELD configuration option	245
1.9.3.12 OUTPUT_FILE - UELD configuration option	246
1.9.3.13 PAGE_HEADERS - UELD configuration option	247
1.9.3.14 PAGE_LENGTH - UELD configuration option	248
1.9.3.15 REMOTE_SERVER - UELD configuration option	249
1.9.3.16 START_TIME - UELD configuration option	250
1.9.3.17 VERSION - UELD configuration option	251
1.10 Universal FTP Client	252
1.10.1 Universal FTP Client Commands	253
1.10.1.1 DELETE - UFTP command	255
1.10.1.2 GET - UFTP command	256
1.10.1.3 LIST - UFTP command	257
1.10.1.4 MDELETE - UFTP command	258
1.10.1.5 MGET - UFTP command	259
1.10.1.6 MKDIR - UFTP command	260
1.10.1.7 MPUT - UFTP command	261
1.10.1.8 PUT - UFTP command	262
1.10.1.9 RMDIR - UFTP command	263
1.10.2 Command Examples	264
1.10.3 Specifying Multiple Files in Commands	266
1.10.4 Universal FTP Client Configuration Options	267
1.10.4.1 ACTIVE_MODE - UFTP configuration option	269
1.10.4.2 AUTHENTICATE_PEER - UFTP configuration option	270
1.10.4.3 BIF_DIRECTORY - UFTP configuration option	271
1.10.4.4 CA_CERTIFICATES - UFTP configuration option	272
1.10.4.5 CERTIFICATE - UFTP configuration option	273
1.10.4.6 CERTIFICATE_EXPIRATION_NOTICE - UFTP configuration option	274
1.10.4.7 CERTIFICATE_REVOCATION_LIST - UFTP configuration option	275
1.10.4.8 COMMAND - UFTP configuration option	276

1.10.4.9	COMMAND_FILE - UFTP configuration option	279
1.10.4.10	DATA_TYPE - UFTP configuration option	280
1.10.4.11	DESTINATION - UFTP configuration option	281
1.10.4.12	ENABLE_SSL - UFTP configuration option	282
1.10.4.13	ENCRYPTED_COMMAND_FILE - UFTP configuration option	283
1.10.4.14	ENCRYPTION_KEY - UFTP configuration option	284
1.10.4.15	EXTENDED_PASSIVE_MODE - UFTP configuration option	285
1.10.4.16	HELP - UFTP configuration option	286
1.10.4.17	HOST - UFTP configuration option	287
1.10.4.18	KEY_PASSPHRASE - UFTP configuration option	289
1.10.4.19	LIST_PARSER - UFTP configuration option	290
1.10.4.20	MESSAGE_LEVEL - UFTP configuration option	291
1.10.4.21	MIN_SSL_PROTOCOL - UFTP configuration option	292
1.10.4.22	MOVE - UFTP configuration option	293
1.10.4.23	NO_SIZE - UFTP configuration option	294
1.10.4.24	PASSIVE_MODE - UFTP configuration option	295
1.10.4.25	PASSWORD - UFTP configuration option	296
1.10.4.26	PLF_DIRECTORY - UFTP configuration option	297
1.10.4.27	PORT - UFTP configuration option	298
1.10.4.28	PRIVATE_KEY - UFTP configuration option	299
1.10.4.29	PRIVATE_KEY_PWD - UFTP configuration option	300
1.10.4.30	PROTOCOL - UFTP configuration option	301
1.10.4.31	QUOTE - UFTP configuration option	302
1.10.4.32	REGEX - UFTP configuration option	304
1.10.4.33	SITE - UFTP configuration option	305
1.10.4.34	SOURCE - UFTP configuration option	306
1.10.4.35	SSH_KEY - UFTP configuration option	307
1.10.4.36	SSL_CIPHER_LIST - UFTP configuration option	308
1.10.4.37	SYSTEM_ID - UFTP configuration option	309
1.10.4.38	UNHIDE - UFTP configuration option	310
1.10.4.39	USER - UFTP configuration option	311
1.10.4.40	VERIFY_HOST_NAME - UFTP configuration option	312
1.10.4.41	VERSION - UFTP configuration option	313
1.11	Universal Message Translator	314
1.11.1	Universal Message Translator - Usage	315
1.11.2	Universal Message Translator for zOS	317
1.11.3	Universal Message Translator for Windows and UNIX	319
1.11.4	Universal Message Translator for IBM i	320
1.11.5	Universal Message Translator Configuration Options	322
1.11.5.1	HELP - UMET configuration option	323
1.11.5.2	IGNORE_CASE - UMET configuration option	324
1.11.5.3	MESSAGE_FILE - UMET configuration option	325
1.11.5.4	MESSAGE_LEVEL - UMET configuration option	326
1.11.5.5	TRANSLATION_TABLE - UMET configuration option	327
1.11.5.6	VERSION - UMET configuration option	328
1.12	Universal Products Install Merge	329
1.12.1	Universal Products Install Merge - Usage	330
1.12.2	Universal Products Install Merge Configuration Options	331
1.12.2.1	BACKUP_DESTINATION - UPIMERGE configuration option	332
1.12.2.2	COMMAND_FILE_ENCRYPTED - UPIMERGE configuration option	333
1.12.2.3	COMMAND_FILE_PLAIN - UPIMERGE configuration option	334
1.12.2.4	COMPONENT_TYPE - UPIMERGE configuration option	335
1.12.2.5	CONFIGURATION_TYPE - UPIMERGE configuration option	336
1.12.2.6	DESTINATION_FILE - UPIMERGE configuration option	337
1.12.2.7	ENCRYPTION_KEY - UPIMERGE configuration option	338
1.12.2.8	HELP - UPIMERGE configuration option	339
1.12.2.9	INSTALLATION_DIRECTORY - UPIMERGE configuration option	340
1.12.2.10	KEEP_NOMATCH - UPIMERGE configuration option	341
1.12.2.11	KEYSTORE_PATH - UPIMERGE configuration option	342
1.12.2.12	MESSAGE_LEVEL - UPIMERGE configuration option	343
1.12.2.13	SOURCE_FILE - UPIMERGE configuration option	344
1.12.2.14	VERSION - UPIMERGE configuration option	345
1.13	Universal Query	346
1.13.1	Universal Query for zOS	347
1.13.2	Universal Query for Windows and UNIX	349
1.13.3	Universal Query for IBM i	351
1.13.4	Universal Query Configuration Options	353
1.13.4.1	BIF_DIRECTORY - UQUERY configuration option	354
1.13.4.2	CODE_PAGE - UQUERY configuration option	355
1.13.4.3	COMMAND_ID - UQUERY configuration option	356
1.13.4.4	COMPONENT_ID - UQUERY configuration option	357
1.13.4.5	HELP - UQUERY configuration option	358
1.13.4.6	MANAGERS - UQUERY configuration option	359
1.13.4.7	MESSAGE_LANGUAGE - UQUERY configuration option	360
1.13.4.8	MESSAGE_LEVEL - UQUERY configuration option	361
1.13.4.9	NLS_DIRECTORY - UQUERY configuration option	362
1.13.4.10	OUTBOUND_IP - UQUERY configuration option	363
1.13.4.11	PING - UQUERY configuration option	364
1.13.4.12	PLF_DIRECTORY - UQUERY configuration option	365

1.13.4.13 REMOTE_HOST - UQUERY configuration option	366
1.13.4.14 REMOTE_PORT - UQUERY configuration option	367
1.13.4.15 REPORT - UQUERY configuration option	368
1.13.4.16 SYSTEM_ID - UQUERY configuration option	369
1.13.4.17 VERSION - UQUERY configuration option	370
1.14 Universal Return Code	371
1.15 Universal Spool List	372
1.15.1 Universal Spool List for zOS	373
1.15.2 Universal Spool List for Windows and UNIX	376
1.15.3 Universal Spool List for IBM i	378
1.15.4 Universal Spool List Output	379
1.15.4.1 Universal Broker Output	380
1.15.4.2 Universal Command Output	382
1.15.4.3 Universal Event Monitor Output	383
1.15.5 Universal Spool List Configuration Options	387
1.15.5.1 BROKER_SPOOL_DIR - USLIST configuration option	388
1.15.5.2 COMPONENT - USLIST configuration option	389
1.15.5.3 HELP - USLIST configuration option	390
1.15.5.4 ID - USLIST configuration option	391
1.15.5.5 LIST - USLIST configuration option	392
1.15.5.6 MESSAGE_LEVEL - USLIST configuration option	393
1.15.5.7 MOUNT_POINT - USLIST configuration option	394
1.15.5.8 UCMD_SPOOL_DIR - USLIST configuration option	395
1.15.5.9 VERSION - USLIST configuration option	396
1.16 Universal Spool Remove	397
1.16.1 Universal Spool Remove for zOS	398
1.16.2 Universal Spool Remove for Windows and UNIX	401
1.16.3 Universal Spool Remove for IBM i	403
1.16.4 Universal Spool Remove Configuration Options	404
1.16.4.1 BROKER_SPOOL_DIR - USLRM configuration option	405
1.16.4.2 COMPONENT - USLRM configuration option	406
1.16.4.3 HELP - USLRM configuration option	407
1.16.4.4 MESSAGE_LEVEL - USLRM configuration option	408
1.16.4.5 MOUNT_POINT - USLRM configuration option	409
1.16.4.6 UCMD_SPOOL_DIR - USLRM configuration option	410
1.16.4.7 UEM_SERIALNO - USLRM configuration option	411
1.16.4.8 VERSION - USLRM configuration option	412
1.17 Universal Submit Job	413
1.17.1 Universal Submit Job Usage	414
1.17.2 Universal Submit Job Configuration Options	416
1.17.2.1 USBMJOB-Specific Configuration Options	417
1.17.2.1.1 COMMAND - USBMJOB configuration option	418
1.17.2.1.2 COPY_SPOOL_FILES - USBMJOB configuration option	419
1.17.2.1.3 ENCRYPTED_COMMAND_FILE - USBMJOB configuration option	420
1.17.2.1.4 JOB_LOG_LIBRARY - USBMJOB configuration option	421
1.17.2.1.5 JOB_STATUS_POLLING_INTERVAL - USBMJOB configuration option	422
1.17.2.1.6 KEY - USBMJOB configuration option	423
1.17.2.1.7 REMOTE_REFRESH_INTERVAL - USBMJOB configuration option	424
1.17.2.1.8 REMOTE_MESSAGE_PREFIX - USBMJOB configuration option	425
1.17.2.1.9 REMOTE_REPLY_COMMAND_PATH - USBMJOB configuration option	426
1.17.2.1.10 REMOTE_REPLY_HOST - USBMJOB configuration option	427
1.17.2.1.11 REMOTE_REPLY_PORT - USBMJOB configuration option	428
1.17.2.1.12 REMOTE_REPLY_USER_ID - USBMJOB configuration option	429
1.17.2.1.13 REMOTE_REPLY_USER_PWD - USBMJOB configuration option	430
1.17.2.1.14 SPECIFY_PRINT_CONTROL_CHARS - USBMJOB configuration option	431
1.17.2.1.15 TRACE - USBMJOB configuration option	432
1.17.2.1.16 USE_REMOTE_REPLY_FACILITY - USBMJOB configuration option	433
1.17.2.2 SBMJOB Encapsulated Configuration Options	434
1.17.3 Universal Submit Job Command Line Syntax	435
1.17.4 Universal Submit Job Remote Reply Facility	436
1.17.5 Universal Submit Job Return Codes	437
1.18 Universal Write-to-Operator	438
1.18.1 Universal Write-to-Operator Configuration Options	440
1.18.1.1 CONSOLE_ID - UWTO configuration option	441
1.18.1.2 CONSOLE_NAME - UWTO configuration option	442
1.18.1.3 HELP - UWTO configuration option	443
1.18.1.4 KEY - UWTO configuration option	444
1.18.1.5 MESSAGE - UWTO configuration option	445
1.18.1.6 MESSAGE_LEVEL - UWTO configuration option	446
1.18.1.7 REPLY - UWTO configuration option	447
1.18.1.8 TIMEOUT - UWTO configuration option	448
1.18.1.9 VERSION - UWTO configuration option	449
1.19 Universal Agent Utilities - Additional Information	450
1.19.1 Character Code Pages - UTIL	451
1.19.2 SSL/TLS Cipher Suites - UTIL	453
1.19.3 UTT Files - UTIL	454

Universal Agent Utilities 7.1.x Reference Guide

- [Universal Agent Utilities](#)
- [Types of Universal Agent Utilities](#)

Universal Agent Utilities

These pages provide detailed information for the Universal Agent utilities, which are installed as part of each Universal Agent package.

There are one or more pages for each utility, including pages that provide detailed information on the [configuration](#) options of each utility.

Some utilities are operating-system specific; they cannot be used on all supported operating system. The pages for each utility identify the operating system(s) on which they can be used.

Types of Universal Agent Utilities

The following table identifies the supported operating system(s) and provides a description for each Universal Agent utility. Each **Utility Name** is a link to detailed information about that utility.

Utility Name	z /OS	Windows	UNIX	IBM i	Description
Universal Certificate	✓	✓	✓		Creates digital certificates and private keys, which Universal Agent programs can use to securely identify users and computer systems.
Universal Control	✓	✓	✓	✓	Provides the ability to start and stop Universal Agent components, and to refresh Universal Agent configuration data.
Universal Copy		✓	✓	✓	Provides a means to copy files from either manager-to-server or server-to-manager.
Universal Database Dump	✓	✓	✓		Berkeley db_dump utility tailored specifically for Universal Agent databases.
Universal Database Load	✓	✓	✓		Berkeley db_load utility tailored specifically for Universal Agent databases.
Universal Display Log File				✓	Reads job log files; formats and writes job logs to standard out.
Universal Encrypt	✓	✓	✓	✓	Encrypts the contents of command files into an unintelligible format (for privacy reasons).
Universal Event Log Dump		✓			Selects records from one of the Windows event logs and writes them to a specified output file.
Universal FTP Client	✓	✓	✓		Transfers files to and from servers using any of four different file transfer protocols.
Universal Message Translator	✓	✓	✓	✓	Translates error messages into return (exit) codes based on a user-defined translation table.
Universal Products Install Merge		✓	✓		Merges options and values from one Universal Agent configuration or component definition file into another.
Universal Query	✓	✓	✓	✓	Queries any Universal Broker for Broker-related and active component-related information.
Universal Return Code		✓			Performs the function of ending a process with a return code that is equal to its command line argument.
Universal Spool List	✓	✓	✓	✓	Provides the ability to list Universal Spool database records.
Universal Spool Remove	✓	✓	✓	✓	Provides the ability to remove component records from the Universal Command and Universal Event Monitor Spool databases.
Universal Submit Job				✓	Encapsulates the IBM Submit Job (SBMJOB) command.
Universal Write-to-Operator	✓				For z/OS USS: Issues Write-to-Operator and Write-to-Operator-with-Reply messages.

Universal Agent Utilities Configuration

- [Overview](#)
- [Configuration Sources](#)
 - [Configuration File](#)
- [Configuration Options Information](#)
 - [Description](#)
 - [Usage](#)
 - [Values](#)
 - [<Additional Information>](#)

Overview

Configuration consists of:

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

Configuration Sources

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.

Detailed information on these methods of configuration can be found in [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.

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.

Configuration Options Information

For each Universal Agent utility configuration option, the following information is provided.

Description

Describes the configuration option and how it is used.

Usage

Provides a table of one or more of the following types of information:

Method	Syntax	IBM i	HP NonStop	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 Agent Utilities configuration options:

- Command Line Option, Short Form
- Command Line Option, Long Form
- Environment Variable
- Configuration File Keyword
- <IBM i> 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 **bold type**.

<Additional Information>

Identifies any additional information specific to the option.

Universal Certificate

Universal Certificate

The Universal Certificate (UCERT) utility creates digital certificates and private keys, which Universal Agent programs can use to securely identify users and computer systems.

The certificates created by Universal Certificate comply with the *Internet X.509 PublicKey Infrastructure* RFC 3280 document; however, not all certificate fields are supported.

The aim of Universal Certificate is to provide a simple certificate creation utility to be used if no Public Key Infrastructure (PKI) is available in your company. It is not a replacement for a corporate PKI.

See [X.509 Certificates](#) for an introduction to X.509 certificates and how they are used by Universal Agent components.

Detailed Information

The following pages provide detailed information for Universal Certificate:

- [Universal Certificate Usage](#)
- [Universal Certificate for z/OS](#)
- [Universal Certificate for UNIX and Windows](#)
- [Universal Certificate Configuration Options](#)

Universal Certificate Usage

- [Overview](#)
- [Certificate](#)
- [Certificate Requests](#)
- [Certificate Revocation List](#)
- [Transport Files](#)
- [Printing](#)
- [Verification](#)
- [File Formats](#)
- [Universal Certificate Database](#)

Overview

Universal Certificate performs the following operations, as specified by command line configuration options:

- Create certificates, certificate requests, private keys, certificate revocation lists (CRLs), and PKCS#12-encoded transport files.
- Print certificates, certificate requests, CRLs, and PKCS#12-encoded transport files.
- Verify certificates

The following sections describe each of these operations.

Certificate

A certificate is an electronic object use for identification purposes. A certificate identifies a person or computer system, as well as the party that issued the certificate. Certificates are issued by Certificate Authorities (CAs). A certificate only can be trusted if the CA that issued the certificate is trusted.

A certificate is created using the following input:

- Certificate request: Identifies the person / computer system for which the certificate is to be issued.
- CA certificate: Identifies the Certificate Authority (CA) that is issuing the certificate.
- CA private key: Signs (digitally) the certificate.

Certificate Requests

A certificate request is a request for a CA to issue a certificate. A certificate request contains all of the information required to identify a user / computer system.

The certificate request is saved in a file that is sent to a CA. The CA is responsible for verifying the information in the request and creating the final certificate based on that information.

When a certificate request is created, its corresponding private key also is created. The private key is written to a file and must remain private. File system security must be used to prevent unauthorized access to the private key file. Additionally, the private key can be protected with a password.

Certificate requests are encoded in Public-Key Cryptography Standards (PKCS) #10 syntax. Private keys are encoded in PKCS #8 syntax.

Certificate Revocation List

A Certificate Revocation List (CRL) is created by the Certificate Authority (CA). The list includes all certificates issued by the CA that subsequently have been revoked by the CA for some reason. The CRL is signed by the issuing CA.

A CRL is used as part of the certificate verification process to ensure that a certificate still is valid.

Transport Files

A transport file is a PKCS #12-encoded file generated by Universal Certificate in order to securely transfer a user's certificate and private key across systems.

Many applications that manage digital certificates - including RACF on z/OS and the Certificate Management add-in for the Microsoft Management Console application on Windows - can import a user's certificate using a transport file.

Universal Certificate also can extract certificate and private key information from PKCS #12-encoded transport files created by other applications. Command line options allow this extracted information to be stored in local files. If a CA's certificate - or the CA certificate chain - was added to the transport file, Universal Certificate can extract it as well.

Printing

Certificates, certificate requests, and transport files are saved in encoded files that are not easily readable (by people). However, they can be printed in text format.

Verification

Certificate verification is the process of verifying that a certificate is valid.

The certificate process consists of

1. Verifying that the certificate is issued by a trusted CA.
2. Verifying that the certificate is not revoked by the CA.

File Formats

Certificates, certificate requests, and private keys are stored in files.

The following file formats are supported:

- Privacy Enhanced Mail (PEM)
PEM is the format described in RFCs 1421-1424. PEM is a base64 encoding with header and trailer lines added to identify the contents. PEM is a text format suitable for email and text file transfers.
- Distinguished Encoding Rules (DER)
DER is an encoding rule based on the Abstract Syntax Notation 1 (ASN.1) specification. DER is a binary file format. When transferred across a network, it must be transferred in a binary or image mode.

All certificates and keys are encoded in an ASN.1 format. The PEM format is a text representation of the DER format.

Note



Universal Certificate supports only the DER format for PKCS #12-encoded transport files.

z/OS



PEM- and DER-formatted files can be written either to a member of a partitioned data set or a sequential data set. The record format must be variable or variable blocked. The record length must be at least 80.

Universal Certificate Database

Universal Certificate uses a database to maintain issued and revoked certificates. The database is required for certificate creation, certificate revocation, and CRL creation.

There is a one-to-one correspondence between a CA and a certificate database. That is, a unique database must be used for each CA, and each CA should use only one database.

The database is a very important element in maintaining a CA. Consequentially, it must be properly managed. The database must be secured from unauthorized updates and routinely backed up. The database is a regular text file.

UNIX	The database file name is specified with the <code>CERT_DB</code> option. If <code>CERT_DB</code> is not used, the database is created in the current working directory with name <code>ucert.db</code> .
Windows	The database file name is specified with the <code>CERT_DB</code> option. If <code>CERT_DB</code> is not used, the database is created in the current working directory with name <code>ucert.db</code> .
z/OS	The database is allocated to ddname <code>UNVDB</code> . The database allocation attributes are <code>DSORG=PS</code> , <code>RECFM={FB F}</code> , and <code>LRECL=1120</code> . The block size must be a multiple of <code>LRECL</code> if <code>RECFM</code> is <code>FB</code> .

Universal Certificate for zOS

Overview

Universal Certificate for z/OS executes as a batch job.

This section describes the Universal Certificate for z/OS JCL and command line options.

JCL Procedure

The following figure illustrates the Universal Certificate for z/OS JCL procedure (**UCRPRC**, located in the **SUNVSAMP** library), that is provide to simplify the execution JCL and future maintenance.

```
//UCRPRC  PROC UPARM=,           -- UCERT options
//          UCRPRE=#SHLQ.UNV,
//          UCRDBPRE=#PHLQ.UNV
// *
//PS1     EXEC PGM=UCERT, PARM='ENVAR(TZ=EST5EDT)/&UPARM'
//STEPLIB DD DSN=&UCRPRE..SUNVLOAD,
//          DISP=SHR
// *
//UNVDB   DD DSN=&UCRDBPRE..UCRDB,
//          DISP=SHR
//UNVNLS  DD DSN=&UCRPRE..SUNVNLS,
//          DISP=SHR
//UNVTRACE DD SYSOUT=*
// *
//SYSPRINT DD SYSOUT=*
//SYSOUT  DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

DD Statements used in JCL Procedure

The following table describes the DD statements used in the Universal Certificate for z/OS [JCL procedure](#), above.

ddname	Description
STEPLIB	Load library in which program UCERT is located.
UNVDB	UCERT certificate database.
UNVNLS	UCERT national language support ddname.
UNVTRACE	UCERT trace ddname.
SYSPRINT	UCERT standard output ddname.
SYSOUT	UCERT standard error ddname.

JCL

The following figure illustrates the Universal Certificate for z/OS JCL using the **UCRPRC**[JCL procedure](#), above.

```
//UCERT      EXEC PGM=UCERT
//STEPLIB   DD  DISP=SHR,DSN=UNV.SUNVLOAD
//UNVNLS    DD  DISP=SHR,DSN=UNV.SUNVNLS
//UNVDB     DD  DISP=SHR,DSN=UNV.UCRDB
//UNVTRACE  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//CEEDUMP   DD  SYSOUT=*
//SYSIN     DD  DUMMY
```

Command Line Syntax

The following figure illustrates the syntax - using the long form of command line options - of Universal Certificate for z/OS.

```
ucert[-codepagecodepage][-level {trace|audit|info|warn|error}] [-fileddname | -encryptedfileddname [-keykey] [-keypathpath]
```

Creating a certificate request.

```
{-createrequest
-request_fileddname [-request_format {pem|der}]
-private_key_fileddname [-private_key_format {pem|der}]
-private_key_infileddname [-private_key_format {pem|der}]
[-private_key_pwdpassword]
[-key_size {512|1024|2048|3072||4096}]
[-countryname]
[-statename]
[-localityname]
[-organizationname]
[-organizational_unitname]
[-common_namename]
{ [-dns_namename] | [-ip_addressname] }
[-sig_algalgorithm]
[-email_addressname]
```

Creating a certificate from a certificate request.

```
| -createcert
-request_fileddname [-request_format {pem|der}]
-cert_fileddname [-cert_format {pem|der}]
[-cert_dbddname]
-private_key_fileddname [-private_key_format {pem|der}]
-private_key_infileddname [-private_key_format {pem|der}]
[-private_key_pwdpassword]
-ca_cert_fileddname [-ca_cert_format {pem|der}]
[-serial_numbernumber]
[-not_before_datedate] [-not_after_datedate]
[-ca {yes|no}]
[-sig_algalgorithm]
```

Creating a certificate from a transport file.

```
| -createcert
-transport_fileddname [-transport_file_pwdpassword]
-cert_fileddname [-cert_format {pem|der}]
-private_key_fileddname [-private_key_format {pem|der}]
-private_key_infileddname [-private_key_format {pem|der}]
-ca_cert_fileddname [-ca_cert_format {pem|der}]
```

Creating a certificate revocation list.

```
| -createcrl
-crl_fileddname [-crl_format {pem|der}]
-ca_cert_fileddname [-ca_cert_format {pem|der}]
-private_key_fileddname [-private_key_format {pem|der}]
-private_key_infileddname [-private_key_format {pem|der}]
[-private_key_pwdpassword]
-next_update_daysdays
-next_update_hourshours
[-cert_dbddname]
```

Creating a transport file.

```
| -createtransport
-transport_fileddname [-transport_file_pwdpassword]
```

```

-cert_fileddname [-cert_format {pem|der}]
-private_key_fileddname [-private_key_formatpem|der}]
-private_key_infileddname [-private_key_format {pem|der}]
[-private_key_pwdpassword]
-ca_cert_fileddname [-ca_cert_format {pem|der}]
[-common_name]
[-keypbeencryption]
[-certpbeencryption]

```

Revoking a certificate.

```

| -revokecert
[-revoke_reason {unspecified|keyCompromise|caCompromised|affiliationChange|superseded|
cessationofOperation|privilegeWithdrawn}]
-cert_fileddname [-cert_format {pem|der}]
[-cert_dbddname]

```

Printing a certificate request.

```

| -printrequest
-request_fileddname [-request_format {pem|der}]

```

Printing a certificate.

```

| -printcert
-cert_fileddname [-cert_format {pem|der}]

```

Printing a certificate revocation list.

```

| -printcrl
-crl_fileddname [-crl_format {pem|der}]

```

Printing a transport file.

```

| -printtransport
-transport_fileddname [-transport_file_pwdpassword]

```

Verifying a certificate.

```

| -verifycert
-cert_fileddname [-cert_format {pem|der}]
-ca_cert_fileddname [-ca_cert_format {pem|der}]
-crl_fileddname [-crl_format {pem|der}]

```

ucert

```
{ -help | -version }
```

Universal Certificate for UNIX and Windows

Overview

Universal Certificate for UNIX and Windows executes as a command line application.

Command Line Syntax

The following figure illustrates the syntax - using the long form of command line options - of Universal Certificate for UNIX and Windows.

```
ucert
[-codepagecodepage]
[-nls_directorycodepage]
[-level {trace|audit|info|warn|error}]
[-filefilename | -encryptedfilefilename [-keykey] [-keypathpath] ]

Creating a certificate request.
{-createrequest-request_filefilename [-request_format {pem|der}]}
-private_key_filefilename [-private_key_format {pem|der}]
-private_key_infilefilename [-private_key_format {pem|der}]
[-private_key_pwdpassword]
[-key_size {512|1024|2048|3072||4096}]
[-countryname]
[-statename]
[-localityname]
[-organizationname]
[-organizational_unitname]
[-common_namename]
{ [-dns_name] | [-ip_addressname] }
[-sig_algorithmlgorithm]
[-email_addressname]

Creating a certificate from a certificate request.
| -createcert-request_filefilename [-request_format {pem|der}]
-cert_filefilename [-cert_format {pem|der}]
[-cert_dbfilename]
-private_key_filefilename [-private_key_format {pem|der}]
-private_key_infilefilename [-private_key_format {pem|der}]
[-private_key_pwdpassword]
-ca_cert_filefilename [-ca_cert_format {pem|der}]
[-serial_numbernumber]
[-not_before_date] [-not_after_date]
[-ca {yes|no}]
[-sig_algorithmlgorithm]

Creating a certificate from a transport file.
| -createcert-transport_filefilename [-transport_file_pwdpassword]
-cert_filefilename [-cert_format {pem|der}]
-private_key_filefilename [-private_key_format {pem|der}]
-private_key_infilefilename [-private_key_format {pem|der}]
-ca_cert_filefilename [-ca_cert_format {pem|der}]

Creating a certificate revocation list.
| -createcrl-crl_filefilename [-crl_format {pem|der}]
-ca_cert_filefilename [-ca_cert_format {pem|der}]
-private_key_filefilename [-private_key_format {pem|der}]
-private_key_infilefilename [-private_key_format {pem|der}]
[-private_key_pwdpassword]
[-next_update_daysdays-next_update_hourshours]
[-cert_dbfilename]

Creating a transport file.
| -createtransport-transport_filefilename [-transport_file_pwdpassword]
-cert_filefilename [-cert_format {pem|der}]
-private_key_filefilename [-private_key_formatpem|der]}
-private_key_infilefilename [-private_key_format {pem|der}]
[-private_key_pwdpassword]
-ca_cert_filefilename [-ca_cert_format {pem|der}]
[-common_namename]
```

```
[-keypbeencryption]
[-certpbeencryption]

Revoking a certificate.
| -revokecert
[-revoke_reason {unspecified|keyCompromise|caCompromised|affiliationChange|superseded|cessationofOperation|privilegeWithdrawn}]
-cert_filefilename [-cert_format {pem|der}]
[-cert_dbfilename]

Printing a certificate request.
| -printrequest-request_filefilename [-request_format {pem|der}]

Printing a certificate.
| -printcert-cert_filefilename [-cert_format {pem|der}]

Printing a certificate revocation list.
| -printcrl-crl_filefilename [-crl_format {pem|der}]

Printing a transport file.
| -printtransport-transport_filefilename [-transport_file_pwdpassword]

Verifying a certificate.
| -verifycert-cert_filefilename [-cert_format {pem|der}]
-ca_cert_filefilename [-ca_cert_format {pem|der}]
-crl_filefilename [-crl_format {pem|der}]

ucert
{ -help | -version }
```

Universal Certificate Configuration Options

Universal Certificate Configuration Options

The following table identifies the Universal Certificate configuration options for the UNIX, Windows, and z/OS operating systems. Each **Option Name** is a link to detailed information about that configuration option.

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

For information on how these options are used, see [Universal Certificate Usage](#).

Option Name	Description
CA	Specification for whether or not the certificate should be marked as a Certificate Authority certificate.
CA_CERT_FILE	Name from which the CA certificate is read.
CA_CERT_FORMAT	Format of the CA certificate file specified by CA_CERT_FILE.
CERT_DB	Certificate database name.
CERT_FILE	File name to which the certificate is written.
CERT_FORMAT	Format of the certificate file specified by CERT_FILE.
CERTPBE	Password-based encryption to use for the certificate in a PKCS #12-encoded transport file.
CODE_PAGE	Character code page used to translate text data.
COMMAND_FILE_ENCRYPTED	Name of an encrypted command file.
COMMAND_FILE_PLAIN	Name of a plain text command file.
COMMON_NAME	Common name of the subject field of a certificate.
COUNTRY	Country name of the subject field of a certificate.
CREATE	Specification that UCERT is to create a certificate request or a certificate.
CRL_FILE	File name to which the Certificate Revocation List (CRL) is written.
CRL_FORMAT	Format of the CRL file specified by CRL_FILE.
DNS_NAME	Domain Name System (DNS) name of the computer system for which the certificate identifies.
EMAIL_ADDRESS	Email address of the entity identified by the certificate.
ENCRYPTION_KEY	Key used to encrypt the command file.
HELP	Writes a description of the command options and their format.
IP_ADDRESS	Internet Protocol (IP) address of the computer system for which the certificate identifies.
KEY_SIZE	Key size of the RSA public / private keys.
KEYPBE	Password-based encryption to use for the private key in a PKCS #12-encoded transport file.
KEYSTORE_PATH	Path to a local or remote Universal Broker service interface from which an encryption key can be obtained.
LOCALITY	Locality name of the subject field of a certificate.
MESSAGE_LEVEL	Level of messages to write.
NEXT_UPDATE_DAYS	Number of days to the next CRL update.
NEXT_UPDATE_HOURS	Number of hours to the next CRL update.
NLS_DIRECTORY	Directory name where the code page UTT files are located.
NOT_AFTER_DATE	Last day for which the certificate is considered valid.
NOT_BEFORE_DATE	First day for which the certificate is considered valid.
ORGANIZATION	Organization name of the subject field of a certificate.
ORGANIZATIONAL_UNIT	Organizational unit name of the subject field of a certificate.
PRINT	Specification that UCERT is to print a certificate request or a certificate.

PRIVATE_KEY_FILE	File from which the RSA private key is read or to which the RSA private key is written.
PRIVATE_KEY_FORMAT	Format of the private key file specified by PRIVATE_KEY_FILE.
PRIVATE_KEY_INFILE	File from which an RSA private key is read.
PRIVATE_KEY_PWD	Password used to read and write the private key file specified by PRIVATE_KEY_FILE.
REQUEST_FILE	File name from which the certificate request is read or to which the certificate request is written.
REQUEST_FORMAT	Format of the certificate request file specified by REQUEST_FILE.
REVOKE	Specification that UCERT is to revoke a certificate.
REVOKE_REASON	Reason a certificate is being revoked.
SERIAL_NUMBER	Unique serial number to be assigned to the created certificate.
SIGNATURE_ALGORITHM	Signature algorithm to use to create a certificate request or certificate.
STATE	State name of the subject field of a certificate.
TRANSPORT_FILE	File containing certificate / private key information.
TRANSPORT_FILE_PWD	Password used to protect the file specified by TRANSPORT_FILE.
VERIFY	Specification that UCERT is to verify a certificate.
VERSION	Writes the program version and copyright information.

CA - Universal Certificate configuration option

Description

The CA option specifies whether or not the certificate should be marked as a Certificate Authority certificate.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-ca <i>option</i>			✓	✓	✓
Environment Variable	UCRCA= <i>option</i>			✓	✓	✓

Values

option is the specification for whether or not the certificate is a CA certificate.

Valid values for *option* are:

- **yes**
Certificate is marked as a CA certificate. This is accomplished by setting the certificate **basicConstraint** extension **cA** to *true*.
- **no**
Certificate is not marked as a CA certificate.

Default is no.

CA_CERT_FILE - Universal Certificate configuration option

Description

The CA_CERT_FILE option specifies either:

- Name of the file from which the CA certificate is read.
- Name of the file to which the CA certificate is written.

(The CA certificate identifies the issuer of the certificate being created.)

The format of the file is specified by the [CA_CERT_FORMAT](#) option.

CA certificate information also can be imported from a transport file (specified via the [TRANSPORT_FILE](#) option). In this case, CA_CERT_FILE specifies the name of the file to which one or more CA certificates are written.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-a <i>ddname</i> or <i>file</i></code>			✓	✓	✓
Command Line, Long Form	<code>-ca_cert_file <i>ddname</i> or <i>file</i></code>			✓	✓	✓
Environment Variable	<code>UCRCACERTFILE=<i>ddname</i> or <i>file</i></code>			✓	✓	✓

Values

ddname or *file* is the name of the file.

CA_CERT_FORMAT - Universal Certificate configuration option

Description

The CA_CERT_FORMAT option specifies the format of the CA certificate file specified by the CA_CERT_FILE option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-A <i>format</i>			✓	✓	✓
Command Line, Long Form	-ca_cert_format <i>format</i>			✓	✓	✓
Environment Variable	UCRCACERTFORMAT= <i>format</i>			✓	✓	✓

Values

format is the format of the CA certificate file.

Valid values for *format* are:

- **pem**
PEM-formatted file
- **der**
A DER-formatted file

Default is pem.

See **File Formats** in [Universal Certificate Usage](#) for details on file formats.

CERT_DB - Universal Certificate configuration option

Description

The CERT_DB option specifies the name of the certificate database.

UNIX and Windows



If the name of the certificate database is not specified, the certificate database is created in the current working directory with name **ucert.db**.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-cert_db <i>ddname</i> or <i>file</i>			✓	✓	✓
Environment Variable	UCRCERTDB= <i>ddname</i> or <i>file</i>			✓	✓	✓

Values

ddname or *file* is the name of the certificate database.

CERT_FILE - Universal Certificate configuration option

Description

The CERT_FILE option specifies either:

- Name of the file to which the certificate is written
- Name of the file from which the certificate is read

The format of the file is specified by the [CERT_FORMAT](#) option.

Certificate information also can be imported from a transport file (specified via the [TRANSPORT_FILE](#) option). In this case, CERT_FILE specifies the file name to which the certificate is written.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-d <i>ddname</i> or <i>file</i>			✓	✓	✓
Command Line, Long Form	-cert_file <i>ddname</i> or <i>file</i>			✓	✓	✓
Environment Variable	UCRCERTFILE= <i>ddname</i> or <i>file</i>			✓	✓	✓

Values

ddname or *file* is the name of the file.

CERT_FORMAT - Universal Certificate configuration option

Description

The CERT_FORMAT option specifies the format of the certificate file specified by the [CERT_FILE](#) option.

Usage

Method	Method	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-D <i>format</i>			✓	✓	✓
Command Line, Long Form	-cert_format <i>format</i>			✓	✓	✓
Environment Variable	UCRCERTFORMAT= <i>format</i>			✓	✓	✓

Values

format is the format of the certificate file.

Valid values for *format* are:

- **pem**
PEM-formatted file
- **der**
A DER-formatted file

Default is pem.

See **File Formats** in [Universal Certificate Usage](#) for details on file formats.

CERTPBE - Universal Certificate configuration option

Description

The CERTPBE option specifies the password-based encryption to use for the certificate in a PKCS #12-encoded transport file.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-certpbe <i>pbe</i>			✓	✓	✓
Environment Variable	n/a					

Values

pbe is the password-based encryption to use for the certificate in a PKCS #12-encoded transport file.

Valid values for *pbe* are:

- SHA1-RC2-40
- SHA1-RC2-64
- SHA1-RC2-128
- SHA1-2DES
- SHA1-3DES

CODE_PAGE - Universal Certificate configuration option

Description

The CODE_PAGE option specifies the character code page used to translate text data.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-t <i>codepage</i>			✓	✓	✓
Command Line, Long Form	-codepage <i>codepage</i>			✓	✓	✓
Environment Variable	UCRCODEPAGE= <i>codepage</i>			✓	✓	✓

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 (see [UTT Files](#) for information on 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**.)

Default

The default code page 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_FILE_ENCRYPTED - Universal Certificate configuration option

Description

The `COMMAND_FILE_ENCRYPTED` option specifies the name of an encrypted command file.

Command files specify an additional source of command line options. The options read from the file are processed exactly like options specified on the command line. Encrypted command files are an excellent place to store sensitive data such as passwords.

Universal Certificate is able to process command files that are either encrypted or plain text (see [COMMAND_FILE_PLAIN](#)). Use the Universal Encrypt utility to encrypt a plain text command file (see [Universal Encrypt](#)).

Command files (encrypted or not) that contain sensitive data should be protected from unauthorized read access with file level security.

Note



If an encrypted file is specified in this option, a plain text file should not be specified additionally in the [COMMAND_FILE_PLAIN](#) option. If it is, the file specified in [COMMAND_FILE_PLAIN](#) will be used.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-x ddname or filename</code>			✓	✓	✓
Command Line, Long Form	<code>-encryptedfile ddname or filename</code>			✓	✓	✓
Environment Variable	n/a					

Values

`ddname` or `filename` is the name of the encrypted command file.

COMMAND_FILE_PLAIN - Universal Certificate configuration option

Description

The `COMMAND_FILE_PLAIN` option specifies the name of a plain text command file.

Command files specify an additional source of command line options. The options read from the file are processed exactly like options specified on the command line.

Universal Certificate is able to process command files that are either encrypted or plain text (see [COMMAND_FILE_ENCRYPTED](#)). Command files (encrypted or not) that contain sensitive data should be protected from unauthorized read access with file level security.

Note



If an encrypted file is specified in this option, a plain text file should not be specified additionally in the [COMMAND_FILE_ENCRYPTED](#) option. If it is, the file specified in this `COMMAND_FILE_PLAIN` option will be used.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-f ddname or filename</code>			✓	✓	✓
Command Line, Long Form	<code>-file ddname or filename</code>			✓	✓	✓
Environment Variable	n/a					

Values

`ddname` or `filename` is the name of the plain text command file.

COMMON_NAME - Universal Certificate configuration option

Description

The COMMON_NAME option specifies the common name of the **subject** field of a certificate.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-common_name <i>name</i>			✓	✓	✓
Environment Variable	UCRCOMMONNAME= <i>name</i>			✓	✓	✓

Values

name is the common name of the **subject** field.

More specifically, *name* is the certificate's **commonName** (CN) relative distinguished name (RDN) attribute of the **subject** distinguished name (DN).

COUNTRY - Universal Certificate configuration option

Description

The COUNTRY option specifies the country name of the **subject** field of a certificate.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-country <i>name</i>			✓	✓	✓
Environment Variable	UCRCOUNTRY= <i>name</i>			✓	✓	✓

Values

name is the country name of the **subject** field.

More specifically, *name* is the certificate's **countryName** (C) relative distinguished name (RDN) attribute of the **subject** distinguished name (DN); a two-character country code as defined by the ISO 3166 standard.

Country Codes

The following table identifies a subset of ISO 3166 country codes.

Code	Country
AU	Australia
BE	Belgium
CA	Canada
DK	Denmark
FR	France
DE	Germany
IT	Italy
NL	Netherlands
NO	Norway
PT	Portugal
ES	Spain
SE	Sweden
CH	Switzerland
GB	United Kingdom
US	United States

CREATE - Universal Certificate configuration option

Description

The CREATE option specifies that Universal Certificate is to create a certificate, certificate request, certificate revocation list (CRL), or a transport file.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-c <i>object</i>			✓	✓	✓
Command Line, Long Form	-create <i>object</i>			✓	✓	✓
Environment Variable	UCRCREATE= <i>object</i>			✓	✓	✓

Values

object is the specification for what Universal Certificate is to create.

Valid values for *object* are:

- **cert**
Create an X.509 certificate.
- **request**
Create a certificate request.
- **crl**
Create a Certificate Revocation List (CRL).
- **transport**
Create a PKCS #12-encoded transport file.

CRL_FILE - Universal Certificate configuration option

Description

The CRL_FILE option specifies the name of the file to which the Certificate Revocation List (CRL) is written.

The format of the file is specified by the [CRL_FORMAT](#) option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-crl_file <i>ddname</i> or <i>file</i>			✓	✓	✓
Environment Variable	UCRCRLFILE= <i>ddname</i> or <i>file</i>			✓	✓	✓

Values

ddname or *file* is the name of the file to which the Certificate Revocation List (CRL) is written.

CRL_FORMAT - Universal Certificate configuration option

Description

The CRL_FORMAT option specifies the format of the CRL file specified by the [CRL_FILE](#) option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-crl_format <i>format</i>			✓	✓	✓
Environment Variable	UCRCRLFORMAT= <i>format</i>			✓	✓	✓

Values

format is the format of the CRL file.

Valid values for *format* are:

- **pem**
PEM-formatted file
- **der**
A DER-formatted file

Default is pem.

See **File Formats**[Universal Certificate Usage](#) for details on file formats.

DNS_NAME - Universal Certificate configuration option

Description

The `DNS_NAME` option specifies the Domain Name System (DNS) name of the computer system identified by the certificate.

Usage


Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	<code>-dns_name <i>name</i></code>			✓	✓	✓
Environment Variable	<code>UCRDNSNAME=<i>name</i></code>			✓	✓	✓

Values

name is the Domain Name System (DNS) name of the computer system.

More specifically, *name* is the **dNSName** component of the **subjectAltName** extension.

Note

 An IP address (for example, **10.20.30.40**) should not be used. IP address values are specified with the [IP_ADDRESS](#) option.

EMAIL_ADDRESS - Universal Certificate configuration option

Description

The EMAIL_ADDRESS option specifies the e-mail address of the entity identified by the certificate.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-email_address <i>name</i>			✓	✓	✓
Environment Variable	UCREMAILADDRESS= <i>name</i>			✓	✓	✓

Values

name is the e-mail address of the entity.

More specifically, *name* is the **rfc822Name** component of the **subjectAltName** extension.

The format of *name* is defined by RFC 822. The name is of the form **local-part@domain**.

ENCRYPTION_KEY - Universal Certificate 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 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; otherwise, the decryption will fail.

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

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-K <i>key</i>			✓	✓	✓
Command Line, Long Form	-key <i>key</i>			✓	✓	✓
Environment Variable	UCRKEY= <i>key</i>			✓	✓	✓

Values

key is the key used to encrypt the command file.

HELP - Universal Certificate configuration option

Description

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

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-h			✓	✓	✓
Command Line, Long Form	-help			✓	✓	✓
Environment Variable	n/a					

Values

(There are no values used with this option.)

IP_ADDRESS - Universal Certificate configuration option

Description

The IP_ADDRESS option specifies the Internet Protocol (IP) address of the computer system identified by the certificate.

Usage


Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-ip_address <i>name</i>			✓	✓	✓
Environment Variable	UCRIPADDRESS= <i>name</i>			✓	✓	✓

Values

name is the Internet Protocol (IP) address of the computer system.

More specifically, *name* is the **IPAddress** component of the **subjectAltName** extension.

Note

 DNS names (for example, **sysa.acme.com**) should not be used. DNS name values are specified with the [DNS_NAME](#) option.

KEY_SIZE - Universal Certificate configuration option

Description

The KEY_SIZE option specifies the key size of the RSA public / private keys.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-s <i>size</i>			✓	✓	✓
Command Line, Long Form	-key_size <i>size</i>			✓	✓	✓
Environment Variable	UCRKEYSIZE= <i>size</i>			✓	✓	✓

Values

size is the key size (number of bits) of the RSA public/private keys.

Valid values for *size* are:

- 512
- 1024
- 2048
- 3072
- 4096

Default is 2048.

KEYPBE - Universal Certificate configuration option

Description

The KEYPBE option specifies the password-based encryption to use for the private key in a PKCS #12-encoded transport file.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-keypbe <i>pbe</i>			✓	✓	✓
Environment Variable	n/a					

Values

pbe is the password-based encryption to use for the private key in a PKCS #12-encoded transport file.

Valid values for *pbe* are:

- SHA1-3DES

KEYSTORE_PATH - Universal Certificate configuration option

Description

The KEYSTORE_PATH option specifies the local or remote Universal Broker service interface from which an [encryption key](#) can be obtained.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-p <i>path</i>			✓	✓	✓
Command Line, Long Form	-keypath <i>path</i>			✓	✓	✓
Environment Variable	n/a					

Value

path is the local or remote Universal Broker service interface..

LOCALITY - Universal Certificate configuration option

Description

The LOCALITY option specifies the locality name of the **subject** field of a certificate.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-locality <i>name</i>			✓	✓	✓
Environment Variable	UCRLOCALITY= <i>name</i>			✓	✓	✓

Values

name is the locality name of the **subject** field.

More specifically, *name* is the certificate's **localityName** (L) relative distinguished name (RDN) attribute of the **subject** distinguished name (DN).

MESSAGE_LEVEL - Universal Certificate configuration option

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l <i>level</i>			✓	✓	✓
Command Line, Long Form	-level <i>level</i>			✓	✓	✓
Environment Variable	UCRLEVEL= <i>level</i>			✓	✓	✓

Values

level indicates either of the following level of messages:

- **trace**
Writes trace messages used for diagnostic purposes (see **Trace Files**, below).

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

UNIX	warn
Windows	warn
z/OS	info

Trace Files

UNIX	Trace file name is ucert.trc . It is created in the current working directory.
Windows	Trace file name is ucert.trc . It is created in the current working directory.
z/OS	Trace file is written to ddname UNVTRACE .

NEXT_UPDATE_DAYS - Universal Certificate configuration option

Description

The NEXT_UPDATE_DAYS option specifies the number of days to the next CRL update.

The CRL **nextUpdate** value is set to the current date plus the number of days specified in this option.

CRL creation requires the use of either this option or [NEXT_UPDATE_HOURS](#).

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-next_update_days <i>days</i>			✓	✓	✓
Environment Variable	UCRNEXTUPDATEDAYS= <i>days</i>			✓	✓	✓

Values

days is the number of days to the next CRL update.

Default is 0.

NEXT_UPDATE_HOURS - Universal Certificate configuration option

Description

The NEXT_UPDATE_HOURS option specifies the number of hours to the next CRL update.

The CRL **nextUpdate** value is set to the current date plus the number of hours specified in this option.

CRL creation requires the use of either this option or [NEXT_UPDATE_DAYS](#).

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-next_update_hours <i>hours</i>			✓	✓	✓
Environment Variable	UCRNEXTUPDATEHOURS= <i>hours</i>			✓	✓	✓

Values

hours is the number of hours to the next CRL update.

Default is 0.

NLS_DIRECTORY - Universal Certificate configuration option

Description

The NLS_DIRECTORY option specifies the name of the directory where the code page UTT files are located.


Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-nls_directory <i>directory</i>			✓	✓	
Environment Variable	UCRNLSDIRECTORY= <i>directory</i>			✓	✓	

Values

directory is the name of the directory.

UNIX and Windows

 Relative path names are relative to the current working directory.

NOT_AFTER_DATE - Universal Certificate configuration option

Description

The NOT_AFTER_DATE option specifies the last day for which the certificate is considered valid.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-not_after_date <i>date</i>			✓	✓	✓
Environment Variable	UCRNOTAFTERDATE= <i>date</i>			✓	✓	✓

Values

date is the last day for which the certificate is considered valid.

The format of date is either:

- *YYYY.MM.DD*
- *DAYS* (number of days after the current date)

Default is 365.

NOT_BEFORE_DATE - Universal Certificate configuration option

Description

The NOT_BEFORE_DATE option specifies the first day for which the certificate is considered valid.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-not_before_date <i>date</i>			✓	✓	✓
Environment Variable	UCRNOTBEFOREDATE= <i>date</i>			✓	✓	✓

Values

date is the first day for which the certificate is considered valid.

The format of *date* is *YYYY.MM.DD*.

Default is current date.

ORGANIZATION - Universal Certificate configuration option

Description

The ORGANIZATION option specifies the organization name of the **subject** field of a certificate.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-organization <i>name</i>			✓	✓	✓
Environment Variable	UCRORGANIZATION= <i>name</i>			✓	✓	✓

Values

name is the organization name of the subject field of a certificate.

More specifically, *name* is the certificate's **organizationName** (O) relative distinguished name (RDN) attribute of the **subject** distinguished name (DN).

ORGANIZATIONAL_UNIT - Universal Certificate configuration option

Description

The ORGANIZATIONAL_UNIT option specifies the organizational unit name of the **subject** field of a certificate.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-organizational_unit <i>name</i>			✓	✓	✓
Environment Variable	UCRORGANIZATIONALUNIT= <i>name</i>			✓	✓	✓

Values

name is the organizational unit name of the **subject** field of a certificate.

More specifically, *name* is the certificate's **organizationalUnitName** (OU) relative distinguished name (RDN) attribute of the **subject** distinguished name (DN).

PRINT - Universal Certificate configuration option

Description

The PRINT option specifies that Universal Certificate is to print a certificate, certificate request, certificate revocation list (CRL), or transport file.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-p <i>object</i>			✓	✓	✓
Command Line, Long Form	-print <i>object</i>			✓	✓	✓
Environment Variable	UCRPRINT= <i>object</i>			✓	✓	✓

Values

object is the specification for what to print.

Valid values for *object* are:

- **cert**
Print an X.509 certificate.
- **request**
Print a certificate request.
- **crl**
Print a Certificate Revocation List (CRL).
- **transport**
Print a PKCS#12-encoded transport file.

PRIVATE_KEY_FILE - Universal Certificate configuration option

Description

The PRIVATE_KEY_FILE option specifies either:

- File from which the RSA private key is read (certificate creation/verification only, use the [PRIVATE_KEY_INFILE](#) option to re-use private keys for request creation).
- File to which the RSA private key is written when generating a certificate request.

The key is encoded in a password encrypted PKCS #8 syntax.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-e ddname or file</code>			✓	✓	✓
Command Line, Long Form	<code>-private_key_file ddname or file</code>			✓	✓	✓
Environment Variable	<code>UCRPRIVATEKEYFILE=ddname or file</code>			✓	✓	✓

Values

ddname or *file* is the name of the file.

The format of the file is specified by the [PRIVATE_KEY_FORMAT](#) option.

PRIVATE_KEY_FORMAT - Universal Certificate configuration option

Description

The `PRIVATE_KEY_FORMAT` option specifies the format of the private key file specified by the `PRIVATE_KEY_FILE` option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-E <i>format</i></code>			✓	✓	✓
Command Line, Long Form	<code>-private_key_format <i>format</i></code>			✓	✓	✓
Environment Variable	<code>UCRPRIVATEKEYFORMAT=<i>format</i></code>			✓	✓	✓

Values

format is the format of the private key file.

Valid values for *format* are:

- **pem**
PEM-formatted file
- **der**
A DER-formatted file

Default is pem.

See [File Formats](#) for details on file formats.

PRIVATE_KEY_INFILE - Universal Certificate configuration option

Description

The PRIVATE_KEY_INFILE option specifies a file from which an RSA private key is read.

It allows an existing PKCS#8-encoded key to be re-used when creating certificate requests.

The format of the file is specified by the [PRIVATE_KEY_FORMAT](#) option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-i <i>ddname</i> or <i>file</i>			✓	✓	✓
Command Line, Long Form	-private_key_infile <i>ddname</i> or <i>file</i>			✓	✓	✓
Environment Variable	UCRPRIVATEKEYINFILE= <i>ddname</i> or <i>file</i>			✓	✓	✓

Values

ddname or *file* is the name of the file.

PRIVATE_KEY_PWD - Universal Certificate configuration option

Description

The PRIVATE_KEY_PWD option specifies the private key password that is used to read and write the private key file specified by the [PRIVATE_KEY_FILE](#) option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-w <i>password</i></code>			✓	✓	✓
Command Line, Long Form	<code>-private_key_pwd <i>password</i></code>			✓	✓	✓
Environment Variable	<code>UCRPRIVATEKEYPWD=<i>password</i></code>			✓	✓	✓

Values

password is the private key password.

REQUEST_FILE - Universal Certificate configuration option

Description

The REQUEST_FILE option specifies either:

- Name of the file from which the certificate request is read.
- Name of the file to which the certificate request is written.

The request is encoded in PKCS #10 syntax.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-r <i>ddname</i> or <i>file</i>			✓	✓	✓
Command Line, Long Form	-request_file <i>ddname</i> or <i>file</i>			✓	✓	✓
Environment Variable	UCRREQUESTFILE= <i>ddname</i> or <i>file</i>			✓	✓	✓

Values

ddname or *file* is the name of the file.

The format of the file is specified by the [REQUEST_FORMAT](#) option.

REQUEST_FORMAT - Universal Certificate configuration option

Description

The REQUEST_FORMAT option specifies the format of the certificate request file specified by the [REQUEST_FILE](#) option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-R <i>format</i>			✓	✓	✓
Command Line, Long Form	-request_format <i>format</i>			✓	✓	✓
Environment Variable	UCRREQUESTFORMAT= <i>format</i>			✓	✓	✓

Values

format is the format of the certificate request file.

Valid values for *format* are:

- **pem**
PEM-formatted file
- **der**
DER-formatted file

Default is pem.

See [File Formats](#) for details on file formats.

REVOKE - Universal Certificate configuration option

Description

The REVOKE option specifies that Universal Certificate is to revoke a certificate.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-revoke <i>object</i>			✓	✓	✓
Environment Variable	UCRREVOKE= <i>object</i>			✓	✓	✓

Values

object is the specification to revoke a certificate.

Valid values for *object* are:

- **cert**
Instructs Universal Certificate to revoke an X.509 certificate.

REVOKE_REASON - Universal Certificate configuration option

Description

The REVOKE_REASON option specifies the reason that a certificate is being revoked.

(Valid reasons for certificate revocation are defined as part of RFC 3280.)

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-revoke_reason <i>reason</i>			✓	✓	✓
Environment Variable	UCRREVOKEREASON= <i>reason</i>			✓	✓	✓

Values

reason is the reason a certificate is being revoked.

Universal Certificate accepts the following valid values for *reason*:

- **unspecified**
No reason is given.
- **keyCompromise**
Subject's private key, or some other aspect of the subject, has been compromised.
- **caCompromised**
CA private key, or some other aspect of the subject, has been compromised.
- **affiliationChange**
Subject's name or other information in the certificate has changed. There is no reason to suspect the private key is compromised.
- **superseded**
Certificate has been superseded by another certificate. There is no reason to suspect the private key is compromised.
- **cessationOfOperation**
Certificate is no longer required for the purpose it was issued. There is no reason to suspect the private key is compromised.
- **privilegeWithdrawn**
Privilege contained within the certificate is withdrawn.

Default is unspecified.

SERIAL_NUMBER - Universal Certificate configuration option

Description

The SERIAL_NUMBER option specifies a unique serial number to be assigned to the created certificate.

If SERIAL_NUMBER is not used to specify a serial number, Universal Certificate automatically generates a random 8-byte serial number for the certificate.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-serial_number <i>number</i>			✓	✓	✓
Environment Variable	UCRSERIALNUMBER= <i>number</i>			✓	✓	✓

Values

number is the serial number to be assigned to the certificate.

SIGNATURE_ALGORITHM - Universal Certificate configuration option

Description

The SIGNATURE_ALGORITHM option specifies the signature algorithm to use to create a certificate request or certificate.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-sig_alg <i>algorithm</i>			✓	✓	✓
Environment Variable	n/a					

Values

algorithm is the signature algorithm to use to create a certificate request or certificate.

Valid values for *algorithm* are:

- MD5
- SHA1
- SHA256
- SHA384
- SHA512

Default is SHA1.

STATE - Universal Certificate configuration option

Description

The STATE option specifies the state name of the **subject** field of a certificate.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-state <i>name</i>			✓	✓	✓
Environment Variable	UCRSTATE= <i>name</i>			✓	✓	✓

Values

name is the state name of the **subject** field.

More specifically, *name* is the certificate's **stateName** (S) relative distinguished name (RDN) attribute of the **subject** distinguished name (DN).

TRANSPORT_FILE - Universal Certificate configuration option

Description

The TRANSPORT_FILE option specifies either:

- Name of the file from which the certificate and private key is read
- Name of the file to which the certificate and private key is written

It is a DER-formatted file encoded in PKCS#12 syntax.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-transport_file <i>ddname</i> or <i>file</i>			✓	✓	✓
Environment Variable	UCRTRANFILE= <i>ddname</i> or <i>file</i>			✓	✓	✓

Values

ddname or *file* is the name of the file.

TRANSPORT_FILE_PWD - Universal Certificate configuration option

Description

The TRANSPORT_FILE_PWD option specifies the password used to protect the transport file (specified by the [TRANSPORT_FILE](#) option).

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	<code>-transport_file_pwd password</code>			✓	✓	✓
Environment Variable	<code>UCRTRANFILEPWD=password</code>			✓	✓	✓

Values

password is the password used to protect the transport file.

VERIFY - Universal Certificate configuration option

Description

The VERIFY option specifies that Universal Certificate is to verify a certificate.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-verify <i>object</i>			✓	✓	✓
Environment Variable	UCRVERIFY= <i>object</i>			✓	✓	✓

Values

object is the specification to verify a certificate.

Valid values for *object* are:

- **cert**
Instructs Universal Certificate to verify an X.509 certificate.

VERSION - Universal Certificate configuration option

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-v			✓	✓	✓
Command Line, Long Form	-version			✓	✓	✓
Environment Variable	n/a					

Values

(There are no values used with this option.)

Universal Control

- [Universal Control Components](#)
- [Universal Control Requests](#)
- [Detailed Information](#)
- [Universal Control Examples](#)

Universal Control Components

Universal Control (UCTL) consists of two components:

- UCTL Manager
- UCTL Server

Universal Control Requests

The UCTL Manager executes one of three different control requests, which is specified via a corresponding configuration option:

1. Start ([START_CMD](#) option)
Starts a component on the specified system.
2. Stop ([STOP_CMD](#) option)
Stops a component on the specified system.
3. Refresh ([REFRESH_CMD](#) option)
Directs Universal Broker on the remote system to refresh the configuration data of all components, including itself, or a single component. Currently, only Universal Event Monitor Server (uems) and Universal Automation Center Agent (uag) are the only individual components that can be refreshed via the REFRESH_CMD option.

One of these control requests must be specified for each execution of Universal Control Manager. Additional input (required and optional) to each execution of the UCTL Manager is made via additional configuration options, which control product behavior and resource allocation for that execution.

Upon execution, UCTL Manager connects to the UCTL Server and processes the request. UCTL Manager registers with a locally running Universal Broker. Consequentially, a Universal Broker must be running in order for a UCTL Manager to execute.

Detailed Information

The following pages provide detailed information for Universal Control.

- [Universal Control Manager for z/OS](#)
- [Universal Control Manager for Windows](#)
- [Universal Control Manager for UNIX](#)
- [Universal Control Manager for IBM i](#)
- [Universal Control Manager Configuration Options](#)
- [Universal Control Server for z/OS](#)
- [Universal Control Server for Windows](#)
- [Universal Control Server for UNIX](#)
- [Universal Control Server for IBM i](#)
- [Universal Control Server Configuration Options](#)
- [Universal Control Component Definition Options](#)
- [Universal Control UACL Entries](#)

Universal Control Examples

See [Refreshing via Universal Control Examples](#) for examples of how to use Universal Control to refresh component configuration data.

See [Starting and Stopping Agent Components - Examples](#) for examples of how to use Universal Control to start and stop Agent components.

Universal Control Manager for zOS

- [Introduction](#)
- [Control Requests](#)
- [JCL Procedure](#)
 - [DD Statements used in JCL Procedure](#)
 - [JCL](#)
- [Configuration](#)
 - [Configuration Sources](#)
 - [Configuration Options Categories](#)
- [Command Line Syntax](#)

Introduction

This Universal Control (UCTL) Manager information is specific to the z/OS operating system.

UCTL Manager for z/OS executes as a batch job.

Control Requests

UCTL Manager for z/OS supports all three Universal Control control requests:

1. Start ([START_CMD](#) option)
2. Stop ([STOP_CMD](#) option)
3. Refresh ([REFRESH_CMD](#) option)

JCL Procedure

The following figure identifies the UCTL Manager for z/OS JCL procedure (**UCTLPRC**, located in the **SUNVSAMP** library) that is provided to simplify the execution JCL and future maintenance.

```
//UCTLPRC  PROC UPARM=,           -- UCTL options
//          UCMDPRE=#SHLQ.UNV
// *
//PS1      EXEC PGM=UCTL, PARM= 'ENVAR(TZ=EST5EDT) /&UPARM '
//STEPLIB  DD  DISP=SHR, DSN=&UCMDPRE..SUNVLOAD
// *
//UNVNLS   DD  DISP=SHR, DSN=&UCMDPRE..SUNVNLS
//UNVTRACE DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
```

DD Statements used in JCL Procedure

The following table describes the DD statements used in the UCTL Manager for z/OS [JCL procedure](#), above.

ddname	DCB Attributes *	Mode	Description
STEPLIB	DSORG=PO, RECFM=U	Input	Load library containing the program being executed.
UNVNLS	DSORG=PO, RECFM=(F, FB, V, VB)	Input	UCTL national language support library. Contains message catalogs and code page translation tables.
UNVTRACE	DSORG=PS, RECFM=(F, FB, V, VB)	Output	UCTL trace output.
SYSPRINT	DSORG=PS, RECFM=(F, FB, V, VB)	Output	stdout file for the UCTL program. UCTL does not write any messages to SYSPRINT.

SYSOUT *	DSORG=PS, RECFM=(F, FB, V, VB)	Output	stderr file for the UCTL program. UCTL writes its messages to SYSOUT.
----------	-----------------------------------	--------	---

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

JCL

The following figure illustrates the UCTL Manager for z/OS JCL using the [UCTLPRCJCL procedure](#), above.

```
//jobname JOB CLASS=A,MSGCLASS=X
//STEP1 EXEC UCTLPRC
//SYSIN DD *
-stop 10312932 -host dallas -userid joe -pwd akkSdiq
/*
```

Job step STEP1 executes **UCTLPRC**.

The configuration options are specified on the SYSIN DD.

Configuration

Configuration consists of:

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

Configuration Sources

Configuration options are read from the following sources:

1. PARM keyword
2. SYSIN ddname
3. Command file ddname
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 PARM keyword override options specified via a SYSIN ddname, and so on.

Detailed information on these methods of configuration can be found in [Configuration Management](#).

Configuration File

The UCTL Manager configuration file is provided to the manager by the local Universal Broker with which it registers. The UCTL Manager configuration file is located in the **UCTCFG00** member of the PDSE allocated to the **UNVCONF** ddname in the Universal Broker started task.

The configuration file, provided by the local Universal Broker, provides the simplest method of specifying configuration options whose values will not change 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 UCTL Manager.

Note



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

Configuration Options Categories

The following table categorizes the configuration options used to execute Universal Control Manager for z/OS 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
Command	Control command to execute.
Remote	Network address of the remote system.

User	User account the Control command executes with on the remote system.
Certificates	X.509 certificate related options.
Events	Options used to define event generation.
Local	Options required for local broker registration.
Messages	Universal Control message options.
Network	Options used to control the process of network data.
Options	Alternative methods to specify command options.
Miscellaneous	Options use to display command help and program versions.

The UCTL Manager configuration options for each category are summarized in the following tables.

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	Location of Manager's PEM-formatted CRL.
PRIVATE_KEY	ddname of Manager's PEM-formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Manager's PRIVATE_KEY.
SSL_IMPLEMENTATION	SSL/TLS implementation to be used for network communications
VERIFY_HOST_NAME	Specification that the Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specification that the Broker's X.509 certificate serial number field must be verified.

Command Category Options

Option Name	Description
COMMAND_ID	Identity of the started component.
REFRESH_CMD	Instruction to a Broker to refresh configuration data.
START_CMD	Instruction to a Broker to start a component.
STOP_CMD	Instruction to stop a component being executed by a Broker.

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 Universal Control 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.

Miscellaneous Category Options

Option Name	Description
HELP	Write command option help.
VERSION	Write program version.

Network Category Options

Option Name	Description
CODE_PAGE	Code page used to translate text data to and from the network.
CTL_SSL_CIPHER_LIST	SSL/TLS cipher list for the control session.
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.

Options Category Options

Option Name	Description
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.

Remote Category Options

Option Name	Description
HOSTNAME_RETRY_COUNT	Number of host connection attempts before ending with an error.
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.

User Category Options

Option Name	Description
USER_ID	User ID or account with which to execute the Control 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 Universal Control Manager for z/OS.

```

uctl
{ -startcompname [-cmdidid] | -stopcompID [-useriduser [-pwdpassword] ] |
-refresh [compname] }
-hosthostaddress
[-fileddname | -encryptedfileddname [-keykey] ] *
[-portport]
[-cmdidid]
[-hostname_retry_countcount]
[-outboundiphost]
[-ssl_implementation {openssl | system}]
[-system_idID]
[-langlanguage]
[-level {trace|audit|info|warn|error}]
[-msg_suppression_listlist ]
[-ca_certsddname [-verify_host_name {yes|no|hostname}]]
[-verify_serial_numbernumber] ]
[-certddname-private_keyddname [-private_key_pwdpassword] ]
[-daysnumber]
[-crlddname]
[-codepagecodepage]
[-ctl_ssl_cipher_listcipherlist]
[-min_ssl_protocoloption]
[-delayseconds]

uctl
{ -help | -version }

```

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

Universal Control Manager for Windows

- [Introduction](#)
- [Command Prompt](#)
- [Universal Configuration Manager](#)
- [Control Requests](#)
- [Configuration](#)
 - [Configuration Sources](#)
 - [Configuration Options Categories](#)
- [Command Line Syntax](#)

Introduction

This Universal Control (UCTL) Manager information is specific to the Windows operating system.

UCTL Manager for Windows is a console application that can be run either from:

- Command prompt
- Universal Configuration Manager

Command Prompt

UCTL Manager runs as a command line program. It provides a command line interface to remote computers running the UCTL Server. On the command line, you must specify the control request that you want the UCTL Manager to execute.

Universal Configuration Manager

The Universal Configuration Manager provides a single interface from which active components can be listed and selected for termination. A list of active components can be obtained from any machine that is running Universal Broker.

Control Requests

UCTL Manager for Windows supports all three Universal Control control requests:

1. Start ([START_CMD](#) option)
2. Stop ([STOP_CMD](#) option)
3. Refresh ([REFRESH_CMD](#) option)

One of these control request options must be specified on the command line (or in a command file) for each execution of Universal Control Manager.

Configuration

Configuration consists of:

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

Configuration Sources

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. The UCTL Manager configuration file is provided to the manager by the local Universal Broker with which it registers.

Detailed information on these methods of configuration can be found in [Configuration Management](#).

Configuration File

The configuration file, **uctl.conf**, provides the simplest method of specifying configuration options whose values will not change 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 UCTL Manager.

Note



For any changes made directly to the UCTL Manager 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 Categories

The following table categorizes configuration options used to execute Universal Control Manager for Windows 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	Control command to execute.
Events	Options used to define event generation.
Installation	Options that specify installation requirements, such as directory locations.
Messages	Universal Control message options.
Miscellaneous	Options use to display command help and program versions.
Network	Options used to control the process of network data.
Options	Alternative methods to specify command options.
Remote	Network address of the remote system.
User	User account the Control command executes with on the remote system.

The Universal Control Manager command options for each of the categories listed above are summarized in the following tables.

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 that the Universal Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specification that the Universal Broker's X.509 certificate serial number field must be verified.

Command Category Options

Option Name	Description
COMMAND_ID	Identity of the started component.
REFRESH_CMD	Instruction to a Broker to refresh configuration data.
START_CMD	Instruction to a Universal Broker to start a component.
STOP_CMD	Instruction to stop a component being executed by a Broker.

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 Universal Control Server 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

Miscellaneous Category Options

Option Name	Description
HELP	Write command option help.
VERSION	Write program version.

Network Category Options

Option Name	Description
CODE_PAGE	Code page used to translate text data to and from the network.
CTL_SSL_CIPHER_LIST	SSL/TLS cipher list for the control session.
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.

Options Category Options

Option Name	Description
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.

Remote Category Options

Option Name	Description
HOSTNAME_RETRY_COUNT	Number of host connection attempts before ending with an error.
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.

User Category Options

Option Name	Description
USER_ID	User ID or account with which to execute the Control command.

USER_PASSWORD

Password associated with USER_ID.

Command Line Syntax

The following figure illustrates the command options syntax — using the command line, long form of the configuration options — of Universal Control Manager for Windows.

```
uctl
{ -startcompname [-cmdidid] | -stopcompID [-useriduser [-pwdpassword] ] |
  -refresh [compname] }
-hosthostaddress
[-filefilename | -encryptedfilefilename [-keykey] ] *
[-portport]
[-cmdidid]
[-hostname_retry_countcount]
[-outboundiphost]
[-langlanguage]
[-level {trace|audit|info|warn|error}]
[-msg_suppression_listlist ]
[-ca_certsfilename [-verify_host_name {yes|no|hostname}]]
[-verify_serial_numbernumber] ]
[-certfilename-private_keyfilename [-private_key_pwdpassword] ]
[-daysnumber]
[-crlfilename]
[-codepagecodepage]
[-ctl_ssl_cipher_listcipherlist]
[-min_ssl_protocoloption]
[-delayseconds]

uctl
{ -help | -version }
```

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

Universal Control Manager for UNIX

- [Introduction](#)
- [Control Requests](#)
- [Configuration](#)
 - [Configuration Sources](#)
 - [Configuration Options Categories](#)
- [Command Line Syntax](#)

Introduction

This Universal Control (UCTL) Manager information is specific to the UNIX operating system.

UCTL Manager for UNIX runs as a command line program. It provides a command line interface to remote computers running the UCTL Server. On the command line, you must specify the control request that you want the UCTL Manager to execute.

Control Requests

UCTL Manager for UNIX supports all three Universal Control control requests:

1. Start ([START_CMD](#) option)
2. Stop ([STOP_CMD](#) option)
3. Refresh ([REFRESH_CMD](#) option)

Configuration

Configuration consists of:

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

Configuration Sources

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. The UCTL Manager configuration file is provided to the manager by the local Universal Broker with which it registers.

Detailed information on these methods of configuration can be found in [Configuration Management](#).

Configuration File

The configuration file, **uctl.conf**, provides the simplest method of specifying configuration options whose values will not change 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 UCTL Manager.

Note



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

Configuration Options Categories

The following table categorizes configuration options used to execute Universal Control Manager for Windows 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	Control command to execute.
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 Control message options.
Miscellaneous	Options use to display command help and program versions.
Network	Options used to control the process of network data.
Options	Alternative methods to specify command options.
Remote	Network address of the remote system.
User	User account the Control command executes with on the remote system.

The Universal Control Manager command options for each of the categories listed above are summarized in the following tables.

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 that the Universal Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specification that the Universal Broker's X.509 certificate serial number field must be verified.

Command Category Options

Option Name	Description
COMMAND_ID	Identity of the started component.
REFRESH_CMD	Instruction to a Broker to refresh configuration data.
START_CMD	Instruction to a Universal Broker to start a component.
STOP_CMD	Instruction to stop a component being executed by a Broker.

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 Universal Control Server 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

Miscellaneous Category Options

Option Name	Description
HELP	Write command option help.
VERSION	Write program version.

Network Category Options

Option Name	Description
CODE_PAGE	Code page used to translate text data to and from the network.
CTL_SSL_CIPHER_LIST	SSL/TLS cipher list for the control session.
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.

Options Category Options

Option Name	Description
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 .

Remote Category Options

Option Name	Description
HOSTNAME_RETRY_COUNT	Number of host connection attempts before ending with an error.
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.

User Category Options

Option Name	Description
USER_ID	User ID or account with which to execute the Control 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 Universal Control Manager for UNIX.

```

uctl
{ -startcompname [-cmdidid] | -stopcompID [-useriduser [-pwdpassword] ] |
-refresh [compname] }
-hosthostaddress
[-filefilename | -encryptedfilefilename [-keykey] ] *
[-portport]
[-cmdidid]
[-hostname_retry_countcount]
[-outboundiphost]
[-bif_directorydirectory]
[-plf_directorydirectory]
[-langlanguage]
[-level {trace|audit|info|warn|error}]
[-msg_suppression_listlist ]
[-ca_certsfilename [-verify_host_name {yes|no|hostname}]
[-verify_serial_numbernumber] ]
[-certfilename-private_keyfilename [-private_key_pwdpassword] ]
[-daysnumber]
[-crlfilename]
[-codepagecodepage]
[-ctl_ssl_cipher_listcipherlist]
[-min_ssl_protocoloption]
[-delayseconds]

uctl
{ -help | -version }

```

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

Universal Control Manager for IBM i

- [Introduction](#)
- [Workload Automation 5 for IBM i Commands](#)
- [Command Execution Environment](#)
- [Control Requests](#)
- [Configuration](#)
 - [Configuration Sources](#)
 - [Configuration Options Categories](#)
- [Command Line Syntax](#)

Currently, IBM i runs Workload Automation 5.1.1. This page provides information for that version.

Introduction

This Universal Control (UCTL) Manager information is specific to the IBM i operating system.

Universal Control Manager for IBM i runs via a command interface. It provides a command line interface to remote computers running the UCTL Server. On the command line, you must specify the control request that you want the UCTL Manager to execute.

This page describes the the Universal Control Manager for IBM i command execution environments, control requests, 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 [Change Release Tag](#) program, **UCHGRLS**, lets you change the tagged command names in **QSYS** to the untagged command names in **UNVPRD511**.

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

Command Execution Environment

The command is valid in all environments:

- Batch input streams
- CL programs
- REXX procedures
- CL ILE modules
- Interactive processing
- Passed to the system program QCMDXEC (or QCAEXEC) for processing

Control Requests

UCTL Manager for IBM i supports all three Universal Control control requests:

1. Start ([START_CMD](#) option)
2. Stop ([STOP_CMD](#) option)
3. Refresh ([REFRESH_CMD](#) option)

Configuration

Configuration consists of:

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

Configuration Sources

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

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

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

Detailed information on these methods of configuration can be found in [Configuration Management](#).

Configuration File

The configuration file, **UNVPRD511/UNVCONF(UCTL)**, provides the simplest method of specifying configuration options whose values will not change 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 UCTL Manager.

Configuration Options Categories

The following table categorizes configuration options used to execute Universal Control Manager for Windows 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	Control command to execute.
Events	Options used to define event generation.
Local	Options required for local broker registration.
Messages	Universal Control message options.
Miscellaneous	Options use to display command help and program versions.
Network	Options used to control the process of network data.
Options	Alternative methods to specify command options.
Remote	Network address of the remote system.
User	User account the Control command executes with on the remote system.

The Universal Control Manager command options for each of the categories listed above are summarized in the following tables.

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 that the Universal Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specification that the Universal Broker's X.509 certificate serial number field must be verified.

Command Category Options

Option Name	Description
COMMAND_ID	Identity of the started component.
REFRESH_CMD	Instruction to a Broker to refresh configuration data.

START_CMD	Instruction to a Universal Broker to start a component.
STOP_CMD	Instruction to stop a component being executed by a Broker.

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.

Miscellaneous Category Options

Option Name	Description
VERSION	Write program version.

Network Category Options

Option Name	Description
CODE_PAGE	Code page used to translate text data to and from the network.
CTL_SSL_CIPHER_LIST	SSL/TLS cipher list for the control session.
MIN_SSL_PROTOCOL	Minimum SSL/TLS protocol level that will be negotiated and used for communications channels. <div style="display: flex; align-items: center;"> <p>Note This option was introduced to IBM i in version 5.1.1.0.</p> </div>
NETWORK_DELAY	Maximum number of seconds considered acceptable to wait for data communications.

Options Category Options

Option Name	Description
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 .

Remote Category Options

Option Name	Description
HOSTNAME_RETRY_COUNT	Number of host connection attempts before ending with an error.
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.
-------------	--

User Category Options

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

Command Line Syntax

The syntax shows the CL Command parameter followed by UNIX/CALL options (in parentheses). These options would be used to invoke Universal Command Manager on a different platform. They are provided to help the user associate STRUCT command options with UCTL command line options on other platforms.

The following figure illustrates the command line syntax — using the STRUCT parameter form of the configuration options — of Universal Control Manager for Windows.

```
STRUCT
{ START(compname) [CMDID(id)] | STOP(compID) [USER(user) [PWD(password)] ] | REFRESH ({yes|no}) [RFSHCMPNM(compname)] }
HOST(hostaddress)
[CMDFILE(filename) [CMDMBR(member)] | ECMFILE(filename) [ECMMBR(member)] [KEY(key)] ]
[PORT(port)]
[CMDID(id)]
[HSTNMRTYCT(count)]
[OUTBOUNDIP(host)]
[MSGLANG(language)]
[MSGLEVEL(*{trace|audit|info|warn|error})]
[CACERTS(file [lib]) [VFYHSTNM({yes|no|hostname})] [VFYSERNUM(number)] ]
[CERT(file [lib]) PVTKEYF(file [lib]) [PVTKEYPWD(password)] ]
[CRLFILE(file [lib]) [CRLMBR(member)] ]
[CODEPAGE(codepage)]
[CTLCPHRLST(cipherlist)]
[MINSSLPROTOCOL(option)]
[DELAY(seconds)]
[PLFDIR(directory)]

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

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

Universal Control Manager Configuration Options


Universal Control Manager Configuration Options

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

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

Configuration Options List

The following table identifies all Universal Control configuration options.

Option	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
BIF_DIRECTORY	Broker Interface Directory that specifies the location of the Universal Broker interface file.
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	Location of Manager's PEM-formatted CRL.
CODE_PAGE	Code page used to translate text data to and from the network.
COMMAND_FILE_ENCRYPTED	Encrypted command file.
COMMAND_FILE_PLAIN	Plain text command file.
COMMAND_ID	Identity of a started component.
CTL_SSL_CIPHER_LIST	SSL/TLS cipher list for the control session.
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.
HELP	Displays a description of the command line options and their format.
HOSTNAME_RETRY_COUNT	Number of host connection attempts before ending with an error.
INSTALLATION_DIRECTORY	Directory in which the product is installed.
MESSAGE_LANGUAGE	Language of messages formatted.
MESSAGE_LEVEL	Level of messages written.
MSG_SUPPRESSION_LIST	List of message IDs representing Universal messages to be suppressed.
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.
NLS_DIRECTORY	NLS directory.
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 Universal Control 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 .
REFRESH_CMD	Instruct a Broker or component to refresh its configuration.

REMOTE_HOST	TCP/IP host name of the remote computer on which Universal Broker is running and accepting connections.
REMOTE_PORT	TCP/IP port number of the remote computer on which Universal Broker is running and accepting connections.
SAF_KEY_RING	SAF certificate key ring name.
SAF_KEY_RING_LABEL	SAF key ring certificate label.
SSL_IMPLEMENTATION	SSL/TLS implementation to be used for network communications.
START_CMD	Instruction to a Universal Broker to start a component.
STOP_CMD	Instruction to stop a component being executed by a Broker.
SYSTEM_ID	Local Universal Broker with which the Universal Control Manager must register.
USER_ID	User ID or account with which to execute the Control command.
USER_PASSWORD	Password associated with USER_ID.
VERIFY_HOST_NAME	Specification that the Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specification that the Broker's X.509 certificate serial number field must be verified.
VERSION	Write program version.

ACTIVITY_MONITORING - UCTL 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, Short Form	n/a				
Command Line, Long Form	n/a				
Environment Variable	n/a				
Configuration File Keyword	activity_monitoring <i>option</i>	✓	✓	✓	✓
STRUCT Parameter	n/a				

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.

BIF_DIRECTORY - UCTL 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 <i>directory</i>		✓		
Environment Variable	UCTLBIFDIRECTORY= <i>directory</i>		✓		
Configuration File Keyword	n/a				
STRUCT Parameter	n/a				

Values

directory is the name of the BIF directory.

Default is `/var/opt/universal`.

CA_CERTIFICATES - UCTL 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 <i>ddname</i> or <i>file</i>		✓	✓	✓
Environment Variable	UCTLCACERTS= <i>file</i>	✓	✓	✓	
Configuration File Keyword	ca_certificates <i>ddname</i> or <i>file</i>	✓	✓	✓	✓
STRUCT Parameter	CACERTS (<i>file</i> [<i>lib</i>])	✓			

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 and 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 also can be qualified by a library name (*lib*). If it is not, the library list ***LIBL** is searched for the first occurrence of the file name.

CERTIFICATE - UCTL Manager configuration option

Description

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

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

Note



If the CERTIFICATE option is used, the [PRIVATE_KEY](#) option also is required.

Usage

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

Values

IBM i	<i>file</i> is the qualified file name of the X.509 certificate file. The file name can be qualified by a library name.
UNIX	<i>file</i> is the path name of the X.509 certificate file. Relative paths are relative to the current working directory.
Windows	<i>file</i> is the path name of the X.509 certificate file. Relative paths are relative to the current working directory.
z/OS	<i>ddname</i> 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.

CERTIFICATE_EXPIRATION_NOTICE - UCTL 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 <i>number</i>		✓	✓	✓
Environment Variable	UCTLDAYSTILEXPire= <i>number</i>		✓	✓	
Configuration File Keyword	days_til_expire <i>number</i>		✓	✓	✓
STRUCT 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 - UCTL Manager configuration option

Description

The CERTIFICATE_REVOCATION_LIST option specifies the file / 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 <i>ddname</i> or <i>file</i>		✓	✓	✓
Environment Variable	UCTLCRL= <i>file</i>	✓	✓	✓	
Configuration File Keyword	crl <i>ddname</i> or <i>file</i>	✓	✓	✓	✓
STRUCT Parameter	CRLFILE(<i>file{lib}</i>) [CRLMBR(<i>member</i>)]	✓			

Values

IBM i	<i>file</i> 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.
UNIX	<i>file</i> is the path name of the file containing the CRL. Relative paths are relative to the current working directory.
Windows	<i>file</i> is the path name of the file containing the CRL. Relative paths are relative to the current working directory.
z/OS	<i>ddname</i> is the ddname of the file containing the CRL. The value is used only when the SSL_IMPLEMENTATION option is set to openssl .

CODE_PAGE - UCTL 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.

Usage

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

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**.)

Default

The default code page 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_FILE_ENCRYPTED - UCTL Manager configuration option

Description

The `COMMAND_FILE_ENCRYPTED` option specifies the data set (for z/OS) or file containing encrypted values for command line option parameters.

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.

Universal Control 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. If a key was used to encrypt the file, the same key must be supplied using the [ENCRYPTION_KEY](#) option.

Note



If a data set / file is specified in this option, it should not be specified additionally in the [COMMAND_FILE_PLAIN](#) option. If it is, the data set / file specified in [COMMAND_FILE_PLAIN](#) will be used.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	<code>-x ddname or filename</code>		✓	✓	✓
Command Line, Long Form	<code>-encryptedfile ddname or filename</code>		✓	✓	✓
Environment Variable	<code>UCTLENCRYPTEDFILE=filename</code>	✓	✓	✓	
Configuration File Keyword	n/a				
STRUCT Parameter	<code>ECMFILE(filename) [UTIL:ECMMBR(member)]</code>	✓			

Value

`ddname` or `filename` is the name of the data set or file, respectively, containing the encrypted command parameter values.

COMMAND_FILE_PLAIN - UCTL Manager configuration option

Description

The `COMMAND_FILE_PLAIN` option specifies the data set (for z/OS) or file containing plain text values for command line option parameters.

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.

Universal Control 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 data set / file is specified in this option, it should not be specified additionally in the [COMMAND_FILE_ENCRYPTED](#) option. If it is, the data set / file specified in `COMMAND_FILE_PLAIN` will be used.

Usage

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

Value

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

COMMAND_ID - UCTL Manager configuration option

Description

The COMMAND_ID option identifies a started Universal Automation Center component.

If the COMMAND_ID option is not specified, the component name is used.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-C <i>id</i>		✓	✓	✓
Command Line, Long Form	-cmdid <i>id</i>		✓	✓	✓
Environment Variable	UCTLCMDID <i>id</i>	✓	✓	✓	
Configuration File Keyword	n/a				
STRUCT Parameter	CMDID(<i>id</i>)	✓			

Value

id is any value that identifies the component.

It is used primarily for recognition of components started by Universal Control.

IBM i



If *id* contains non-alphanumeric characters including spaces, it must be enclosed in single (') quotation marks. If a single (') quotation mark is part of the command, enter two single (') quotation marks to represent one.

CTL_SSL_CIPHER_LIST - UCTL 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.


Usage

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

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	<p>128-bit DES encryption with SHA-1 message digest.</p> <p>Note  As of Universal Agent 6.7.0.0, DES-CBC-SHA is supported only on HP-UX.</p> <p>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.</p>

ENCRYPTION_KEY - UCTL Manager configuration option

Description

The ENCRYPTION_KEY option specifies the key used to encrypt the command file.

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	UCTLKEY= <i>key</i>				
Configuration File Keyword	n/a				
STRUCT Parameter	KEY(<i>key</i>)	✓			

Value

key is the key used to encrypt the command file.

EVENT_GENERATION - UCTL 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	<code>event_generation types</code>	✓	✓	✓	✓
STRUCT 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:

- Inclusion operator is an asterisk (*).
- Exclusion operator is **X** or **x**.

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).

HELP - UCTL Manager configuration option

Description

The HELP option displays a description of the Universal Control 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				
STRUCT Parameter	n/a				

Value

(There are no values for the HELP option.)

HOSTNAME_RETRY_COUNT - UCTL Manager configuration option

Description

The `HOSTNAME_RETRY_COUNT` option specifies the number of times that the Universal Control Manager will attempt to establish a connection with a specified Universal Broker before it ends with a connect error.

The Universal Control Manager will sleep for one second between connection 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 the Universal Control 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	<code>-hostname_retry_count count</code>		✓	✓	✓
Environment Variable	<code>UCTLHOSTNAMERETRYCOUNT=count</code>	✓	✓	✓	
Configuration File Keyword	<code>hostname_retry_count count</code>	✓	✓	✓	✓
STRUCT Parameter	<code>HSTNMRTYCT(count)</code>	✓			

Value

count is the number of times that Universal Control will attempt to establish a connection.

Default is 1.

INSTALLATION_DIRECTORY - UCTL Manager configuration option

Description

The `INSTALLATION_DIRECTORY` option specifies the directory in which the Universal Control Manager is installed.

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	<code>installation_directory</code> <i>directory</i>		✓	✓	
STRUCT Parameter	n/a				

Value

directory is the directory in which the Universal Control Manager is installed.

Default

UNIX	<code>/opt/universal/uctlmgr</code>
Windows	<code>c:\Program Files\Universal\uctlmgr</code>

MESSAGE_LANGUAGE - UCTL Manager configuration option

Description

The MESSAGE_LANGUAGE 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 <i>language</i>		✓	✓	✓
Command Line, Long Form	-lang <i>language</i>		✓	✓	✓
Environment Variable	UCTLLANG= <i>language</i>	✓	✓	✓	✓
Configuration File Keyword	language <i>language</i>	✓	✓	✓	✓
STRUCT Parameter	MSGLANG(<i>language</i>)	✓			

Values

language is any UMC file provided by Stonebranch Inc.

There are different UMC files for different languages.

IBM i	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	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	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	The first three characters of the language name are used as a three-character suffix in the UMC member name. UMC files are read from the partitioned data set allocated on ddname UNVNLS. Universal Control message catalog member names start with UCTMC.

Default is ENGLISH.

MESSAGE_LEVEL - UCTL Manager configuration option

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-l <i>level</i>		✓	✓	✓
Command Line, Long Form	-level <i>level</i>		✓	✓	✓
Environment Variable	UCTLLEVEL= <i>level</i>	✓	✓	✓	✓
Configuration File Keyword	message_level <i>level</i>	✓	✓	✓	✓
STRUCT Parameter	MSGLEVEL(* <i>level</i>)	✓			

Values

level indicates either of the following level of messages:

- **trace**
Writes trace messages used for diagnostic purposes (see [Trace Files](#), below).

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

IBM i	info
UNIX	warn
Windows	warn
z/OS	info

Trace Files

IBM i	Trace file name is *CURLIB/UNVTRCUCT(UCTn), where n is the job number of the job invoking Universal Control.
UNIX	Trace file name is uctl.trc . It is created in the working directory of Universal Control Manager.
Windows	Trace file name is uctl.trc . It is created in the working directory of Universal Control Manager.
z/OS	Trace file is written to the data set referenced by the UNVTRACE ddname.

MIN_SSL_PROTOCOL - UCTL 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 <i>option</i>		✓	✓	✓
Environment Variable	UCTLMINSSLPROTOCOL <i>option</i>	✓	✓	✓	
Configuration File Keyword	min_ssl_protocol <i>option</i>	✓	✓	✓	✓
STRUCT Parameter	MINSSLPROTOCOL <i>option</i>	✓			

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 - UCTL 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 <i>list</i>		✓	✓	✓
Environment Variable	UCTLMSGSUPPRESSIONLIST= <i>list</i>		✓	✓	✓
Configuration File Keyword	msg_suppression_list <i>list</i>		✓	✓	✓
STRUCT Parameter	n/a				

Values

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

NETWORK_DELAY - UCTL Manager configuration option

Description

The NETWORK_DELAY option specifies the maximum acceptable delay in transmitting data over the network between the Universal Control Manager and Universal Control 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 Control'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 Control times out waiting for a packet to arrive in the period of time specified by NETWORK_DELAY.

Note



An understanding of the TCP/IP protocol and the network configuration between the Universal Control Manager and Universal Control Server is required in order to determine the appropriate delay value.

Usage

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

Values

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

Default is 120.

NLS_DIRECTORY - UCTL Manager configuration option

Description

The NLS_DIRECTORY option specifies the directory in which Universal Agent NLS files are installed.

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 <i>directory</i>		✓	✓	
STRUCT Parameter	n/a				

Values

directory is the directory in which NLS files are installed.

Default

UNIX	<code>/opt/universal/nls</code>
Windows	<code>..\nls</code>

OUTBOUND_IP - UCTL Manager configuration option

Description

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

Note

 By default, the OUTBOUND_IP option is not set.

Usage

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

Values

host is the host or IP address.

PLF_DIRECTORY - UCTL 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 Universal Control Manager process to store manager process termination information for the Universal Broker.

IBM i



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

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-plf_directory <i>directory</i>		✓		
Environment Variable	UCTLPLFDIRECTORY= <i>directory</i>		✓		
Configuration File Keyword	n/a				
STRUCT Parameter	PLFDIR(<i>directory</i>)	✓			

Values

directory is the name of the PLF directory.

A full path name must be specified.

UNIX



Default = /var/opt/universal/tmp.

IBM i




Default = /tmp.

PRIVATE_KEY - UCTL Manager configuration option

Description

The PRIVATE_KEY option specifies the location of the PEM-formatted RSA private key that corresponds to the X.509 certificates 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 <i>ddname</i> or <i>file</i>		✓	✓	✓
Environment Variable	UCTLPRIVATEKEY= <i>file</i>	✓	✓	✓	
Configuration File Keyword	private_key <i>ddname</i> or <i>file</i>	✓	✓	✓	✓
STRUCT Parameter	PVTKEYF(<i>file</i> [<i>lib</i>])	✓			

Values


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

PRIVATE_KEY_PWD - UCTL 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	<code>-private_key_pwd password</code>		✓	✓	✓
Environment Variable	<code>UCTLPRIVATEKEYPWD=password</code>	✓	✓	✓	
Configuration File Keyword	<code>privatekeypwd=password</code>	✓	✓	✓	✓
STRUCT Parameter	<code>PVTKEYPWD(password)</code>	✓			

Values

password is the password for the private key.

REFRESH_CMD - UCTL Manager configuration option

Description

The REFRESH_CMD option directs Universal Broker to refresh the configuration data that it maintains for all components, including itself, or a single, specified component type.

A REFRESH_CMD option that does not specify a component tells Universal Broker to refresh the configuration data that it maintains for all components. The Broker will reread the configuration data and refresh its copy that it keeps in memory for all components.

A REFRESH_CMD option that specifies a component type tells Universal Broker to refresh its copy of the configuration data for that component type and forward the option to components of that type.

Note



Currently, Universal Event Monitor Server (uems) and Universal Automation Center Agent (uag) are the only individual components that can be refreshed via the REFRESH_CMD option.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-R [<i>compname</i>]		✓	✓	✓
Command Line, Long Form	-refresh [<i>compname</i>]		✓	✓	✓
Environment Variable	n/a				
Configuration File Keyword	n/a				
STRUCT Parameter	REFRESH(* <i>option</i>) [RFSHCMPNM(<i>compname</i>)]	✓			

Values

compname is the name of the component to which Universal Broker forwards this option.

Currently, only *uems* (Universal Event Monitor Server) and *uag* (Universal Automation Center Agent) are valid values for *compname*.

If *compname* is not specified, Universal Broker refreshes the configuration data for all components (including itself).

IBM i



Valid values for *option* are:

- **yes**
Refresh the configuration data.
- **no**
Do not refresh the configuration data.

REMOTE_HOST - UCTL Manager configuration option

Description

The REMOTE_HOST option specifies the IP address or host name of the Universal Broker on the remote computer on which to run the command.

The remote computer must have a Universal Broker running and accepting connections.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-i <i>hostaddress</i>		✓	✓	✓
Command Line, Long Form	-host <i>hostaddress</i>		✓	✓	✓
Environment Variable	UCTLHOST <i>hostaddress</i>	✓	✓	✓	✓
Configuration File Keyword	host <i>hostaddress</i>	✓	✓	✓	✓
STRUCT Parameter	HOST(<i>hostaddress</i>)	✓			

Values

hostaddress is the IP address of the host computer.

The format of *hostaddress* can be either:

- IP address in dotted form (for example, 1.2.3.4)
- Host name (for example, *dallas*).

REMOTE_PORT - UCTL 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 port number.

Usage

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

Values

port is the TCP port on the remote computer.

The format of *port* can be either:

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

Default is 7887.

SAF_KEY_RING - UCTL Manager configuration option

Description

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

The key ring must be associated with the user profile with which Universal Control 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 <i>name</i>				✓
Environment Variable	n/a				
Configuration File Keyword	saf_key_ring <i>name</i>				✓
STRUCT Parameter	n/a				

Values

name is the name of the SAF certificate key ring.

SAF_KEY_RING_LABEL - UCTL 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 Universal Control 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 <i>label</i>				✓
Environment Variable	n/a				
Configuration File Keyword	saf_key_ring_label <i>label</i>				✓
STRUCT Parameter	n/a				

Values

label is the label of the SAF certificate key ring.

Default is default certificate in the key ring.

SSL_IMPLEMENTATION - UCTL 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 <i>option</i>				✓
Environment Variable	UCTLSSLIMPLEMENTATION= <i>option</i>				✓
Configuration File Keyword	ssl_implementation <i>option</i>				✓
STRUCT 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 7.1.x Installation, Upgrade, and Applying Maintenance](#) for a description of the prerequisites before using System SSL.)

Default is openssl.

START_CMD - UCTL Manager configuration option

Description

The `START_CMD` option specifies the name of the Universal Agent component that is to be started on the machine specified by the `REMOTE_HOST` option.

Only those Universal Agent server components that do not require interaction with a Manager application can be started with Universal Control. The Universal Broker checks the requested component's type against a list of component types that can be started by Universal Control. The Universal Broker will reject any attempt to start an ineligible component.

In Universal Agent 7.1.x, the only components that Universal Control can start are:

- Event-driven Universal Event Monitor (UEM) Server (component type `uems`)
- Universal Automation Center Agent (UAG) Server (component type `uags`)
- Universal Message Service (OMS) Server (component type `oms`)

If the Universal Control Server is configured with security enabled, a user identifier and password will be required to start a component. The component, once started, will run under the same security context of the Universal Broker.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	<code>-start compname</code>		✓	✓	✓
Environment Variable	<code>UCTLSTART=compname</code>	✓	✓	✓	✓
Configuration File Keyword	n/a				
STRUCT Parameter	<code>START(compname)</code>	✓			

Values

`compname` is the name of the component to be started.

STOP_CMD - UCTL Manager configuration option

Description

The STOP_CMD option specifies the ID of the component that is to be terminated on the remote computer.

A component ID is assigned to a component when Universal Broker starts it. The Universal Query utility can be used to obtain a list of active components (along with their component IDs) that are managed by an instance of Universal Broker. (See [Universal Query](#) for information about Universal Query.)

STOP_CMD can require the user identifier and password of the user account with which the component to be stopped is executing. If so, the user identifier must be the same as the user identifier with which the component is executing; otherwise, STOP_CMD will fail. The user ID and password are specified with the [USER_ID](#) and [USER_PASSWORD](#) options.

Whether or not STOP_CMD requires a user ID and password depends on the Universal Control Server configuration.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-s <i>compID</i>		✓	✓	✓
Command Line, Long Form	-stop <i>compID</i>		✓	✓	✓
Environment Variable	UCTLSTOP= <i>compID</i>	✓	✓	✓	✓
Configuration File Keyword	n/a				
STRUCT Parameter	STOP(<i>compID</i>)	✓			

Values

compID is the ID of the component to be terminated.

SYSTEM_ID - UCTL Manager configuration option

Description

The SYSTEM_ID option identifies the local Universal Broker with which the Universal Control 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 <i>ID</i>				✓
Environment Variable	UCTLSYSTEMID= <i>ID</i>				✓
Configuration File Keyword	n/a				
STRUCT 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.)

USER_ID - UCTL Manager configuration option

Description

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

The Universal Control Server determines if this option is required.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-u <i>user</i>		✓	✓	✓
Command Line, Long Form	-userid <i>user</i>		✓	✓	✓
Environment Variable	UCTLUSERID= <i>user</i>	✓	✓	✓	✓
Configuration File Keyword	userid <i>user</i>	✓	✓	✓	✓
STRUCT Parameter	USER(<i>user</i>)	✓			

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.

USER_PASSWORD - UCTL 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, regardless of how encryption is configured on the Universal Control Manager and Universal Control Server.

Usage

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

Values

password is the password for the user identifier.

Note



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

VERIFY_HOST_NAME - UCTL 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 s*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 Universal Control Manager terminates.

Usage

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

Values

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

Valid values for *option* are:

- **yes**
Certificate identity is verified using the host name specified by the [REMOTE_HOST](#) option.
- **no**
Certificate identity is not verified.
- *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 - UCTL 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 Universal Control Manager terminates.

Usage

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

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 - UCTL 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				
STRUCT Parameter	VERSION(* <i>option</i>)	✓			

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 Control Server for zOS

- [Introduction](#)
- [Environment](#)
- [User Identification](#)
- [Additional Information](#)

Introduction

This Universal Control (UCTL) Server information is specific to the z/OS operating system.

Environment

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

As with all components managed by the Universal Broker, UCTL Server inherits the message language from the Universal Broker. All messages generated by the Universal Control Server are sent to Universal Broker for processing.

User Identification

UCTL Server can operate with user security active or inactive, based on the [USER_SECURITY](#) configuration option.

- With user security active, the UCTL Server requires the UCTL Manager to supply a valid z/OS user ID and a password. The user profile must have a properly defined OMVS segment.
- With user security inactive, the UCTL Server does not require the UCTL Manager to supply a valid user ID. Essentially, any operation that the UCTL Server is capable of can be requested by any UCTL Manager.

Additional Information

The following pages provide additional detailed information for Universal Control Server for z/OS:

- [Universal Control Server for z/OS - Component Definition](#)
- [Universal Control Server for z/OS - Configuration](#)
- [Universal Control Server for z/OS - Universal Access Control List](#)

Universal Control Server for zOS - 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. (For details on how Universal Broker manages components, see [Universal Broker 7.1.x Reference Guide](#).)

The syntax of a component definition file is the same as a configuration file.

The UCTL Server for z/OS component definition is located in the component definition library **UNVCOMP** allocated to the Universal Broker ddname **UNVCOMP**. The UCTL Server component definition member is **UTSCMP00**.

Component Definition Options

The following table identifies all of the options that comprise the UCTL 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 UCTL Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCTL Server,
CONFIGURATION_FILE	Name of the UCTL Server's configuration file,
RUNNING_MAXIMUM	Maximum number of UCTL Servers that can run simultaneously,
START_COMMAND	Member name of the UCTL Server program,
WORKING_DIRECTORY	Directory used as the working directory of the UCTL Server,

Universal Control Server for zOS - Configuration

- [Overview](#)
- [Configuration File](#)
- [Configuration Option](#)

Overview

UCTL Server configuration consists of defining runtime and default values.

Configuration File

The configuration file provides the simplest method of specifying configuration values that will not change with each command invocation. This file can be edited manually with any text editor.

The UCTL Server configuration file name is specified in the Universal Control Server component definition. The default name is **UTSCFG00**. The name refers to a member in the PDS allocated to the Universal Broker ddname **UNVCONF**.

Note



In order for any changes to the UCTL Server configuration file to become active, either:

- A Universal Broker refresh is required.
- The Universal Broker started task must be restarted.

Configuration Option

The following table identifies all of the UCTL 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.
CODE_PAGE	Code page used for text translation.
EVENT_GENERATION	Events to be generated as persistent events.
MESSAGE_LEVEL	Level of messages written.
MSG_SUPPRESSION_LIST	List of message IDs representing Universal messages to be suppressed.
TMP_DIRECTORY	HFS directory name used for temporary files.
USER_SECURITY	Specification for whether or not user authentication is active.

Universal Control Server for zOS - Universal Access Control List

Overview

The UCTL Server uses the Universal Access Control List (UACL) file as an extra layer of security. The UACL file contains entries for the Universal Control Server. These entries contain Access Control List (ACL) rules that permit or deny access to the UCTL Server.

UACL Entries

The syntax of a UACL entry file is the same as the UCTL configuration file.

The following table identifies all UCTL Server for z/OS UACL entries. Each **UACL Entry Name** is a link to detailed information about that option.

UACL Entry Name	Description
UCTL_ACCESS	Allows or denies access to UCTL Server services,
UCTL_REQUEST	Allows or denies access to UCTL Server services based on client identification and request type,

Universal Control Server for Windows

- [Introduction](#)
- [Environment](#)
- [User Identification](#)
- [Additional Information](#)

Introduction

This Universal Control (UCTL) Server information is specific to the Windows family of operating systems.

Environment

The UCTL Server runs as a background process. It does not interact with a console or desktop.

As with all components managed by the Universal Broker, UCTL Server inherits the message language from the Universal Broker. All messages generated by the UCTL Server are sent to Universal Broker for processing.

User Identification

UCTL Server can operate with user security active or inactive, based on the [USER_SECURITY](#) configuration option.

- With user security active, the UCTL Server requires the UCTL Manager to supply a valid user ID for the local system and a password.
- With user security inactive, the UCTL Server does not require the UCTL Manager to supply a valid user ID. Essentially, any operation that the UCTL Server is capable of can be requested by any UCTL Manager.

Additional Information

The following pages provide additional detailed information for Universal Control Server for Windows:

- [Universal Control Server for Windows - Component Definition](#)
- [Universal Control Server for Windows - Configuration](#)
- [Universal Control Server for Windows - Universal Access Control List](#)

Universal Control Server for Windows - Component Definition

- [Overview](#)
- [Universal Configuration Manager](#)
- [Component Definition Options](#)

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. (For details on how Universal Broker manages components, see the Universal Broker Reference Guide.)

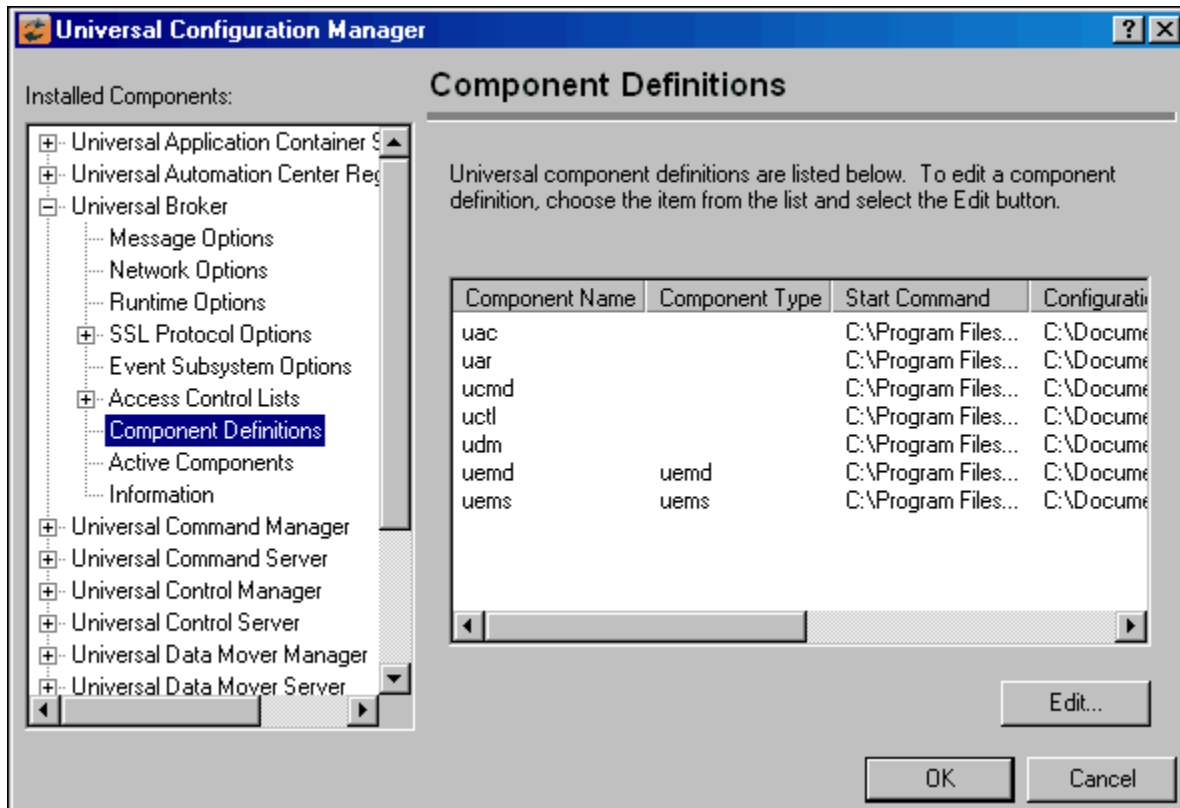
The syntax of a component definition file is the same as a configuration file.

Universal Configuration Manager

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 (see below).



Component Definition Options

The following table identifies all of the options that comprise the UCTL 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 UCTL Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCTL Server.
CONFIGURATION_FILE	Name of the UCTL Server configuration file.
RUNNING_MAXIMUM	Maximum number of UCTL Servers that can run simultaneously.
START_COMMAND	Full path name of the UCTL Server program.
WORKING_DIRECTORY	Directory used as the working directory of the UCTL Server.

Universal Control Server for Windows - Configuration

- [Overview](#)
- [Configuration File](#)
- [Configuration Options](#)

Overview

UCTL Server configuration consists of defining run-time and default values. This page describes the Server configuration options.

Configuration File

The configuration file provides a simple method of specifying configuration values that will not change with each command invocation. This file can be edited manually with any text editor (for example, Notepad).

However, the [Universal Configuration Manager](#) application, accessible via the Control Panel, is the recommended way to set Windows 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.

The UCTL Server configuration file name is specified in the Universal Control Server component definition.

Note



In order for any changes made directly to the UCTL Server configuration file to become active, either:

1. A Universal Broker refresh is required.
2. The Universal Broker service must be restarted.

However, 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 of the UCTL 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.
CODE_PAGE	Code page used for text translation.
EVENT_GENERATION	Events to be generated as persistent events.
INSTALLATION_DIRECTORY	Base directory in which Universal Control Server is installed.
LOGON_METHOD	Method of how users are logged onto the system.
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.
TMP_DIRECTORY	Directory name used for temporary files.
USER_SECURITY	Specification for whether or not user authentication is active.

Universal Control Server for Windows - Universal Access Control List

Overview

The Universal Control Server uses the Universal Access Control List (UACL) file as an extra layer of security. The UACL file contains entries for the Universal Control Server. These entries contain Access Control List (ACL) rules that permit or deny access to the Universal Control Server.

UACL Entries

The syntax of a UACL file is the same as the Universal Control configuration file.

The following table identifies all UCTL Server for Windows UACL entries. Each **UACL Entry Name** is a link to detailed information about that option.

UACL Entry Name	Description
UCTL_ACCESS	Allows or denies access to Universal Control Server services.
UCTL_REQUEST	Allows or denies access to Universal Control Server services based on client identification and request type.

Universal Control Server for UNIX

- [Introduction](#)
- [Environment](#)
- [User Identification](#)
- [Additional Information](#)

Introduction

This Universal Control (UCTL) Server information is specific to the UNIX operating system.

Environment

The Universal Control Server runs as a background process. It does not interact with a console.

As with all components managed by the Universal Broker, Universal Control Server inherits the message language from the Universal Broker. All messages generated by the Universal Control Server are sent to Universal Broker for processing.

User Identification

Universal Control Server can operate with user security active or inactive, based on the [USER_SECURITY](#) configuration option.

- With user security active, the Universal Control Server requires the Universal Control Manager to supply a valid user ID for the local system and a password.
- With user security inactive, the Universal Control Server does not require the Manager to supply a valid user ID. Essentially, any operation the Control Server is capable can be requested by any Control Manager.

Additional Information

The following pages provide additional detailed information for Universal Control Server for UNIX:

- [Universal Control Server for UNIX - Component Definition](#)
- [Universal Control Server for UNIX - Configuration](#)
- [Universal Control Server for UNIX - Universal Access Control List](#)

Universal Control 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. (For details on how Universal Broker manages components, see the [Universal Broker 7.1.x Reference Guide](#).)

The syntax of a component definition file is the same as a configuration file.

Component Definition Options

The following table identifies all of the options that comprise the UCTL 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 UCTL Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCTL Server.
CONFIGURATION_FILE	Name of the UCTL Server configuration file.
RUNNING_MAXIMUM	Maximum number of UCTL Servers that can run simultaneously.
START_COMMAND	Full path name of the UCTL Server program.
WORKING_DIRECTORY	Directory used as the working directory of the UCTL Server.

Universal Control Server for UNIX - Configuration

- [Overview](#)
- [Configuration File](#)
- [Configuration Options](#)

Overview

Universal Control Server configuration consists of defining runtime and default values. This page describes the Server configuration options.

Configuration File

The configuration file provides the simplest method of specifying configuration values that will not change with each command invocation. This file can be edited manually with any text editor.

The Universal Control Server configuration file name is specified in the Universal Control Server component definition. The default name is **uctls.conf**. Refer to the component definition file to determine the directory in which it is located.

Note



In order for any changes made directly to the UCTL Server configuration file to become active, either:

- A Universal Broker refresh is required.
- The Universal Broker service must be restarted.

Configuration Options

The following table identifies all of the Universal Control 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.
CODE_PAGE	Code page used for text translation.
EVENT_GENERATION	Events to be generated as persistent events.
INSTALLATION_DIRECTORY	Base directory in which Universal Control Server is installed.
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.
TMP_DIRECTORY	Directory name used for temporary files.
TRACE_DIRECTORY	Location of trace files.
USER_SECURITY	Specification for whether or not user authentication is active.

Universal Control Server for UNIX - Universal Access Control List

Overview

The Universal Control Server uses the Universal Access Control List (UACL) file as an extra layer of security. The UACL file contains entries for the Universal Control Server. These entries contain Access Control List (ACL) rules that permit or deny access to the Universal Control Server.

UACL Entries

The syntax of a UACL entry file is the same as the Universal Control configuration file.

The following table identifies all UCTL Server for UNIX UACL entries. Each **UACL Entry Name** is a link to detailed information about that option.

UACL Entry Name	Description
UCTL_ACCESS	Allows or denies access to Universal Control Server services.
UCTL_REQUEST	Allows or denies access to Universal Control Server services based on client identification and request type.

Universal Control Server for IBM i

- [Introduction](#)
- [Environment](#)
- [User Identification](#)
- [Current Library and Working Directory](#)
- [Additional Information](#)

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

Introduction

This Universal Control (UCTL) Server information is specific to the IBM i operating system.

Environment

The Universal Control Server runs under the **UNVUBR511** subsystem's pre-start job **UNVSRV**. When Universal Broker receives a request for a Universal Command component, it passes the request to the **UCTSRV** program running under the **UNVSRV** pre-start job.

As with all components managed by the Universal Broker, Universal Control Server inherits the message language from the Universal Broker. All messages generated by the Universal Control Server are sent to Universal Broker for processing.

User Identification

Universal Control Server can operate with user security active or inactive, based on the [USER_SECURITY](#) configuration option.

- With user security active, the Server requires the 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 Server does not require the Manager to supply a valid user ID. The user command executes with the user profile of the Server. The user profile of the Server is inherited from the Broker. The inherited profile is **UNVUBR511**; as installed, this profile provides a very high level of authority including *ALLOBJ.

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 is designated by the user profile and the working directory is the home directory of the user profile.
- With user security inactive, the current library is that for the user profile (the installation default, **UNVUBR511**) associated with the service program. Note that the default value used for the current library is **UNVTMP511**. Care should be taken to avoid name clashes and other consequences of multiple processes sharing a common current library and working directory.

Additional Information

The following pages provide additional detailed information for Universal Control Server for IBM i:

- [Universal Control Server for IBM i - Component Definition](#)
- [Universal Control Server for IBM i - Configuration](#)
- [Universal Control Server for IBM i - Universal Access Control List](#)

Universal Control 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. (For details on how Universal Broker manages components, see the [Universal Broker 7.1.x Reference Guide](#).)

The default location for Universal Broker component definition files is **UNVPRD511/UNVCOMP**. The UCTL Server component member is **UCTL**.

The syntax of a component definition file is the same as a configuration file.

Component Definition Options

The following table identifies all of the options that comprise the UCTL Server 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 UCTL Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCTL Server.
CONFIGURATION_FILE	Name of the UCTL Server configuration file.
RUNNING_MAXIMUM	Maximum number of UCTL Servers that can run simultaneously.
START_COMMAND	Full path name of the UCTL Server program.
WORKING_DIRECTORY	Directory used as the working directory of the UCTL Server.

Universal Control Server for IBM i - Configuration

- [Overview](#)
- [Configuration File](#)
- [Configuration Options](#)

Overview

Universal Control Server configuration consists of defining runtime and default values. This page describes the Server configuration options.

Configuration File

The configuration file provides the simplest method of specifying configuration values that will not change with each command invocation. This file can be edited manually with any text editor (for example, Notepad).

The Universal Control Server configuration file name is specified in the Universal Control Server component definition. The default file name is **UNVPRD511/UNVCONF(UCTS)**.

Configuration Options

The following table identifies all of the Universal Control Server for IBM i configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
CODE_PAGE	Code page used for text translation.
MESSAGE_LEVEL	Level of messages written.
USER_SECURITY	Specification for whether or not user authentication is active.

Universal Control Server for IBM i - Universal Access Control List

Overview

The Universal Control Server uses the Universal Access Control List (UACL) file as an extra layer of security. The UACL file contains entries for the Universal Control Server. These entries contain Access Control List (ACL) rules that permit or deny access to the Universal Control Server.

UACL Entries

The syntax of a UACL entry file is the same as the Universal Control configuration file.

The following table identifies all UCTL Server for IBM i UACL entries. Each **UACL Entry Name** is a link to detailed information about that option.

UACL Entry Name	Description
UCTL_ACCESS	Allows or denies access to Universal Control Server services.
UCTL_REQUEST	Allows or denies access to Universal Control Server services based on client identification and request type.

Universal Control Server Configuration Options

Universal Control Server Configuration Options

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

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

Configuration Options List

The following table identifies all Universal Control Server configuration options.

Option	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
CODE_PAGE	Code page used for text translation.
EVENT_GENERATION	Events to be generated as persistent events.
INSTALLATION_DIRECTORY	Base directory in which Universal Control Server is installed.
LOGON_METHOD	Method of how users are logged onto the system.
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.
TMP_DIRECTORY	Directory name used for temporary files.
TRACE_DIRECTORY	Location of trace files.
USER_SECURITY	Specification for whether or not user authentication is active.

ACTIVITY_MONITORING - UCTL Server configuration option

Description

The ACTIVIITY_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 <i>option</i>	✓	✓	✓	✓

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 no.

CODE_PAGE - UCTL 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.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	codepage <i>codepage</i>	✓	✓	✓	✓

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](#) for information on 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**.)

Default

The default code page 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.

EVENT_GENERATION - UCTL 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	<code>event_generation types</code>	✓	✓	✓	✓

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:

- Inclusion operator is an asterisk (*).
- Exclusion operator is **X** or **x**.

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 - UCTL Server configuration option

Description

The `INSTALLATION_DIRECTORY` option specifies the location in which Universal Control Server is installed.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	<code>installation_directory <i>directory</i></code>		✓	✓	

Values

directory is the location in which UCMD Server is installed.

Default

UNIX	<code>/opt/universal/uctlsrv</code>
Windows	<code>c:\Program Files\Universal\uctlsrv</code>

LOGON_METHOD - UCTL 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 <i>option</i>			✓	

Values

option is the user's log on method.

Valid values for *option* are:

- batch**
 Windows log on type is **batch**. A batch logon is provided to establish an execution environment for processes that execute on a user's behalf, but without that user's direct interaction. When this logon type is used, the user account must have the "Allow log on as a batch job" permission granted to it. Likewise, the account must not have the "Deny log on as a batch job" policy assigned to it.
- interactive**
 Windows logon type is **interactive**. An interactive logon establishes an execution environment similar to one a user could expect to have when physically logged in to a workstation. When this logon type is used, the user account must have the "Allow log on locally" permission granted to it. Likewise, the account must not have the "Deny log on locally" policy assigned to it.

Default is interactive.

MESSAGE_LEVEL - UCTL 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 <i>level</i>	✓	✓	✓	✓

Values

level indicates either of the following level of messages:

- **trace**
Writes trace messages used for diagnostic purposes (see **Trace Files**, below).

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	Trace file name is *CURLIB/UNVTRCUCT(UCTn), where n is the Universal Server job number under which the Universal Control Server program is running. *CURLIB is the temporary library designated during the Universal Agent installation process; the default temporary library is UNVTMP511.
UNIX	Trace file name is uctsrv-N.trc , where N is the process ID of the Universal Control Server. It is created in the working directory of the Universal Control Server.
Windows	Trace file name is uctsrv-N.trc , where N is the process ID of the Universal Control Server. It is created in the working directory of the Universal Control Server.
z/OS	<p>There are two possible destinations of the trace data:</p> <ol style="list-style-type: none"> 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.UCT.Dyymmdd.Thhmmss.Cnnnnnnn, where: <ul style="list-style-type: none"> • #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. • nnnnnnn is the last seven digits of the Server's component ID in hexadecimal format. <p>Each time that a server is restarted, its sequence number is incremented. If a server is restarted more than 15 times, tracing is disabled.</p> 2. If ddname UNVTRMDL is not defined in the UBROKER started task procedure, member name Tnnnnnnn is created in the PDS or PDS/E allocated to the UNVTRACE ddname, where: <ul style="list-style-type: none"> • nnnnnnn is the last seven digits of the Server's component ID in hexadecimal format. • s is the component ID's sequence number from 0- F. <p>Each time that a server is restarted, its sequence number is incremented. If a server is restarted more than 15 times, tracing is disabled.</p> <p>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.</p>

MSG_SUPPRESSION_LIST - UCTL 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	<code>msg_suppression_list list</code>				

Values

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

NLS_DIRECTORY - UCTL Server configuration option

Description

The NLS_DIRECTORY option specifies the name of the directory where the Universal Control Server message catalog and code page tables are located.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	nls_directory <i>directory</i>		✓	✓	

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	<code>/opt/universal/nls</code>
Windows	<code>..\nls</code>

TMP_DIRECTORY - UCTL Server configuration option

Description

The TMP_DIRECTORY option specifies the directory name that the Universal Control Server uses for temporary files.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	tmp_directory <i>directory</i>		✓	✓	✓

Values

directory is the name of the directory.

It should specify a fully qualified path name.

UNIX	<code>/var/opt/universal/tmp</code>
Windows	<code>.. \tmp</code>
z/OS	<code>/tmp</code>

TRACE_DIRECTORY - UCTL Server configuration option

Description

The TRACE_DIRECTORY option specifies the directory name that the Universal Control Server uses for its trace files.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	trace_directory <i>directory</i>		✓	✓	

Values

directory is the name of the trace file directory.

Relative path names are relative to the Universal Control Server installation directory. Full path names are recommended.

UNIX	<code>/var/opt/universal/trace</code>
Windows	<code>C:\Program Files\Universal\UCtSrv</code>

USER_SECURITY - UCTL 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 remote Universal Control Manager requesting command execution is required to supply a local user ID and password. The user's command is started as that user.

If user security is not activated, the user ID and password is not required from the remote user. The user's process is started with the same user ID as Universal Control Server ID.

Usage


Method	Syntax	IBM i	UNIX	Windows	z/OS
Configuration File Keyword	security <i>option</i>	✓	✓	✓	✓

Values

option is the specification (and method) for activating user security.

IBM i	<ul style="list-style-type: none"> default Security is activated and uses IBM i authentication. none Security is not activated.
UNIX	<ul style="list-style-type: none"> default Use UNIX default user authentication method, <i>*/etc/passwd*</i> or <i>*/etc/shadow*</i>. inherit Universal Control Server will inherit the user account of the broker which started it. The user's action is performed with the same user ID as that inherited from the Broker. none Universal Control Server will inherit the user account of the broker which started it. The user's action is performed with the same user ID as that inherited from the Broker. pam Use the Pluggable Authentication Modules (PAM) interface. trusted Use HP Trust Security authentication.
Windows	<ul style="list-style-type: none"> default User-supplied user ID and password is authenticated against the user profile. inherit Universal Control Server will inherit the user account of the broker which started it. The user's action is performed with the same user ID as that inherited from the Broker. Not recommended. none Universal Control Server will inherit the user account of the broker which started it. The user's action is performed with the same user ID as that inherited from the Broker. Not recommended.
z/OS	<ul style="list-style-type: none"> default Use z/OS SAF user authentication method. The user ID must have an OMVS segment. inherit Universal Control Server will inherit the user account of the broker which started it. The user's action is performed with the same user ID as that inherited from the Broker. Not recommended. none Universal Control Server will inherit the user account of the broker which started it. The user's action is performed with the same user ID as that inherited from the Broker. Not recommended.

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 Control Component Definition Options

- [Universal Control Component Definition Options](#)
- [Component Definition Options Information](#)
 - [Description](#)
 - [Usage](#)
 - [Values](#)
- [Component Definition Options](#)

Universal Control Component Definition Options

This page provides links to detailed information about the options that comprise Universal Control (UCTL) 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	HP NonStop	UNIX	Windows	z/OS
Component Definition Keyword	<Format / Value>					

Method

Identifies the method used for specifying a Universal Control 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 (with a ?) the operating systems for which the 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**.

Component Definition Options

The following table identifies all of the options that can comprise a Universal Control component definition.


Component	Description
AUTOMATICALLY_START	Specification for whether or not the UCTL Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCTL Server.
CONFIGURATION_FILE*	Name of the UCTL Server's configuration file.
RUNNING_MAXIMUM	Maximum number of UCTL Servers that can run simultaneously.
START_COMMAND*	Program name of the UCTL Server.
WORKING_DIRECTORY*	Directory used as the working directory of the UCTL Server.
* These options are required in all component definitions.	

AUTOMATICALLY_START - UCTL Component Definition option





Description

The AUTOMATICALLY_START option specifies whether or not the Universal Control Server starts automatically when Universal Broker is started.

Note

 AUTOMATICALLY_START is optional in a component definition.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Component Definition Keyword	auto_start <i>option</i>				

Values

option is the specification for how the Universal Control Server is started.

The only valid value for *option* is:

- **no**
Universal Control Server is not started automatically when Universal Broker is started. It is started only on demand.

COMPONENT_NAME - UCTL Component Definition option

Description

The COMPONENT_NAME option specifies the name of the Universal Control Server.

Component start requests refer to Universal Control 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

Specification Method	Syntax	IBM i	UNIX	Windows	z/OS
Component Definition Keyword	component_name <i>name</i>	✓	✓	✓	✓

Values

name is the name of the Universal Control Server.

The only valid value for *name* is:

- **uctl**
(This is the name of the Universal Control Server component definitions file / member.)

Note




This name should not be changed.

CONFIGURATION_FILE - UCTL Component Definition option

Description

The CONFIGURATION_FILE option specifies the name of the Universal Control 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 <i>member</i> or configuration_file <i>filename</i>	✔	✔	✔	✔

Values

member / filename is the name of the configuration member / file.

IBM i	Non-qualified file names are located in the library list *LIBL.
UNIX	Relative paths are relative to the component's working directory.
Windows	Relative paths are relative to the component's working directory.
z/OS	Member names are located in the UNVCONF library allocated to the Universal Broker ddname UNVCONF . The installation default is UTS CFG00 .

RUNNING_MAXIMUM - UCTL Component Definition option

Description

The RUNNING_MAXIMUM option specifies the maximum number of UCTL Servers that can run simultaneously.

If this maximum number is reached, any command received to start a UCTL 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 <i>maximum</i>	✓	✓	✓	✓

Values

maximum is the maximum number of UCTL Servers that can run simultaneously.

Default is 100.

Note



If you specify 0 for *maximum*, the default (1) will be used. To use 0 for the maximum number of servers, specify -1 or less for *maximum*.


START_COMMAND - UCTL Component Definition option

Description

The START_COMMAND option specifies the full path name (member name for z/OS) of the Universal Control 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 <i>member</i>	✓	✓	✓	✓
	start_command <i>name</i> [<i>options</i>]				

Values

member / name is the program name of Universal Control Server.

options is the optional list of command line options.

IBM i	<i>name</i> is the Universal Control Server program. If the program name is non-qualified, the library list *LIBL is searched.
UNIX	<i>name</i> is the full path name of the Universal Control Server program.
Windows	<i>name</i> is the full path name of the Universal Control Server program.
z/OS	<i>member</i> is the program object of the Universal Control 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 Broker's STEPLIB ddname. Alternatively, starting with Universal Broker 5.1.0.4, <i>member</i> can be the fully specified path of a USS external link to the UCTL Server program.


WORKING_DIRECTORY - UCTL Component Definition option

WORKING_DIRECTORY




Description

The WORKING_DIRECTORY option specifies the full path name of the directory used as the working directory of UCTL Server.

Note

 WORKING_DIRECTORY is required in a component definition.

Usage


Method	Syntax	IBM i	UNIX	Windows	z/OS
Component Definition Keyword	working_directory <i>directory</i>				

Values

directory is the full path name of the working directory.

Default is (.).

Note

 Do not change this directory.

IBM i	working_directory serves as a required placeholder only.
UNIX	<i>directory</i> is the full path name of the directory Universal Control Server uses as its working directory.
Windows	<i>directory</i> is the full path name of the directory Universal Control Server uses as its working directory.
z/OS	<i>directory</i> is the HFS directory name that the Universal Control Server uses as its working directory.

Universal Control UACL Entries

- [Universal Control UACL Entries](#)
- [UACL Entries Information](#)
 - [Description](#)
 - [Usage](#)
 - [Values](#)
- [UACL Entries List](#)

Universal Control UACL Entries

This page provides links to detailed information on the Universal Access Control List (UACL) entries available for use with Universal Control.

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

UACL Entries Information

For each UACL entry, these pages provide the following information.

Description

Describes the UACL entry and how it is used.

Usage

Provides a table of the following information:

Method	Syntax	IBM i	HP NonStop	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
- HP NonStop
- 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 Control UACL Entries.

UACL Entry	Description
UCTL_ACCESS	<p>Allows or denies access to Universal Control Server services.</p> <p>There are two forms to this entry:</p> <ul style="list-style-type: none">• uctl_access• uctl_cert_access
UCTL_REQUEST	<p>Allows or denies access to Universal Control Server services based on client identification and request type.</p>

UCTL_ACCESS - UCTL UACL entry

Description

A UCTL_ACCESS UACL entry either allows or denies access to Universal Control Server services.

If access is permitted, UCTL_ACCESS also specifies whether or not user authentication is required.

There are two forms of the UCTL_ACCESS entry based on the client identification method:

- **uctl_access** form is for IP-based client identification.
- **uctl_cert_access** is for X.509 certificate-based client identification.

A **uctl_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 **uctl_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 [Universal Access Control List \(UACL\)](#) for details on *host*, *remote_user*, *local_user*, and *certid* specification syntax.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
UACL File Keyword	<code>uctl_access host,remote_user,local_user,access,auth</code>	✓	✓	✓	✓
	<code>uctl_cert_access certid,local_user,access,auth</code>				

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**
User ID provided by the Manager does not have to match the user process being stopped.

IBM i, UNIX, z/OS

Additionally, **noauth** specifies that the 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.

Windows

To set **noauth** via the Universal Configuration Manager, de-select **Requirematchinglocaluseraccount** when you are adding or editing an Access ACL (uctl_access) entry.

Note

noauth should be used with care. Turning off user authentication may violate your local security policies on the Server system.

UCTL_REQUEST - UCTL UACL entry

Description

A UCTL_REQUEST UACL entry allows or denies access to Universal Control Server services based on client identification and request type.

If access is permitted, the UCTL_REQUEST also specifies whether or not user authentication is required.

There are two forms of the UCTL_REQUEST entry based on the client identification method:

- **uctl_request** form is for IP-based client identification.
- **uctl_cert_request** is for X.509 certificate-based client identification.

A **uctl_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 **uctl_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	uctl_request <i>host,remote_user,local_user,req_type,req_name,access,auth</i>	✓	✓	✓	✓
	uctl_cert_request <i>certid,local_user,req_type,req_name,access,auth</i>				

Values

req_type specifies what type of request the Manager is requesting.

req_name further qualifies the request. The value of *req_name*, which depends on the value of *req_type*, is defined for each *req_type* below.

Valid values for *req_type* are:

- **refresh**
Manager request is for the refresh of an active component's configuration. *req_name* is a type of component, as specified in each component's definition. Not all component types can have their configurations refreshed from Universal Control.

Note



refresh does not include a Universal Broker REFRESH command, which is not processed by any UACL entry.

- **start**
Manager is requesting the start of the component.
req_name is the type of component which corresponds to an installed component definition. Not all components can be started from Universal Control.
- **stop**
Manager is requesting component termination.
req_name value is blank.

See [Universal Access Control List \(UACL\)](#) for details on *host*, *remote_user*, *local_user*, and *certid* specification syntax.

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**
User ID provided by the Manager does not have to match the user process being stopped.

IBM i, UNIX, z/OS



Additionally, **noauth** specifies that the 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.

Universal Copy

- [Universal Copy](#)
- [Usage](#)
- [Detailed Information](#)
- [Universal Copy Examples](#)

Universal Copy

Universal Copy (UCOPY) provides a means to copy files from either:

- Manager to a Server
- Server to Manager

Usage

Universal Copy copies files specified on its command line to stdout or a specified output file. The files are concatenated in the order specified on the command line. If no files are specified, it copies from stdin.

The default transfer mode used for the Universal Copy command is binary. In order to force end-of-line character interpretation, mode of text must be specified as a parameter of the Universal Copy command.

The default mode of transfer for standard in, standard out, and standard error is **text**. If binary is required, mode of **binary** must be specified on the standard file parameters.

Detailed Information

The following pages provide detailed information for Universal Copy:

- [Universal Copy for Windows and UNIX](#)
- [Universal Copy for IBM i](#)
- [Universal Copy Configuration Options](#)

Universal Copy Examples

See [Copying Files to and from Remote Systems](#) for examples of how to copy files using Universal Copy.

Universal Copy for Windows and UNIX

- [Configuration Options](#)
- [Command Line Syntax](#)
- [Command Operands](#)
 - [FILE](#)

Configuration Options

The following table identifies all Universal Copy for Windows and UNIX configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
HELP	Write a description of the command options and their format.
MESSAGE_LEVEL	Level of messages that will be displayed.
MODE	Mode in which input files are read and output files are written.
OUTPUT	File name that data is written to instead of standard output.
REPLACE	Specification for whether or not the file name specified with the OUTPUT option is replaced if it already exists.
TRANSACTIONAL	Specification for whether or not the copy operation is performed in transactional mode.
VERSION	Writes the program version and copyright information.

Command Line Syntax

The following figure illustrates the syntax — using the long form of command line options — of Universal Copy for Windows and UNIX.

```
ucopy
[-level {trace|audit|info|warn|error}]
[-mode {binary|text}]
[-outputfilename [-transactional {yes|no}] [-replace {yes|no}] ]
[file ...]

ucopy
{ -version | -help }
```


Command Operands

FILE

The file operand specifies the input files. Full or relative paths can be specified.

If no input files are specified, standard input is used.

z/OS USS

 z/OS USS permits the specification of files located in the hierarchical file system (HFS) and z/OS data sets.

HFS files are specified simply as UNIX file names.

z/OS data sets are specified using the IBM USS `//` convention, which prefixes the data set name with the characters `//`. The syntax is as follows:

```
//[']data.set.name[(member)][']
```

In order for the USS shell to interpret the forward slash (`/`) characters correctly, the complete file name must be enclosed in double (`"`) quotation marks on the USS command line.

The data set name adheres to TSO naming conventions; if it is not enclosed in apostrophes, your USS user name is used as the high-level qualifier. For example, `//my.data` refers to data set **USERID.MY.DATA**.

Universal Copy for IBM i

- [Overview](#)
- [Workload Automation 5 for IBM i Commands](#)
- [Universal Copy for IBM i Command](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

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

Overview

This page describes the configuration options and command line syntax of Universal Copy for the IBM i operating systems.

Note



Universal Copy became available for the IBM i environment with PTF 0UC0104 (level 1.2.1).

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 Universal Agent 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 [Change Release Tag](#) program, **UCHGRLS**, lets you change the tagged command names in **QSYS** to the untagged command names in **UNVPRD511**.

Note



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

Universal Copy for IBM i Command

The Universal Copy for IBM i command is **STRUCP**.

STRUCP copies files specified by FRMFILE and FRMFILES parameters to STDOUT or to a file specified by the TOFILE parameter. The files are concatenated in the order specified, starting with FRMFILE and continuing with the FRMFILES list. If no files are specified, it copies from STDIN to STDOUT.

Configuration Options

The following table identifies all Universal Copy for IBM i configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
CPY_MODE	Copy mode for reading and writing files.
FRMFILE	Name of a file to copy.
FRMFILES	Names of additional files to copy.
FRMMBR	Name of a member in the file specified by FRMFILE.
INPUTMODE	Mode for opening the file for input.
MESSAGE_LEVEL	Level of messages displayed by Universal Copy.
OUTPUTMODE	Mode for opening the file for output.
REPLACE	Specification for whether or not existing output file should be replaced.
TOFILE	Name of the output file that receives the specified concatenated input files.

TOMBR	Name of a member within the file specified by TOFILE.
TRANSACTIONAL	Specification for whether or not the copy operation is performed in transactional mode.
VERSION	Writes the program version and copyright information.

Command Line Syntax

The following figure illustrates the syntax — using the STRUCP parameter form of command line options — of Universal Copy for IBM i.

```

STRUCP
[FRMFILE([{*lib|*curlib|library name}/] {*stdin|filename}) [FRMMBR({*first|*all|member name})]
[FRMFILES( ([{*lib|*curlib|library name}/] file [*first|*all}])...)] ]
[TOFILE( [{*lib|*curlib|library name}/] {*stdout|filename}) [TOMBR(member)] ]
[CPYMODE(*binary|*text|*savf)]

Additional Options
[REPLACE(*yes|*no)]
[INPUTMODE('option')]
for input files.
[OUTPUTMODE('option')]
for output files.
[MSGLEVEL(*trace|*audit|*info|*warn|*error)]

STRUCP
VERSION(yes|no)
    
```

Note: Overrides CPYMODE

Note: Overrides CPYMODE

Warning



Under IBM i, the STRUCP default mode (CPYMODE) is **text**. For all other platforms, the default is **binary**.

Universal Copy Configuration Options

- [Universal Copy Configuration Options](#)
- [Configuration Options](#)

Universal Copy Configuration Options

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

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

Configuration Options

The following table identifies the Universal Copy configuration options.

Option Name	Description
COPY_MODE	Copy mode for reading and writing files.
FRMFILE	Name of a file to copy.
FRMFILES	Names of additional files to copy.
FRMMBR	Name of a member in the file specified by FRMFILE.
HELP	Displays a description of the command line options and their format.
INPUTMODE	Mode for opening the file for input.
MESSAGE_LEVEL	Level of messages displayed by Universal Copy.
MODE	Mode in which input files are read and output files are written.
OUTPUT	File name that data is written to instead of standard output.
OUTPUTMODE	Mode for opening the file for output.
REPLACE	Specification for whether or not the file name specified with the OUTPUT option is replaced if it already exists.
TOFILE	Output file that receives specified concatenated input files.
TOMBR	Name of a member in the file specified by TOFILE.
TRANSACTIONAL	Specification for whether or not the copy operation is performed in transactional mode.
VERSION	Writes the program version and copyright information.

CPY_MODE - UCOPY configuration option

Description

The CPY_MODE option sets the copy mode for reading and writing files.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	n/a				
STRUCP Parameter	CPYMODE(*mode)	✓			

Values

mode is the copy mode for reading and writing files.

Valid values for *mode* are:

- **binary**
Copy the data as binary data. The data is not translated in any manner.
- **text**
Copy the data as text data. All trailing blank characters in a record are ignored. A new-line character is inserted after the last non-blank character. The data is subjected to code page conversions.
- **save**
Copy the data as a save file. This is required when working with save files. The data is not translated in any manner.

Default is text.

FRMFILE - UCOPY configuration option

Description

The FRMFILE option specifies the name of a file to copy.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	n/a				
STRUCP Parameter	FRMFILE([<i>library</i>] <i>file</i>)	✓			

Values

file is the name of a file to copy.

Valid values for *file* are:

- ***stdin**
Data is copied from the job's standard input file.
 - If executed from an interactive job, standard input is allocated to the terminal. An ILE session manager screen is displayed in which the user enters the data to copy.
 - If executed from a batch job, file **QINLINE** is allocated to standard input.
- *file name_*
Data is copied from the specified file.

Default is *stdin.

library is the name of a library with which *file* optionally can be qualified.

- ***libl**
File is located in the library list.
- ***curlib**
File is located in the current library.
- *library name*
File is located in the specified library.


FRMFILES - UCOPY configuration option

Description

The FRMFILES option specifies the names of additional files to copy.

Files are copied in the order listed, starting with the file specified by the [FRMFILE](#) option and continuing with the files specified in FRMFILES. The resulting output file is a concatenation of all input files. Up to 39 files can be specified in the list.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	n/a				
STRUCP Parameter	FRMFILES([<i>library/</i>] <i>file</i> [<i>member</i>])...				

Values

file is the name of an additional file to copy.

library is the name of a library with which *file* optionally can be qualified.

Valid values for *library* are:

- ***libl**
File is located in the library list.
- ***curlib**
File is located in the current library.
- *library name*
File is located in the specified library.

member is the name of a member in the specified file to copy.

Valid values for *member* are:

- ***first**
First member in the file is processed.
- ***all**
All members in the file are processed as one.

Default is *first.

FRMMBR - UCOPY configuration option

Description

The FRMMBR option specifies the name of a member in the file specified by [FRMFILE](#).

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	n/a				
STRUCP Parameter	FRMMBR (<i>member</i>)	✓			

Values

member is the name of a member in the file.

Valid values for *member* are:

- ***first**
First member in the file is processed.
- ***all**
All members in the file are processed as one.
- *member name*
Specified member name is processed.

Default is *first.

HELP - UCOPY configuration option

Description

The HELP option displays a description of the Universal Copy 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		✓	✓	
STRUCP Parameter	n/a				

Values

(There are no values for this option.)

INPUTMODE - UCOPY configuration option

Description

The INPUTMODE option specifies the mode for opening the file for input.

If this option is used, it overrides the [CPY_MODE](#) option for input files. See the **fopen()** function in the C Runtime Library manual (*ILE C/C++ for iSeries Run-Time Library Functions*) for information.

The default is to use the [CPY_MODE](#) option.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	n/a				
STRUCP Parameter	INPUTMODE('option')	✓			

Values

'option' is the mode for opening the file.

MESSAGE_LEVEL - UCOPY configuration option

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-l <i>level</i>		✓	✓	
Command Line, Long Form	-level <i>level</i>		✓	✓	
STRUCP Parameter	MSGLEVEL(* <i>level</i>)	✓			

Values

level indicates either of the following level of messages:

- **trace**
Writes trace messages used for diagnostic purposes.

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

IBM i	info
UNIX	warn
Windows	warn

MODE - UCOPY configuration option

Description

The MODE option specifies the mode in which input files are read and output files are written.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-m <i>mode</i>		✓	✓	
Command Line, Long Form	-mode <i>mode</i>		✓	✓	
STRUCP Parameter	n/a				

Values

mode is the mode in which the files are read and written.

Valid values for *mode* are:

- **binary**
Treats all data as binary data. No interpretation of end-of-line characters or end-of-file characters is performed.
- **text**
Treats all data as text data. End-of-line characters are interpreted.

Default is binary.


UNIX	There is no difference between binary and text.
-------------	---

OUTPUT - UCOPY configuration option

Description

The OUTPUT option specifies the name of a file to which data is written instead of standard output.

Note

See  **FILE** in [Universal Copy for Windows and UNIX](#) for operating system-specific file naming conventions.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-o <i>filename</i>		✓	✓	
Command Line, Long Form	-output <i>filename</i>		✓	✓	
STRUCP Parameter	n/a				

Values

filename is the name of the file to which data is written.

Default is standard output.

OUTPUTMODE - UCOPY configuration option

Description

The OUTPUTMODE option specifies the mode for opening the file for output.

If this option is used, it overrides the [CPY_MODE](#) option for output files. See the **fopen()** function in the C Runtime Library manual (*ILE C/C++ for iSeries Run-Time Library Functions*) for information.

The default is to use the [CPY_MODE](#) option.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	n/a				
STRUCP Parameter	OUTPUTMODE('option')	✓			

Values

mode is the mode for opening the file.

REPLACE - UCOPY configuration option

Description

The REPLACE option specifies whether or not the file specified by the [OUTPUT](#) option is replaced (if it already exists).

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-r <i>option</i>		✓	✓	
Command Line, Long Form	-replace <i>option</i>		✓	✓	
STRUCP Parameter	REPLACE(* <i>option</i>)	✓			

Values

option is the specification for whether or not to replace the file.

Valid values for *option* are:

- **yes**
File is replaced.
- **no**
File is not replaced.

Default is yes.

TOFILE - UCOPY configuration option

Description

The TOFILE option specifies name of the output file that receives the specified concatenated input files.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	n/a				
STRUCP Parameter	TOFILE([<i>library</i> '] <i>file</i>)	✓			

Values

file is the name of the output file receiving the input files.

Valid values for *file* are:

- ***stdout**
Output is written to standard output.
 - If executed from an interactive job, standard output is allocated to the terminal from which STRUCP is executed. The ILE session terminal is displayed to view the output.
 - If executed from a batch job, standard output is allocated to file **QPRINT**.
- *filename*
Output is written to the specified file name. If the file is not found, it is created as a physical source file with a record length of 266.

Default is *stdout.

library is optional name of a library with which *file* can be qualified.

Valid values for library are:

- ***libl**
File is located in the library list.
- ***curlib**
File is located in the current library.
- *library name*
File is located in the specified library.

TOMBR - UCOPY configuration option

Description

The TOMBR option specifies the name of a member in the file specified by the TOFILE option.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	n/a				
STRUCP Parameter	TOMBR(<i>member</i>)	✓			

Values

member is the name of a member in the file.

Default is the file name.

TRANSACTIONAL - UCOPY configuration option

Description

The TRANSACTIONAL option specifies whether or not the copy operation is performed in transactional mode.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-t <i>option</i>		✓	✓	
Command Line, Long Form	-transactional <i>option</i>		✓	✓	
STRUCP Parameter	n/a				

Values

option is the specification for whether or not the copy operation is performed in transactional mode.

Valid values for *option* are:

- **yes**
Data is copied in a transactional mode. The data first is copied to a temporary file on the same file system. When the copy operation completes successfully, the temporary file is renamed to the file name specified by the OUTPUT option.
- **no**
Data is not copied in a transactional mode.

Default is no.

VERSION - UCOPY configuration option

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-v		✓	✓	
Command Line, Long Form	-version		✓	✓	
STRUCP Parameter	VERSION (<i>option</i>)	✓			

Values

There are no values for this option.

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 Database Dump

Universal Database Dump

Universal Agent databases are implemented using Oracle's Berkeley Database product. The Berkeley Database provides utilities to perform administrative database tasks.

The Universal Database Dump (UDBDUMP) utility is the Berkeley **db_dump** utility tailored specifically for Universal Agent databases.

UDBDUMP and the [Universal Database Load](#) (UDBLOAD) utility are provided to enable recovery from a corrupted Berkeley database. Databases can potentially become corrupt due to system and address spaces ending abnormally.

Oracle documentation on **db_dump** and all other utility commands is provided at the following URL:

http://docs.oracle.com/cd/E17076_04/html/api_reference/C/utilities.html

Usage

UDBDUMP invokes the Berkeley **db_dump** utility. The UDBDUMP command line options are passed to **db_dump**. UDBDUMP reads a specified database file and dumps the contents to a database dump file.

This dump file can be loaded into a database using UDBLOAD.

Detailed Information

The following pages provide detailed information for Universal Database Dump:

- [Universal Database Dump for z/OS](#)
- [Universal Database Dump for Windows and UNIX](#)
- [Universal Database Dump Configuration Options](#)

Universal Database Dump for zOS

- [JCL Procedure](#)
- [DD Statements used in JCL Procedure](#)
- [JCL](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

JCL Procedure

The following figure illustrates the Universal Database Dump for z/OS JCL procedure (**UDBDPRC**, located in the **SUNVSAMP** library) that is provided to simplify the execution JCL and future maintenance.

```
//UDBLPRC  PROC DBOPTS=-r,
//          DBFILE=,
//          SHLQ=#SHLQ,
//          DMPDSN=,
//          DBHFS=
// *
//S1       EXEC PGM=UDBDUMP,
//PARM= 'ENVAR (TZ=EST5EDT) /&DBHFS&DBOPTS&DBFILE'
//STEPLIB DD DSN=&SHLQ..UNV.SUNVLOAD,
//          DISP=SHR
// *
//UNVOUT  DD DSN=&DMPDSN,
//          DISP=SHR
// *
//SYSPRINT DD SYSOUT=*
//SYSOUT  DD SYSOUT=*
//SYSIN   DD DUMMY
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//          PEND
```

DD Statements used in JCL Procedure

The following table describes the DD statements used in the Universal Database Dump for z/OS [JCL procedure](#), above.

ddname	Description
STEPLIB	Load library in which program UDBDUMP program is located.
UNVOUT	Database dump file. The dump data set must be a physically sequential data set with a variable-block record format, a record length of 32756, and a block size of 32760.
SYSPRINT	UDBDUMP standard output ddname.
SYSOUT	UDBDUMP standard error ddname.
SYSIN	UDBDUMP standard input.

JCL

The following figure illustrates the Universal Database Dump for z/OS JCL using the **UDBDPRC** [JCL procedure](#), above.

```
//S1 EXEC PGM=UDBDUMP,
// PARM=' ENVAR(TZ=EST5EDT)/&DBHFS&DBOPTS&DBFILE'
//STEPLIB DD DISP=SHR,DSN=&SHLQ..UNV.SUNVLOAD
// *
//UNVOUT DD DISP=SHR,DSN=&DMPDSN
// *
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD DUMMY
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

Configuration Options

The following table identifies the Universal Database Dump for z/OS configuration options. Each **Option Name** is a link to detailed information about that option.

It describes only those options relevant for database recovery. For details on all options, see the Oracle documentation on the **db_dump** utility at URL:

<http://www.oracle.com/technology/documentation/berkeley-db/db/index.html>

Option Name	Description
DATABASE_FILE	Database file to be dumped.
DUMP_OPTIONS	Controls database dump behavior, including the format of the dump output and the extent to which data is recovered from a possibly corrupt database file

Note



UDBDUMP accepts configuration options only on the PARM keyword of the **EXEC** statement.

Command Line Syntax

The following figure illustrates the command line syntax of Universal Database Dump for z/OS. It identifies only those options that are relevant for database recovery.

```
[-r-p]
database
```

Universal Database Dump for Windows and UNIX

Configuration Options

The following table identifies the UDBDUMP for Windows and UNIX configuration options. Each **Option Name** is a link to detailed information about that option.

It describes only those options relevant for database recovery. For details on all options, see the Oracle documentation on the **db_dump** utility at URL:

<http://www.oracle.com/technology/documentation/berkeley-db/db/index.html>

Option Name	Description
DATABASE_FILE	Database file to be dumped.
DUMP_OPTIONS	Controls database dump behavior, including the format of the dump output and the extent to which data is recovered from a possibly corrupt database file

Command Line Syntax

The following figure illustrates the syntax of UDBDUMP for Windows and UNIX. It identifies only those options that are relevant for database recovery.

```
udb_dump  
[-r-p]  
database > dump
```

Universal Database Dump Configuration Options

Universal Database Dump Configuration Options

This page provides links to detailed information on the configuration options available for use with Universal Database Dump (UDBDUMP). UDBDUMP is the Berkeley **db_dump** utility tailored specifically for Universal Agent databases.

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

Configuration Options List

The following table identifies the Universal Database Dump configuration options.

Option Name	Description
DATABASE_FILE	Database file to be dumped.
DUMP_OPTIONS	Controls database dump behavior, including the format of the dump output and the extent to which data is recovered from a possibly corrupt database file.


DATABASE_FILE - UDBDUMP configuration option

Description

The DATABASE_FILE option specifies the database file to be dumped.

DATABASE_FILE is the last option specified on the command line.

z/OS

 The database file must be located in the root directory of the HFS data set allocated on the UNVDB ddname. The HFS data set must be mounted prior to running UDBLOAD.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	<i>database</i>			✓	✓	✓

Values

database is the database file to be dumped.

DUMP_OPTIONS - UDBDUMP configuration option

Description

The DUMP_OPTIONS option controls database dump behavior, including the format of the dump output and the extent to which data is recovered from a possibly corrupt database file.

There are two forms of the DUMP_OPTIONS option:

- **-p** controls the format of the dump output, causing some printable characters to be dumped as text characters. This form of DUMP_OPTIONS is useful if you intend use standard text editors and tools to modify the contents of databases upon reload.
- **-r** controls data recovery behavior, and instructs the utility to recover as many records as possible.

Note



UDBDUMP also accepts a **-R** option. However, the upper case **-R** is not recommended. It specifies that aggressive recovery should be performed, which can result in potentially bad or deleted records being recovered.

When using DUMP_OPTIONS, only use lower case **-r**.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-p -r			✓	✓	✓
Command Line, Long Form	n/a					

Values

(There are no values used with this option.)

Universal Database Load

Universal Database Load

Universal Agent databases are implemented using Oracle's Berkeley Database product. The Berkeley Database provides utilities to perform administrative database tasks.

The Universal Database Load (UDBLOAD) utility is the Berkeley **db_load** utility tailored specifically for Universal Agent databases.

UDBLOAD and the [Universal Database Dump](#) (UDBDUMP) utility are provided to enable recovery from a corrupted Berkeley databases. Databases can potentially become corrupt due to system and address spaces ending abnormally.

Oracle documentation on **db_load** and all other utility commands is provided at the following URL:

http://docs.oracle.com/cd/E17076_04/html/api_reference/C/utilities.html

Usage

UDBLOAD invokes the Berkeley **db_load** utility. The UDBLOAD command line options are passed to **db_load**. UDBLOAD reads the database dump file and loads the contents into the specified database file.

The database dump file is created with UDBDUMP.

Note



By default, the load operation overwrites the specified database. To simply update the database, omit the **OVERWRITE** configuration option.

You should back up the database file prior to performing any load operation.

Detailed Information

The following pages provide detailed information for Universal Database Load:

- [Universal Database Load for z/OS](#)
- [Universal Database Load for Windows and UNIX](#)
- [Universal Database Load Configuration Options](#)

Universal Database Load for zOS

- [JCL Procedure](#)
- [DD Statements used in JCL Procedure](#)
- [JCL](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

JCL Procedure

The following figure illustrates the Universal Database Load for z/OS JCL procedure (**UDBLPRC**, located in the **SUNVSAMP** library) that is provided to simplify the execution JCL and future maintenance.

```
//UDBLPRC PROC DBOPTS=-o,
//          DBFILE=,
//          SHLQ=#SHLQ,
//          DMPDSN=,
//          DBHFS=
// *
//S1       EXEC PGM=UDBLOAD,
// PARM=' ENVAR(TZ=EST5EDT)/&DBHFS&DBOPTS&DBFILE'
//STEPLIB DD DSN=&SHLQ..UNV.SUNVLOAD,
//          DISP=SHR
// *
//UNVIN    DD DSN=&DMPDSN,
//          DISP=SHR
// *
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSIN    DD DUMMY
//CEEDUMP  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//          PEND
```

DD Statements used in JCL Procedure

The following table describes the DD statements used in the Universal Database Load for z/OS [JCL procedure](#), above.

ddname	Description
STEPLIB	Load library in which program UDBLOAD program is located.
UNVIN	Database dump file produced by Universal Database Dump .
SYSPRINT	UDBLOAD standard output ddname.
SYSOUT	UDBLOAD standard error ddname.
SYSIN	UDBLOAD standard input.

JCL

The following figure illustrates the Universal Database Load for z/OS JCL using the **UDBLPRC**[JCL procedure](#), above.

```
//S1 EXEC PGM=UDBLOAD,
// PARM=' ENVAR(TZ=EST5EDT) /&DBHFS&DBOPTS&DBFILE '
//STEPLIB DD DISP=SHR,DSN=&SHLQ..UNV.SUNVLOAD
// *
//UNVIN DD DISP=SHR,DSN=DB.DUMP
// *
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD DUMMY
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// *
```

Configuration Options

The following table identifies the Universal Database Load for z/OS configuration options. Each **Option Name** is a link to detailed information about that option.

It describes only those options relevant for database recovery. For details on all options, see the Oracle documentation on the **db_dump** utility at URL:

<http://www.oracle.com/technology/documentation/berkeley-db/db/index.html>

Option Name	Description
DATABASE_FILE	Database file to be loaded.
OVERWRITE	Specification to overwrite the database file, not update it.

Note



UDBLOAD accepts configuration options only on the PARM keyword of the **EXEC** statement.

Command Line Syntax

The following figure illustrates the command line syntax of Universal Database Load for z/OS. It identifies only those options that are relevant for database recovery.

```
[-o]
database
```

Universal Database Load for Windows and UNIX

Configuration Options

The following table identifies the UDBLOAD for Windows and UNIX configuration options. Each **Option Name** is a link to detailed information about that option.

It describes only those options relevant for database recovery. For details on all options, see the Oracle documentation on the **db_dump** utility at URL:

<http://www.oracle.com/technology/documentation/berkeley-db/db/index.html>

Option Name	Description
DATABASE_FILE	Database file to be loaded.
OVERWRITE	Specification to overwrite the database file, not update it. (Update is the default load operation.)

Command Line Syntax

The following figure illustrates the syntax of UDBLOAD for Windows and UNIX. It identifies only those options that are relevant for database recovery.

```
udb_load  
[-o]  
database< dump
```

Universal Database Load Configuration Options

Overview

This page provides links to detailed information on the configuration options available for use with Universal Database Load (UDBLOAD). UDBLOAD is the Berkeley **db_load** utility tailored specifically for Universal Agent databases.

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

Configuration Options List

The following table identifies the Universal Database Load configuration options.

Option Name	Description
DATABASE_FILE	Database file to be loaded.
OVERWRITE	Specification to recover as many records as possible from a possibly corrupt database file.

DATABASE_FILE - UDBLOAD configuration option

Description

The DATABASE_FILE option specifies the database file to be loaded.

DATABASE_FILE is the last option specified on the command line.

z/OS



The database file must be located in the root directory of the HFS data set allocated on the UNVDB ddname. The HFS data set must be mounted prior to running UDBLOAD.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	<i>database</i>			✓	✓	✓

Values

database is the database file to be loaded.

OVERWRITE - UDBLOAD configuration option

Description

The OVERWRITE option specifies that the database file is to be overwritten, not updated.

To simply update the database, omit this option.

OVERWRITE is specific to Universal Database Load, not to the Berkeley **db_load** utility.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-o			✓	✓	✓
Command Line, Long Form	n/a					

Values

(There are no values specified for this option.)

Universal Display Log File

- [Universal Display Log File](#)
- [Workload Automation 5 for IBM i Commands](#)
- [Command Line Syntax](#)
- [Detailed Information](#)

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

Universal Display Log File

Universal Display Log File (**UDSPLOGF**) is a command for the IBM i environment. It consists of a command line program followed by a list of configuration options.

Universal Display Log File reads job log output files that were created as a result of API **QMHCTLJL** or command **DSPJOBLOG**. The job log is formatted and written to standard output.

Optionally, **UDSPLOGF** can delete the job log file members after writing. File member deletion is controlled by the [REMOVE_MEMBERS](#) option. The default behavior is to leave the members unaltered.

Note



Universal Display Log File became available for the IBM i environment with PTF 0UC0114 (maintenance level 1.2.1.10).

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 Universal Agent 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 [Change Release Tag](#) program, **UCHGRLS**, lets you change the tagged command names in **QSYS** to the untagged command names in **UNVPRD511**.

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

Command Line Syntax

The following figure illustrates the command line syntax of Universal Display Log File.

```
UDSPLOGF
[ PRMRYFILE(filename[library]) [PRMRYMBR(member)] ]
[ SCNDRYFILE(filename[library]) [SCNDRYMBR(member)] ]
[ REMOVE({yes|no}) ]
```

Detailed Information

The following pages provide detailed information for Universal Display Log File:

- [Universal Display Log File Configuration Options](#)

Universal Display Log File Configuration Options

Overview

This page provides links to detailed information on the configuration options available for use with Universal Display Log File.

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

Configuration Options List

The following table identifies the Universal Display Log File configuration options.

Option Name	Description
PRIMARY_FILE	Name of the primary output file.
PRIMARY_MEMBER	Name of the primary output file member.
REMOVE_MEMBERS	Controls the deletion of job log output file members.
SECONDARY_FILE	Name of the secondary output file.
SECONDARY_MEMBER	Name of the secondary output file member.

PRIMARY_FILE - UDSPLOGF configuration option

Description

The PRIMARY_FILE option specifies the name of the primary output file.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
UDSPJOGF Parameter	PRMRYFILE (<i>filename</i> [<i>library</i>])	✓				

Values

filename is the name of the primary output file.

filename can be qualified by a *library* name.

PRIMARY_MEMBER - UDSPLOGF configuration option

Description

The PRIMARY_MEMBER option specifies the name of the primary output file member.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
UDSPJOGF Parameter	PRMRYMBR (<i>member</i>)	✓				

Values

member is the name of the primary output file member.

REMOVE_MEMBERS - UDSPLOGF configuration option

Description

The REMOVE_MEMBERS option controls the deletion of job log output file members.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
UDSPJOGF Parameter	REMOVE (<i>*option</i>)					

Values

**option* is the specification for controlling the deletion of the job log output file members.

Valid values for **option* are:

- **yes**
Specified members will be deleted as a result of running this command.
- **no**
Specified job log output members will not be deleted as a result of running this command.

Default is no.

SECONDARY_FILE - UDSPLOGF configuration option

Description

The SECONDARY_FILE option specifies the name of the secondary output file.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
UDSPJOGF Parameter	SCNDRYFILE (<i>filename</i> { <i>library</i> })	✓				

Values

filename is the name of the secondary output file.

filename can be qualified by a *library* name.

SECONDARY_MEMBER - UDSPLOGF configuration option

Description

The SECONDARY_MEMBER option specifies the name of the secondary output file member.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
UDSPJOGF Parameter	SCNDRYMBR (<i>member</i>)	✓				

Values

member is the name of the secondary output file member.

Universal Encrypt

- [Universal Encrypt](#)
- [Usage](#)
- [Detailed Information](#)
- [Universal Encrypt Examples](#)
- [Universal Broker Key Store](#)

Universal Encrypt

Universal Agent programs have the ability to read command line options contained in command files. Command files that contain private information must be protected by using local file system security. This ensures that only authorized accounts have read access.

The Universal Encrypt (**UENCRYPT**) utility adds an additional layer of security by encrypting the contents of command files into an unintelligible format.

Although all command line options can be encrypted with the Universal Encrypt utility, most organizations use it to encrypt and store authentication credentials such as user ID and/or password.

An encrypted command file can be decrypted only by Stonebranch product programs. No decrypt command is provided to decrypt the command file.

Note



Universal Encrypt should not be used as a replacement for file system security.

Usage

Universal Encrypt reads a command file from its standard input and writes an encrypted command file to its standard output.

The encrypted command file is a text file that can be used on any operating system by any Universal Agent component. Lines starting with a hash (#) character in column one are comments; blank lines are ignored.

Universal Encrypt performs operations specified by the command options.

Detailed Information

The following pages provide detailed information for Universal Encrypt:

- [Universal Encrypt for z/OS](#)
- [Universal Encrypt for Windows and UNIX](#)
- [Universal Encrypt for IBM i](#)
- [Universal Encrypt Configuration Options](#)

Universal Encrypt Examples

See [Encryption - Examples](#) in the Universal Agent 7.1.x Reference Guide for examples of how to encrypt files using Universal Encrypt.

Universal Broker Key Store

During installation, you can request the generation of an encryption key, which is stored in a Universal Broker key store.

If a Universal Agent component wants to use this encryption key, it requests it from the Universal Broker.

For detailed information on encryption keys and the key store, see [Universal Broker Key Store](#).

Universal Encrypt for zOS

- [JCL](#)
- [DD Statements used in JCL](#)
- [Configuration](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

JCL

The following figure illustrates the Universal Encrypt for z/OS JCL.

```
//UENCRYPT EXEC PGM=UENCRYPT
//STEPLIB DD DISP=SHR,DSN=UNV.SUNVLOAD
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//UNVIN DD DISP=SHR,DSN=MY.CLEAN.COMDFILE
//UNVOUT DD DISP=SHR,DSN=MY.ENCRYPT.COMDFILE
//SYSIN DD *
      -KEY DF#98AD@ -AES YES,/*
```

DD Statements used in JCL

The following table describes the DD statements used in the Universal Encrypt for z/OS [#JCL](#), above.

ddname	Description
STEPLIB	Load library in which program UENCRYPT is located.
SYSPRINT	UENCRYPT standard output ddname.
SYSOUT	UENCRYPT standard error ddname.
UNVIN	Clear text command file to encrypt.
UNVOUT	Encrypted command file.
SYSIN	UENCRYPT standard input from which parameters are read.

Configuration

Universal Encrypt operations are controlled by the configuration options specified either on the PARM keyword of the **EXEC** statement or in the SYSIN ddname.

Configuration Options

The following table identifies the Universal Encrypt for z/OS configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AES	Specification for whether or not AES encryption is used.
CODE_PAGE	Character code page used to translate text data received and encrypted.
ENCRYPTION_KEY	Encryption key used by the encryption algorithm.

<code>GENERATE_KEY</code>	Specification for whether or not to generate an encryption key.
<code>HELP</code>	Writes a description of the command options and their format.
<code>KEYSTORE_PATH</code>	Path to the encryption key file.
<code>STORE_KEY</code>	Specification for whether or not to store the encryption key (generated or specified) in the local Universal Broker key store.
<code>VERSION</code>	Writes the program version and copyright information.

Command Line Syntax

The following figure illustrates the command line syntax - using the long form of command line options - of Universal Encrypt for z/OS.

```
uencrypt
[-keykey]
[-genkey {yes|no}]
[-keypathpath]
[-store {yes|no}]
[-codepagecodepage]
[-aes {yes|no|legacy}]

uencrypt
{ -help | -version }
```

Universal Encrypt for Windows and UNIX

Configuration Options

The following table identifies the Universal Encrypt for Windows and UNIX configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AES	Specification for whether or not AES encryption is used.
BIF_DIRECTORY	Broker Interface File (BIF) directory where the Universal Broker interface file, ubroker.bif , is located.
CODE_PAGE	Character code page used to translate text data received and encrypted.
ENCRYPTION_KEY	Encryption key used by the encryption algorithm.
GENERATE_KEY	Specification for whether or not to generate an encryption key.
HELP	Writes a description of the command options and their format.
KEYSTORE_PATH	Path to the encryption key file.
NLS_DIRECTORY	Directory name where Universal Encrypt can find its code page tables.
STORE_KEY	Specification for whether or not to store the encryption key (generated or specified) in the local Universal Broker key store.
VERSION	Writes the program version and copyright information.

Command Line Syntax

The following figure illustrates the command line syntax - using the long form of configuration options - of Universal Encrypt for Windows and UNIX.

Note



If Universal Agent is installed in user mode, the `uencrypt` command works only when `-bif_directory` is specified on the command line.

```
uencrypt
[-keykey]
[-genkey {yes|no}]
[-keypathpath]
[-store {yes|no}]
[-bif_directorydirectory]
[-codepagecodepage]
[-nlsdirdirectory]
[-aes {yes|no|legacy}]

uencrypt
{ -help | -version }
```

Universal Encrypt for IBM i

- [Universal Agent for IBM i Commands](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

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

Universal Agent 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 [Change Release Tag](#) program, **UCHGRLS**, lets you change the tagged command names in **QSYS** to the untagged command names in **UNVPRD511**.

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

Configuration Options

The following table identifies the Universal Encrypt for IBM i configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AES	Specification for whether or not AES encryption is used.
CODE_PAGE	Character code page used to translate text data.
ENCRYPTION_KEY	Encryption key used to encrypt the input file.
INPUT_FILE	Input file that is to be encrypted.
OUTPUT_FILE	File to which the encrypted input file is written.
VERSION	Writes the program version information and copyright.

Command Line Syntax

The following figure illustrates the command line syntax - using the STRUEN parameter form of configuration options - of Universal Encrypt for IBM i.

```
STRUEN
[ INFILE(input_file) [ INMBR(member) ] ]
[ OUTFILE(output_file) [ OUTMBR(member) ] ]
[ KEY(key) ]
[ AES(*{yes|no}) ]
[ CODEPAGE(codepage) ]
```

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

Note

Options entered into plain text files or encrypted files must be in short form or long form syntax (see [COMMAND_FILE_PLAIN](#) and [COMMAND_FILE_ENCRYPTED](#)).

Universal Encrypt Configuration Options

Universal Encrypt Configuration Options

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

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

Configuration Options List

The following table identifies the Universal Encrypt configuration options.

Option Name	Description
AES	Specification for whether or not AES encryption is used.
BIF_DIRECTORY	Broker Interface File (BIF) directory where the Universal Broker interface file, ubroker.bif , is located.
CODE_PAGE	Character code page used to translate text data received and encrypted.
ENCRYPTION_KEY	Encryption key used by the encryption algorithm.
GENERATE_KEY	Specification for whether or not to generate an encryption key.
HELP	Displays a description of the command line options and their format.
INPUT_FILE	Input file that is to be encrypted.
KEYSTORE_PATH	Path to the encryption key file.
NLS_DIRECTORY	Directory name where Universal Encrypt can find its code page tables.
OUTPUT_FILE	File to which the encrypted input file is written.
STORE_KEY	Specification for whether or not to store the encryption key (generated or specified) in the local Universal Broker key store.
VERSION	Writes the program version and copyright information.

AES - UENCRYPT configuration option

Description

The AES option specifies whether or not AES (Advanced Encryption Standard) encryption is used.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-a <i>option</i>		✓	✓	✓
Command Line, Long Form	-aes <i>option</i>		✓	✓	✓
STRUEN Parameter	AES(<i>*option</i>)	✓			

Values


option is the specification for whether or not to use AES encryption.

Valid values for *option* are:

- **yes**
Use AES 256-bit encryption in CBC mode.
- **no**
Use DES 56-bit encryption.
- **legacy**
Use AES 256-bit encryption in CBC mode with pre-6.3.0.1 internal key (if user key is not specified).

Default is yes.

Note

 In version 6.3.0.1, the internal key used for AES encryption was enhanced to provide greater security. As a result, encrypted command files generated with version 6.3.0.1 or later that do not use a user-specified key, **by default**, will not be compatible with Universal Agent components earlier than 6.3.0.1.

However, specifying AES **legacy** will force UENCRYPT to generate an encrypted command file using the pre-6.3.0.1 internal key, allowing for backwards compatibility.

Version 6.3.0.1 and later Agent components are fully backwards compatible with encrypted command files generated with any version of UENCRYPT.

BIF_DIRECTORY - UENCRYPT 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 <i>directory</i>		✓	✓	
Environment Variable	UENBIFDIRECTORY= <i>directory</i>		✓	✓	
Configuration File Keyword	n/a				

Values

directory is the name of the BIF directory.

Default is the main installation directory.

CODE_PAGE - UENCRYPT configuration option

Description

The CODE_PAGE option specifies the character code page used to translate text data.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-t <i>codepage</i>		✓	✓	✓
Command Line, Long Form	-codepage <i>codepage</i>		✓	✓	✓
STRUEN Parameter	CODEPAGE(<i>codepage</i>)	✓			

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 (see [UTT Files](#) for information on 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**.)

Default

The default code page 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.

ENCRYPTION_KEY - UENCRYPT configuration option

Description

The ENCRYPTION_KEY option specifies the key used to encrypt the command file.

Note

 The key specified by this ENCRYPTION_KEY option also must be provided to the Universal Agent component command for which the command file is intended.

If you specify an encryption key with this option, and the [Universal Broker Key Store](#) contains an encryption key, UENCRYPT uses the encryption key specified with this option.

If you do not specify an encryption key with this option, UENCRYPT will use an application-provided default encryption key or the encryption key stored in the Universal Broker Key Store.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-K <i>key</i>		✓	✓	✓
Command Line, Long Form	-key <i>key</i>		✓	✓	✓
STRUEN Parameter	KEY(<i>key</i>)	✓			

Values

key is the key used to encrypt the command file.

key can contain from 1 to 32 characters long. However, it is recommended that *key* contain at least 8 characters.

GENERATE_KEY - UENCRYPT configuration option

Description

The GENERATE_KEY option specifies whether or not to generate an encryption key.

GENERATE_KEY either writes a generated encryption key to the *local* Universal Broker key store specified by the [KEYSTORE_PATH](#) Universal Encrypt configuration option or, if the [STORE_KEY](#) Universal Encrypt configuration option is yes, to a *remote* key store location specified by the [KEYSTORE_PATH](#) Universal Broker configuration option.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-g <i>option</i>		✓	✓	✓
Command Line, Long Form	-genkey <i>option</i>		✓	✓	✓
STRUEN Parameter	n/a				

Values

option is the specification for whether or not to generate an encryption key.

Valid values for *option* are:

- **yes**
Generate an encryption key.
- **no**
Do not generate an encryption key.

Default is yes.

HELP - UENCRYPT configuration option

Description

The HELP option displays a description of the Universal Encrypt 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		✓	✓	✓
STRUEN Parameter	n/a				

Values

(There are no values used with this option.)

INPUT_FILE - UENCRYPT configuration option

Description

The INPUT_FILE option specifies the input file that is to be encrypted.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	n/a				
STRUEN Parameter	INFILE(<i>input_file</i>) [INMBR (<i>member</i>)]	✓			

Values

input_file is the name of the file to be encrypted.

Valid values for *input_file* are:

- ***STDIN**
Input file is read from standard input. 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*
Input file is read from the specified file.

file_name can be qualified by a library name. Otherwise, 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 **INMBR** parameter.

KEYSTORE_PATH - UENCRYPT configuration option

Description

The KEYSTORE_PATH option specifies the path to the local Universal Broker key store.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-p <i>path</i>		✓	✓	✓
Command Line, Long Form	-keypath <i>path</i>		✓	✓	✓
STRUEN Parameter	n/a				

Value

path is the path to the local Universal Broker key store.

NLS_DIRECTORY - UENCRYPT configuration option

Description

The NLS_DIRECTORY option specifies the name of the directory where the code page UTT files are located.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-nlsdir <i>directory</i>		✓	✓	
STRUEN Parameter	n/a				

Values

directory is the name of the directory.

Relative path names are relative to the installation directory. Full path names are recommended.

OUTPUT_FILE - UENCRYPT configuration option

Description

The OUTPUT_FILE option specifies the file to which the encrypted input file is written.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	n/a				
STRUEN Parameter	OUTFILE(<i>output_file</i>) [OUTMBR(<i>member</i>)]	✔			

Values

output_file is the name of the file which the input file is written.

Valid values for *output_file* are:

- ***STDOUT**
Encrypted input file is written to stdout. Standard output is allocated to the terminal if STRUEN is executed interactively. The ILE session terminal is displayed to view the output. Standard output is allocated to file **QPRINT** if STRUEN is executed in batch.
- *file_name*
Encrypted input file is written to the specified file.

file_name can be qualified by a library name. Otherwise, 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 **OUTMBR** parameter.

STORE_KEY - UENCRYPT configuration option

Description

The STORE_KEY option specifies whether or not to store the encryption key (generated or specified explicitly) in a remote Universal Broker key store specified by the Universal Broker [KEYSTORE_PATH](#) configuration option.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-s <i>option</i>		✓	✓	✓
Command Line, Long Form	-store <i>option</i>		✓	✓	✓
STRUEN Parameter	n/a				

Values

option is the specification for whether or not to store the encryption key in the key store of the local Universal Broker.

Valid values for *option* are:

- **yes**
Store the encryption key.
- **no**
Do not store the encryption key.

Default is yes.

VERSION - UENCRYPT configuration option

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-v		✓	✓	✓
Command Line, Long Form	-version		✓	✓	✓
STRUEN Parameter	VERSION(<i>option</i>)	✓			

Values

There are no values for this option.

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 Event Log Dump

- [Universal Event Log Dump](#)
- [Usage](#)
- [Detailed Information](#)
- [Universal Event Log Dump Examples](#)

Universal Event Log Dump

Universal Event Log Dump (UELD) is a utility that selects records from one of the Windows event logs and writes them to a specified output file.

All records from a log can be dumped, or event records can be selected according to the date and time that they were generated.

UELD can be run any time as a stand-alone application. It also is designed to work with Universal Command, which provides centralized control from any operating system and additional options for redirecting output.

Usage

Universal Event Log Dump consists of the command line program (**ueld**) followed by a list of configuration options.

Detailed Information

The following pages provide detailed information for Universal Event Log Dump:

- [Universal Event Log Dump - Configuration](#)
- [Universal Event Log Dump - Command Line Syntax](#)
- [Universal Event Log Dump Configuration Options](#)

Universal Event Log Dump Examples

See [Windows Event Log Dump - Examples](#) for examples of how to run Universal Event Log Dump.

Universal Event Log Dump - Configuration

- [Configuration](#)
- [Configuration Sources](#)
 - [Configuration File](#)
- [Configuration Options Categories](#)
- [Local Category Options](#)
- [Log Category Options](#)
- [Message Category Options](#)
- [Miscellaneous Category Options](#)
- [Output Category Options](#)

Configuration

Configuration consists of:

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

Configuration Sources

Configuration options are read from the following sources:

1. Command line
2. 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 the configuration file.

Configuration File

The configuration file, **ueld.conf**, provides the simplest method of specifying configuration options whose values will not change 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 Categories

The following table categorizes the options used to execute Universal Event Log Dump into logical areas of application. Each **Category** name is a link to a table of options in that category. Each **Option Name** is a link to detailed information about that option.

Category	Description
Local	Options required for local broker registration.
Log	Event records to select from which event log and what actions should be taken.
Messages	Utility message options.
Miscellaneous	Options use to display command help and program versions.
Output	How the event log records are written.

Local Category Options

Option Name	Description
INSTALLATION_DIRECTORY	Base directory where product is installed.
NLS_DIRECTORY	UMC and UTT file directory.

Log Category Options

Option Name	Description
BACKUP_LOG	Causes the log to be backed up before it is cleared.
CLEAR_LOG	Causes the records in the log to be deleted from the log.
END_TIME	Ending date and time.
LOG_TYPE	Event log to be dumped.
REMOTE_SERVER	Name of a remote computer from which event log records should be retrieved.
START_TIME	Starting date and time.

Message Category Options

Option Name	Description
LOG_DIRECTORY	Log file directory.
MESSAGE_DESTINATION	Location where messages are written.
MESSAGE_LANGUAGE	Universal Message Catalog (UMC) file that will be used to write messages.
MESSAGE_LEVEL	Level of messages that will be displayed.

Miscellaneous Category Options

Option Name	Description
HELP	Writes a description of the command options and their format.
VERSION	Writes the program version and copyright information.

Output Category Options

Option Name	Description
OUTPUT_FILE	Complete path to the file that will be used to store the selected event log records.
PAGE_HEADERS	Enables or disables the printing of page headers.
PAGE_LENGTH	Number of lines that should be printed on each page.

Universal Event Log Dump - Command Line Syntax

Command Line Syntax

The following figure illustrates the command line syntax - using the long form of command line options - of UELD.

```
ueld
[-logtype {system|application|security}]
[-clear [-backupfilename] ]
[-stimestartdate [,starttime] ]
[-etimeenddate [,endtime] ]
[-serverservername]
[-filefilename]
[-header {yes|no}]
[-lengthpagelength]
[-level {trace|audit|info|warn|error}]
[-dest {stderr|logfile}]
[-langlanguage]

ueld
{ -help | -version }
```

Universal Event Log Dump Configuration Options

Universal Event Log Dump Configuration Options

This page provides links to detailed information on the configuration options available for use with Universal Event Log Dump.

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

Configuration Options List

The following table identifies the Universal Event Log Dump configuration options.

Option Name	Description
BACKUP_LOG	Causes the log to be backed up before it is cleared.
CLEAR_LOG	Causes the records in the log to be deleted from the log.
END_TIME	Ending date and time.
HELP	Displays a description of command line options and their format.
LOG_DIRECTORY	Log file directory.
LOG_TYPE	Event log to be dumped.
MESSAGE_DESTINATION	Location where messages are written.
MESSAGE_LANGUAGE	Universal Message Catalog (UMC) file that will be used to write messages.
MESSAGE_LEVEL	Level of messages that will be displayed.
INSTALLATION_DIRECTORY	Base directory where product is installed.
NLS_DIRECTORY	UMC and UTT file directory.
OUTPUT_FILE	Complete path to the file that will be used to store the selected event log records.
PAGE_HEADERS	Enables or disables the printing of page headers.
PAGE_LENGTH	Number of lines that should be printed on each page.
REMOTE_SERVER	Name of a remote computer from which event log records should be retrieved.
START_TIME	Starting date and time.
VERSION	Writes the program version and copyright information.

BACKUP_LOG - UELD configuration option

Description

The BACKUP_LOG option causes the log (the System event log or the log specified by the LOG_TYPE option) to be backed up before it is cleared via the CLEAR_LOG option.

BACKUP_LOG is valid only when used with CLEAR_LOG.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-b <i>filename</i>				✓	
Command Line, Long Form	-backup <i>filename</i>				✓	
Configuration File Keyword	n/a					

Values

filename is the name of the log to be backed up.

CLEAR_LOG - UELD configuration option

Description

The CLEAR_LOG option causes the records in the log (the System event log or the log specified by the [LOG_TYPE](#) option) to be deleted from the log.

When CLEAR_LOG used with other options (except the [BACKUP_LOG](#) option), the log first is dumped, then cleared.

Note



Administrator access is required to dump any of the event logs.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-c				✓	
Command Line, Long Form	-clear				✓	
Configuration File Keyword	n/a					

Values

(There are no values for this option.)

END_TIME - UELD configuration option

Description

The END_TIME option specifies the ending date and time of records to be dumped.

If END_TIME is not used, Universal Event Log Dump will select records up to and including the last (that is, the most recent) record in the log specified by the LOG_TYPE option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-e <i>enddate</i> [<i>,endtime</i>]				✓	
Command Line, Long Form	-etime <i>enddate</i> [<i>,endtime</i>]				✓	
Configuration File Keyword	n/a					

Values

enddate is the ending date of records to be dumped.


Event log records for the current date can be selected by using an asterisk (*).

If Event Log Dump will run over consecutive days, a rolling date can be specified with an asterisk (*) followed by a negative value. For example, *-2 selects records that were generated prior to 2 days before the current date.

endtime, optionally, specifies the ending time of records to be dumped.

If *endtime* is omitted, a value of **23:59** is used.

Note

 *enddate* and *endtime* must match the short date and short time styles, respectively, as specified in the Control Panel.

If a 12-hour time format is used, and *endtime* is specified, *enddate* and *endtime* together must be enclosed in double (") quotation marks (for example, "12/31/2011, 11:59 PM"). This ensures that the value will be read correctly from the command line.

HELP - UELD configuration option

Description

The HELP option displays a description of the Universal Event Log Dump command line options and their format.

Usage

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

Values

(There are no values for this option.)

INSTALLATION_DIRECTORY - UELD configuration option

Description

The INSTALLATION_DIRECTORY option specifies the Universal Event Log Dump base installation directory

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	n/a					
Configuration File Keyword	installation_directory <i>directory</i>					

Value

directory is the name of the Universal Event Log Dump base installation directory.

Specify a full path name.

LOG_DIRECTORY - UELD configuration option

Description

The LOG_DIRECTORY option specifies the directory name where log files are created.

Log file creation is specified by the [MESSAGE_DESTINATION](#) option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	n/a					
Configuration File Keyword	log_directory <i>directory</i>				✓	

Value

directory is the directory where log files are created.

LOG_TYPE - UELD configuration option

Description

The LOG_TYPE option specifies the event log to be dumped.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-t <i>logtype</i>				✓	
Command Line, Long Form	-logtype <i>logtype</i>				✓	
Configuration File Keyword	logtype <i>logtype</i>				✓	

Values

logtype is the event log to be dumped.

Valid values for *logtype* are:

- **system**
Dumps records from the System event log.
- **application**
Dumps records from the Application event log.
- **security**
Dumps records from the Security event log.

Note



Administrator access is required for this dump.

Default is system.

MESSAGE_DESTINATION - UELD configuration option

Description

The MESSAGE_DESTINATION option specifies the location where messages are to be written.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-d <i>destination</i>				✓	
Command Line, Long Form	-dest <i>destination</i>				✓	
Configuration File Keyword	message_dest <i>destination</i>				✓	

Values

destination is the location where messages are to be written.

Valid values for destination are:

- **stderr**
Writes the messages to the console.
- **logfile**
Write the messages to a log file. The log file location is located beneath the **ueld** folder, in the log folder. The current log file name is **unv.log**.

Past generation log files are named **unvNNNN.log**, where **NNNN** equals the generation number. Currently, five generations are kept.

Default is stderr.

MESSAGE_LANGUAGE - UELD configuration option

Description

The MESSAGE_LANGUAGE option specifies the Universal Message Catalog (UMC) file that will be used to write messages.

Each UMC file contains messages for a specific language.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-L <i>language</i>				✓	
Command Line, Long Form	-lang <i>language</i>				✓	
Configuration File Keyword	language <i>language</i>				✓	

Values

language is the UMC file that will be used to write messages.

The first three characters of the language name are used as a three-character suffix in the UMC file base name (for example, **uelmceng.umc**). All UMC files have a **.umc** extension.

Default is uelmceng.umc.

MESSAGE_LEVEL - UELD configuration option

Description

The MESSAGE_LEVEL option specifies the level of messages to be written.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l <i>level</i>				✓	
Command Line, Long Form	-level <i>level</i>				✓	
Configuration File Keyword	n/a					

Values

level is the level of messages to be written.

Valid values for *level* are:

- **trace**
Writes trace messages used for debugging. The trace file name is **ueld.trc**. It is created in the directory where the **ueld** program is located. Use only as directed by Stonebranch Customer Support.

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 warn.

NLS_DIRECTORY - UELD configuration option

Description

The NLS_DIRECTORY option specifies the directory name where the Universal Event Log Dump message catalog and code page tables are located.

Usage

Method	Syntax*	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	n/a					
Configuration File Keyword	nls_directory <i>directory</i>					

Values

directory is the name of the directory where the files are located.

OUTPUT_FILE - UELD configuration option

Description

The OUTPUT_FILE option specifies the complete path to the file that will be used to store the selected event log records.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-f <i>filename</i>				✓	
Command Line, Long Form	-file <i>filename</i>				✓	
Configuration File Keyword	n/a					

Values

filename is the complete path to the file that will be used to store the selected event log records.

PAGE_HEADERS - UELD configuration option

Description

The PAGE_HEADERS option enables or disables the writing of page headers.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-r <i>option</i>				✓	
Command Line, Long Form	-header <i>option</i>				✓	
Configuration File Keyword	header <i>option</i>				✓	

Values

option is the specification for enabling or disabling the writing of page headers.

Valid values for *option* are:

- **yes**
Display column headings at the top of each page.
- **no**
Display report columns without headings.

Default is no.

PAGE_LENGTH - UELD configuration option

Description

The PAGE_LENGTH option specifies the number of lines (records) to be written on each page.

If PAGE_LENGTH is not used, and PAGE_HEADERS is set to **yes**, a header will be written only at the top of the first page (since there is, in effect, only one page).

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-n <i>pagelength</i>				✓	
Command Line, Long Form	-length <i>pagelength</i>				✓	
Configuration File Keyword	length <i>pagelength</i>				✓	

Values

pagelength is the number of lines (records) to be written on each page.

REMOTE_SERVER - UELD configuration option

Description

The REMOTE_SERVER option specifies the name of a remote computer from which event log records should be retrieved.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-S <i>servername</i>				✓	
Command Line, Long Form	-server <i>servername</i>				✓	
Configuration File Keyword	n/a					

Values

servername is the name of a remote computer from which event log records should be retrieved.

servername must be specified using the Universal Naming Convention (UNC) format, where the computer name is preceded by two back slashes (for example, \\RMT1).

START_TIME - UELD configuration option

Description

The `START_TIME` option specifies the starting date and time of records to be dumped.

If `START_TIME` is not used, Universal Event Log Dump will start with the first (that is, the earliest) record in the log specified by the `LOG_TYPE` option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-s startdate [,starttime]</code>				✓	
Command Line, Long Form	<code>-stime startdate [,starttime]</code>				✓	
Configuration File Keyword	n/a					

Values

startdate is the starting date of records to be dumped.

Event log records for the current date can be selected by using an asterisk (*).

If Event Log Dump will run over consecutive days, a rolling date can be specified with an asterisk (*) followed by a negative value. For example, `*-2` selects records that were generated in the previous 2 days.

starttime, optionally, specifies the starting time of records to be dumped.

If *starttime* is omitted, a value of `00:00` (midnight) is used.

VERSION - UELD configuration option

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-v				✓	
Command Line, Long Form	-version				✓	
Configuration File Keyword	n/a					

Values

(There are no values for this option.)

Universal FTP Client

- [Overview](#)
- [Detailed Information](#)

Overview

The Universal FTP Client utility (UFTP) transfers files across a network.

UFTP is implemented on top of the widely-used libcurl, which offers support for transfers over FTP, FTP with SSL/TLS (FTPS), FTP over SSH (SFTP), Trivial FTP (TFTP).

UFTP is designed to adhere to standards outlined by RFC 959, and is intended to work with any FTP server that also adheres to those guidelines. In such cases where an FTP server implementation may deviate from established standards, UFTP is flexible enough to accommodate such deviations.

UFTP eliminates the Universal Agent's reliance on 3rd-party FTP clients, which existed prior to the 6.4.0.0 release. Results from these external clients were often unpredictable, and handling variations required frequent Universal Agent changes. UFTP introduces a measure of stability and predictability along with a clean interface between it and the Universal Agent server (UAGSRV).

While UFTP can run directly from the command line, it is frequently executed via [File Transfer](#) tasks defined in the Universal Controller. Starting with Universal Agent 6.4.0.0, FTP File Monitor tasks and File Transfer tasks (excluding those with a type of **UDM**) execute UFTP.

UFTP is provided with UNIX, Windows, and z/OS agents.

Detailed Information

The following pages provide detailed information for Universal FTP Client:

- [Universal FTP Client Commands](#)
- [Command Examples](#)
- [Specifying Multiple Files in Commands](#)
- [Universal FTP Client Configuration Options](#)

Universal FTP Client Commands

- [Overview](#)
- [Command Line Syntax](#)
- [Configuration Options for Sending Commands](#)

Overview

Use the [COMMAND](#) configuration option to specify the transfer operation for UFTP.

[QUOTE](#) and [SITE](#) options also can be specified to send transfer directives or instructions directly to the FTP server.

Available COMMANDs are listed below.

Command Name	Description
GET	Retrieves a single file from an FTP Server to Universal FTP Client.
MGET	Retrieves multiple files from an FTP Server to Universal FTP Client.
PUT	Transfers a single file from Universal FTP Client to an FTP Server.
MPUT	Transfers multiple files from Universal FTP Client to an FTP Server.
DELETE	Deletes a single file on a remote host.
MDELETE	Deletes multiple files on a remote host.
MKDIR	Creates a directory on a remote host.
RMDIR	Removes a directory on a remote host.
LIST	Lists files in a directory.

Command Line Syntax

The following figure illustrates the command line syntax - using the long form of command line options - of Universal FTP Client.

```
uftp
-host[protocol://]ipaddr [-portport] [-protocol{ FTP | FTPS | SFTP | TFTP }]
-useridname-passwordpassword
[-cmd{ GET | MGET | PUT | MPUT | DELETE | MDELETE | MKDIR | RMDIR | LIST } | [-quote cmd] [-site cmd]
[-cmdfile filename] | [-encryptedfile file] [-kkey] ]
[-srcfilepath] [-dstpath]
[-data_type type]
[{-actvipaddress:port | -epsv | -pasv } ]
[-keypath [-key_passphrase phrase] ]
[-ssl_cipher_list list]
[-min_ssl_protocol { TLS1_0 | TLS1_2 } ]
[-certfilepath] [-private_keyfilepath [-private_key_pwd password] ]
[-ca_certsfilepath] [-crlfilepath]
[-nosize] [-regex] [-unhide]
[-list_parserfilepath]
```

Configuration Options for Sending Commands

UFTP provides commands and transfer instructions to the FTP server using the [COMMAND](#), [QUOTE](#), and [SITE](#) options. [COMMAND](#) typically is used, but [QUOTE](#) and/or [SITE](#) also can be specified, depending on the requirements of a particular transfer.

- [COMMAND](#)
For sending generic, protocol-agnostic, OS-agnostic commands (although not all protocols support all commands; for example, TFTP does not support LIST).
- [QUOTE](#)
For sending protocol-specific commands (different commands for FTP(S) and SFTP).

- [SITE](#)
For sending host-specific commands (z/os server has its own set of commands).

DELETE - UFTP command

- [Description](#)
- [Configuration Options](#)

Description

The DELETE command deletes a single file on a remote host.

Configuration Options

The following configuration options are used with DELETE:

Required	SOURCE
Optional	MOVE

GET - UFTP command

- [Description](#)
- [Configuration Options](#)

Description

The GET command retrieves a single file from an FTP Server to Universal FTP Client.

Configuration Options

The following configuration options are used with GET:

Required	SOURCE
Optional	DESTINATION MOVE

LIST - UFTP command

- [Description](#)
- [Configuration Options](#)


Description

The LIST command lists files in a directory.

By default, if the FTP server is running under an operating system that does not provide file sizes with lists, UFTP will calculate the file size by transferring the file to the client and counting the bytes received. The file contents are not saved.

Although this provides an accurate size for files, it comes with a cost in terms of expending the resources necessary for this. Using the [NO_SIZE](#) bypasses this operation. The file sizes will not be available, but the overhead will have been bypassed. If [NO_SIZE](#) is used with a server that provides sizes, it has no effect. It is effective for a z/OS FTP server.

Note

 z/OS is the only FTP server operating system known to not provide file size.

Configuration Options

The following configuration options are used with LIST:

Required	(none)
Optional	NO_SIZE SOURCE

Note

 Do not use the LIST command with the [QUOTE](#) option; this will not work correctly with `libcurl`.

MDELETE - UFTP command

- [Description](#)
- [Configuration Options](#)

Description

The MDELETE command deletes multiple files on a remote host.

(See [Specifying Multiple Files in Commands](#).)

Configuration Options

The following configuration options are used with MDELETE:

Required	SOURCE
Optional	(none)

The [DESTINATION](#) option is not specified.

MGET - UFTP command

- [Description](#)
- [Configuration Options](#)

Description

The MGET command retrieves multiple files from an FTP Server to Universal FTP Client.

(See [Specifying Multiple Files in Commands.](#))

Configuration Options

The following configuration options are used with MGET:

Required	SOURCE
Optional	DESTINATION

MKDIR - UFTP command

- [Description](#)
- [SOURCE / DESTINATION Options](#)

Description

The MKDIR command creates a directory on a remote host.

SOURCE / DESTINATION Options

The following configuration options are used with MKDIR:

Required	SOURCE
Optional	(none)

The [DESTINATION](#) option is not specified.

MPUT - UFTP command

- [Description](#)
- [Configuration Options](#)

Description

The MPUT command transfers multiple files from Universal FTP Client to an FTP Server.

(See [Specifying Multiple Files in Commands](#).)

Configuration Options

The following configuration options are used with MPUT:

Required	SOURCE
Optional	DESTINATION

PUT - UFTP command

- [Description](#)
- [Configuration Options](#)

Description

The PUT command transfers a single file from Universal FTP Client to an FTP Server.

Configuration Options

The following configuration options are used with PUT:

Required	SOURCE
Optional	DESTINATION

RMDIR - UFTP command

- [Description](#)
- [Configuration Options](#)

Description

The RMDIR command removes a directory on a remote host.

Configuration Options

The following configuration options are used with RMDIR:

Required	SOURCE
Optional	(none)

The [DESTINATION](#) option is not specified.

Command Examples

- [GET Command](#)
- [PUT Command](#)
- [MGET \(Wildcard\)](#)
- [MGET \(Regular Expression\)](#)
- [MPUT \(Wildcard\)](#)
- [MPUT \(Regular Expression\)](#)
- [DELETE](#)
- [MDELETE \(Wildcard\)](#)
- [MDELETE \(Regular Expression\)](#)
- [MKDIR](#)
- [RMDIR](#)
- [LIST](#)

GET Command

The following example is for a GET command with absolute paths and destination filename specified:

```
$ uftp -host ftp://sb-123-x64 -user test -pwd test -f /home/build1/workspaces/ftp-client/GET_Command
```

Command file contents:

```
-cmd GET -src /qa/durability_export/readme -dst /home/build1/workspaces/ftp-client/uftp/GET_test
```

PUT Command

The following example is a PUT command with relative paths and using source filename in destination:

```
$ uftp -host ftp://sb-123-x64 -user test -pwd test -f /home/build1/workspaces/ftp-client/PUT_Command
```

Command file contents:

```
-cmd PUT -src uftp/put_test -dst durability_export
```

MGET (Wildcard)

The following example is for an MGET wildcard command with no paths:

```
$ uftp -host ftp://sb-123-x64 -user test -pwd test -f /home/build1/workspaces/ftp-client/MGET_Command_Wildcard
```

Command file contents:

```
-cmd MGET -src ?mit.???*
```

MGET (Regular Expression)

The following example is for an MGET Regular Expression command with absolute paths:

```
$ uftp -host ftp://sb-123-x64 -user test -pwd test -cmd MGET -src /qa/sA-Cstore.d(b|b.orig) -dst /home/build1/workspaces/ftp-client -regex
```

MPUT (Wildcard)

The following example is for an MPUT Wildcard command with relative paths:

```
$ uftp -host ftp://sb-123-x64 -user test -pwd test -f /home/build1/workspaces/ftp-client/MPUT_Command_Wildcard
```

Command file contents:

```
-cmd MPUT -src ftp-client/uftp/?mit.???* -dst qa
```

MPUT (Regular Expression)

The following example is for an MPUT Regular Expression command with no paths:

```
$ uftp -host ftp://sb-123-x64 -user test -pwd test -f /home/build1/workspaces/ftp-client/MPUT_Command_Regex
```

Command file contents:

```
-cmd MPUT -src r-tmit.(script|transaction) -regex
```

DELETE

The following example is for a DELETE command with an absolute path:

```
$ uftp -f /home/build1/workspaces/ftp-client/FTP_DELETE_Command
```

Command file contents:

```
-host ftp://sb-123-x64 -src /qa/durability_export/ftp-client_test_file -user test -password test -cmd DELETE
```

MDELETE (Wildcard)

The following example is for an MDELETE Wildcard command with a relative path:

```
$ uftp -f /home/build1/workspaces/ftp-client/FTP_MDELETE_Command_Wildcard
```

Command file contents:

```
-host ftp://sb-123-x64 -src durability_export/?mit.???* -user test -password test -cmd MDELETE
```

MDELETE (Regular Expression)

The following example is for an MDELETE Regular Expression command with no path:

```
$ uftp -f /home/build1/workspaces/ftp-client/FTP_MDELETE_Command
```

Command file contents:

```
-host ftp://sb-123-x64 -src r-tmit.(script|transaction) -user test -password test -cmd MDELETE -regex
```

MKDIR

The following example is for an MKDIR command with an absolute path:

```
$ uftp -f /home/build1/workspaces/ftp-client/FTP_MKDIR_Command
```

Command file contents:

```
-host ftp://sb-123-x64 -src /qa/durability_export/ftp-client/ -user test -password test -cmd MKDIR
```

RMDIR

The following example is for a RMDIR command with a relative path:

```
$ uftp -f /home/build1/workspaces/ftp-client/FTP_RMDIR_Command
```

Command file contents:

```
-host ftp://sb-123-x64 -src ftp-client/ -user test -password test -cmd RMDIR
```

LIST

The following example is for a LIST command with no path:

```
$ uftp -host ftp://sb-123-x64 -user test -pwd test -cmd LIST
```

Specifying Multiple Files in Commands

Methods of Specifying Files

There are four methods available to specify multiple files in the [LIST](#), [MGET](#), [MPUT](#), and [MDELETE](#) UFTP commands.

Method	Example
Using wildcards.	\$ MGET *.htm?
Using Regular Expressions (regex).	\$ MDELETE smit.(script transaction)
Providing an explicit comma-delimited list.	\$ MPUT options.cfg data.txt,spreadsheet.xls
Using a single literal filename (no pattern matching).	\$ MGET smit.script

For [MGET](#), the [-dst](#) configuration option specifies the local directory to save the transferred files.

For [MPUT](#), the [-dst](#) configuration option specifies the remote directory to save the transferred files.

Note



Wildcard matching and Regular Expression matching are not compatible due to the use by each of the `*` and `?` matching characters. The default is wildcard matching. The [REGEX](#) configuration option lets you select Regular Expression matching for [MGET](#), [MPUT](#), and [MDELETE](#).

Universal FTP Client Configuration Options

- [Universal FTP Client Configuration Options](#)
- [Configuration Options](#)

Universal FTP Client Configuration Options

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

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

Configuration Options

The following table identifies the Universal FTP Client configuration options.

Option Name	Description
ACTIVE_MODE	Specification that the data connection mode of the FTP Client server is Active.
AUTHENTICATE_PEER	Specification for whether or not UFTP will certify that an FTP server's certificate was issued by a known and trusted authority.
BIF_DIRECTORY	Broker Interface Directory (BIF) directory where the Universal Broker interface file, ubroker.bif , is located.
CA_CERTIFICATES	Location of a PEM-formatted trusted CA x.509 certificate.
CERTIFICATE	Location of a PEM-formatted X-509 certificate (that is, a public key).
CERTIFICATE_EXPIRATION_NOTICE	Number of days prior to certificate expiration to begin issuing informational messages about the expiration.
CERTIFICATE_REVOCATION_LIST	Location of a PEM-formatted certificate revocation list.
COMMAND	Command to be executed by the host.
COMMAND_FILE	File (ddname for z/OS) of a plain text command file containing one or more configuration options.
DATA_TYPE	Modifies the behavior of GET/MGET and PUT/MPUT by explicitly setting the data representation type used for data transfer.
DESTINATION	Destination path of the file being transferred.
ENABLE_SSL	Specification for whether or not UFTP will request an explicit FTPS session to the server specified by the HOST option.
ENCRYPTED_COMMAND_FILE	File / ddname containing encrypted values for command line options.
ENCRYPTION_KEY	Key that is used to decrypt the file specified by the ENCRYPTED_COMMAND_FILE option.
EXTENDED_PASSIVE_MODE	Specification that the data connection mode of the FTP Client server is Extended Passive.
HELP	Displays a description of the command line options and their format.
HOST	Name of the remote host to connect to on which the command is to be run.
KEY_PASSPHRASE	Passphrase which may have been used to secure a private key file used in SFTP transfers.
LIST_PARSER	Path to a file containing entries that describe the format of file listings received from FTP servers, based on operating system and/or file system.
MESSAGE_LEVEL	Level of messages written.
MIN_SSL_PROTOCOL	Minimum SSL/TLS protocol level that will be negotiated and used for communication channels.
MOVE	Modifies the behavior of GET / MGET and PUT / MPUT commands by deleting the source file after it has been transferred to the destination.
NO_SIZE	Disables the GET command File Size operation.
PASSIVE_MODE	Specification that the data connection mode of the FTP Client server is Passive.

PASSWORD	Password to use in authentication.
PLF_DIRECTORY	Program Lock File (PLF) directory where the program lock files are located.
PORT	Port number of the remote port to connect to.
PRIVATE_KEY	Location of a PEM-formatted RSA private key.
PRIVATE_KEY_PWD	The (optional) password required to access the file specified by PRIVATE_KEY .
PROTOCOL	Default protocol to be used for a transfer, if HOST and COMMAND configuration options have been specified.
QUOTE	Specification to send a command string directly to the host.
REGEX	Specification to use of Regular Expression matching for the MGET, MPUT, and MDELETE commands.
SITE	Formats directory listings (LIST) so that they can be parsed by the FTP Client.
SOURCE	Source directory of the file being transferred.
SSH_KEY	Path to the SSH key file.
SSL_CIPHER_LIST	List of SSL/TLS ciphers used to secure control and data sessions for FTPS transfers.
SYSTEM_ID	Local Universal Broker with which UFTP must register before it performs any request.
UNHIDE	Specification for UFTP to include any hidden files in files reported to the user.
USER	Remote user name to use in authentication.
VERIFY_HOST_NAME	Specification for whether or not UFTP will require identifying information in an FTP server's certificate that corresponds to the specified HOST value.
VERSION	Writes the program version and copyright information.

ACTIVE_MODE - UFTP configuration option

Description

The ACTIVE_MODE option specifies that the data connection mode of the FTP Client server is Active.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-actv (<i>ip address:port</i>)			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

ip address:port specifies the IP address and port of the Active mode FTP Client.

This value is optional; if it is not specified, the FTP Client defaults to `eth0:0`, which sends the ipv address of the Client's `eth0` connection to the server. Port 0 causes the FTP Client to select an available port for use.

AUTHENTICATE_PEER - UFTP configuration option

Description

The AUTHENTICATE_PEER option specifies whether UFTP will certify that an FTP server's certificate was issued by a known and trusted authority.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-authenticate_peer <i>opt</i>			✓	✓	✓
Environment Variable	UFTPAUTHENTICATEPEER= <i>opt</i>			✓	✓	✓
Configuration File Keyword	authenticate_peer <i>opt</i>			✓	✓	✓

Value

opt specifies whether UFTP will certify the certificate presented by the FTP server during a TLS/SSL handshake.

opt can be:

- YES - the FTP server's certificate must be authenticated by one of the trusted authorities in the file specified by the CA_CERTIFICATES option.
- NO - UFTP will allow a connection to a TLS/SS-enabled FTP server without verifying the FTP server's certificate.

Default is no for the FTP [PROTOCOL](#) option.

This option is not used when [PROTOCOL](#) is SSH (which doesn't use TLS/SSL) or when it is FTPS (which requires peer authentication).

Notes for Explicit FTPS (FTPES) Support

Explicit FTPS support was added to UFTP for Universal Agent 7.1.0.0. When an FTP server enables *explicit* TLS/SSL support, it allows clients to request encrypted sessions over the standard FTP port 21. This is different from *implicit* TLS/SSL support, which requires a connection to a unique, well-known port (990 by default) that ONLY accepts TLS/SSL-enabled client requests.

UFTP has always offered implicit FTPS support via the FTPS [PROTOCOL](#) value.

Because an FTP client may not always know whether the FTP server's port 21 is accepting SSL-enabled connections (i.e., it may only accept unencrypted, plain-text FTP sessions), explicit FTPS is not really a distinct protocol per se. Therefore, UFTP offers explicit FTPS support by applying new options and supported values to the existing FTP [PROTOCOL](#) value.

UFTP requests an explicit FTPS session using the ENABLE_SSL option or by prefixing the HOST option's value with `ftpes://`.

The `-authenticate_peer` option is only configurable for explicit FTPS sessions. When the FTPS [PROTOCOL](#) is used, `-authenticate_peer yes` is implied. The `-authenticate_peer` option is ignored for the SSH [PROTOCOL](#).

BIF_DIRECTORY - UFTP 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	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-bif_directory <i>directory</i>			✓	✓	
Environment Variable	UFTPBIFDIRECTORY= <i>directory</i>			✓	✓	
Configuration File Keyword	n/a					

Values

directory is the name of the BIF directory.

Default is the main installation directory.

CA_CERTIFICATES - UFTP configuration option

Description

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

The file specified by this option can contain several CA certificates concatenated together. This allows UFTP to verify the identity of any number of FTP servers that may present public keys signed by various entities.

This option is only used when the [transfer protocol](#) is FTPS (implicit FTPS) or when the [transfer protocol](#) is FTP with SSL support enabled (explicit FTPS).

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-ca_certs <i>file</i>			✓	✓	
Environment Variable	UFTPCACERTS= <i>file</i>			✓	✓	
Configuration File Keyword	ca_certificates <i>file</i>			✓	✓	

Values

file is the path name of the X.509 certificates file. Relative paths are relative to the current working directory.

CERTIFICATE - UFTP configuration option

Description

The CERTIFICATE option specifies the file / ddname name of the PEM-formatted X.509 certificate that contains the UFTP client's public key. When this option is specified, the PRIVATE_KEY option generally is also required.

This option is used only when the transfer protocol is FTPS.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-cert <i>file</i>			✓	✓	
Environment Variable	UFTPCERT= <i>file</i>			✓	✓	
Configuration File Keyword	certificate <i>file</i>			✓	✓	

Values

file is the path name of the X.509 certificates file. Relative paths are relative to the current working directory.

CERTIFICATE_EXPIRATION_NOTICE - UFTP 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	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-days <i>number</i>			✓	✓	
Environment Variable	UFTPDAYSTILEXPIRE= <i>number</i>			✓	✓	
Configuration File Keyword	days_til_expire <i>number</i>			✓	✓	

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 - UFTP configuration option

Description

The CERTIFICATE_REVOCATION_LIST option specifies the PEM-formatted file that contains a Certificate Revocation List (CRL) issued by a trusted Certificate Authority.

This option is used only when the [transfer protocol](#) is FTPS.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-crl <i>file</i>			✓	✓	
Environment Variable	UFTPCRL= <i>file</i>			✓	✓	
Configuration File Keyword	crl <i>file</i>			✓	✓	

Values

file is the path name of the file containing the CRL. Relative paths are relative to the current working directory.


COMMAND - UFTP configuration option

Description

The COMMAND option specifies the [command](#) to be executed by the host.

A COMMAND option command is processed by the FTP Client, which interprets the command and then configures an appropriate command that is then sent to the host server.

Note

 Protocol-specific commands not supported by COMMAND can be sent directly to the host server using the [QUOTE](#) configuration option.

COMMAND option commands are supported by the following protocols:

- FTP
- FTPS
- SFTP
- TFTP

Command	FTP/S	SFTP	TFTP
GET	✓	✓	✓
PUT	✓	✓	✓
MGET	✓	✓	✓*
MPUT	✓	✓	✓
DELETE	✓	✓	
MDELETE	✓	✓	
MKDIR	✓	✓	
RMDIR	✓	✓	
LIST	✓	✓	

FTP/S differs from FTP only in the use of SSL/TLS to provide security. All commands are implemented exactly the same for these protocols.

TFTP is limited to a subset of the available server commands. Additionally, TFTP supports limited MGET functionality. File lists are supported, but regex and wildcard pattern matching are not supported.

Configuration Requirements

The following table identifies the [SOURCE](#) and [DESTINATION](#) configuration option requirements for each type of Universal FTP Client [Command](#) that can be specified by a COMMAND option:

Command	SOURCE option	DESTINATION option
GET	required	optional
PUT	required	optional
MGET	required	optional
MPUT	required	optional
DELETE	required	
MDELETE	required	
MKDIR	required	
RMDIR	required	
LIST	optional	

SOURCE and DESTINATION Endpoints

SOURCE (-src) and **DESTINATION** (-dst) are the two endpoints involved in a file transfer. The word "endpoint" is used to refer to either a source or a destination. Some operations may only use a single endpoint, the source (see [#Configuration Requirements](#), above).

Endpoints are either files or directories and are represented as strings.

Endpoints use the slash (/) character as a directory separator. Endpoints starting with a slash are "absolute," and are relative to the root of the file system. Endpoints starting with any other character are relative to the user's default directory (home directory). An empty path name is valid, and it refers to the user's default directory (usually the user's home directory).

A file designation may include a directory designation (the path).

An endpoint within a Windows FTP Client should use the backward slash (**) as a directory separator.

An FTP Client running under an operating system other than Windows should use the forward slash (/) as a directory separator.

A source or destination within a remote FTP Server should use the forward slash (/) as a directory separator, regardless of the operating system that the Server is running on.

Sources

Source format is command-dependent.

GET, PUT, DELETE	Source endpoint for these commands must always be a file.
MGET, MPUT, MDELETE	Source endpoint for these commands can be a regular expression (regex), a wildcard pattern, a comma-delimited list of file names, or a single file name (as in GET, PUT, and DELETE).
MKDIR, RMDIR	Source endpoint for these commands must be a directory.
LIST	The optional source endpoint for a LIST command can be a directory, a regular expression (regex), a wildcard pattern, a comma-delimited list of file names, or a single file name. If source is not specified, it defaults to the FTP Server's current working directory.

Destinations

Destination format is command-dependent.

GET, PUT	The optional destination endpoint for these commands can be a file or a directory. If no file name is specified, the transferred file defaults to the name of the source file. If a file name is specified, the transferred file is renamed to the destination file. If no destination is specified, the file is transferred to the current working directory with the same name as the source file.
MGET, MPUT	The optional destination endpoint for these commands must be a directory. Since multiple files can be transferred, each file retains the name of the source file.
DELETE, MDELETE, MKDIR, RMDIR, LIST	These commands never use a destination endpoint.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-c <i>command</i>					
Command Line, Long Form	-cmd <i>command</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

command is the name of the command to execute.

Valid values:

- GET
- MGET
- PUT
- MPUT
- DELETE
- MDELETE
- MKDIR
- RMDIR
- LIST

COMMAND_FILE - UFTP configuration option

Description

The COMMAND_FILE option specifies the file name (ddname for z/OS) of a plain text [command file](#).

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-f <i>file</i>			✓	✓	✓
Command Line, Long Form	-cmdfile <i>file</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

file is the file name (ddname for z/OS) of a plain text [command file](#).

DATA_TYPE - UFTP configuration option

Description

The `DATA_TYPE` option modifies the behavior of the `GET`, `MGET`, `PUT`, and `MPUT` commands by explicitly setting the data representation type used for data transfer.

Example:

```
$ uftp -host ftp://sb-123-x64 -user test -pwd test -cmd MGET -src [r-t]mit.(script|transaction) -regex -move -data_type TEXT
```

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-data_type <i>type</i>			✓	✓	✓
Environment Variable	UFTPDATA_TYPE= <i>type</i>			✓	✓	✓
Configuration File Keyword	n/a					

Value

type is the data representation type used for a transfer.

Valid values:

- TEXT (for ASCII text files)
- BINARY (binary files)

Default is BINARY for FTP(S).

(For SFTP, `DATA_TYPE` is irrelevant.)

DESTINATION - UFTP configuration option

Description

The DESTINATION option specifies the complete path or single file name for the destination file.

DESTINATION is not required for any command.

DESTINATION is optional for the following commands:

- GET
- MGET
- PUT
- MPUT

For GET and PUT commands, DESTINATION can be used to specify the complete path or single file name for the destination file.

If DESTINATION identifies only a file name, the current directory is used for the destination host. If DESTINATION identifies only a directory, the file name specified in the SOURCE option is used.

For MGET and MPUT commands, DESTINATION cannot be used to specify a single file name for the destination file(s).

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-dst <i>path</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

path is the complete path or single file name for the destination file.

Note



For UFTP running under Windows, a DESTINATION (or SOURCE) option value cannot start with / (Windows XPS interprets this as the configuration option prefix: -), but it can start with \.

ENABLE_SSL - UFTP configuration option

Description

The ENABLE_SSL option specifies whether UFTP will request an explicit FTPS session to the server specified by the [HOST](#) option.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-enable_ssl <i>opt</i>			✓	✓	✓
Environment Variable	UFTPENABLESSL= <i>opt</i>			✓	✓	✓
Configuration File Keyword	enable_ssl <i>opt</i>			✓	✓	✓

Value

opt specifies whether UFTP will request an encrypted session with the FTP server that resides on [HOST](#).

opt can be:

- YES - request an encrypted session secured with TLS/SSL when the FTP [PROTOCOL](#) is used.
- NO - create an unencrypted, plain-text FTP session.

Default is no for the FTP [PROTOCOL](#) option.

This option is not used when [PROTOCOL](#) is SSH (which doesn't use TLS/SSL) or when it is FTPS (in which TLS/SSL is required or "implicit").

Notes for Explicit FTPS (FTPES) Support

Explicit FTPS support was added to UFTP for Universal Agent 7.1.0.0. When an FTP server enables *explicit* TLS/SSL support, it allows clients to request encrypted sessions over the standard FTP port 21. This is different from *implicit* TLS/SSL support, which requires a connection to a unique, well-known port (990 by default) that ONLY accepts TLS/SSL-enabled client requests.

UFTP has always offered implicit FTPS support via the FTPS [PROTOCOL](#) value.

Because an FTP client may not always know whether the FTP server's port 21 is accepting SSL-enabled connections (i.e., it may only accept unencrypted, plain-text FTP sessions), explicit FTPS is not really a distinct protocol per se. Therefore, UFTP offers explicit FTPS support by applying new options and supported values to the existing FTP [PROTOCOL](#) value.

Enabling SSL Via the HOST Option

One way that UFTP provides FTPES support is by allowing the [HOST](#) value to be prefixed with `ftpes://`. When this prefix is used, UFTP behaves as though the `-enable_ssl` option were yes.

ENCRYPTED_COMMAND_FILE - UFTP configuration option

Description

The ENCRYPTED_COMMAND_FILE option specifies the encrypted file / ddname containing 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.

Universal FTP Client can process command files that are either encrypted or in plain text (see the [COMMAND_FILE](#) 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. 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](#) option. If it is, the ddname / file specified in [COMMAND_FILE](#) will be used.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-x <i>file</i>			✓	✓	✓
Command Line, Long Form	-encryptedfile <i>file</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

file is the encrypted file / ddname containing command line options.

ENCRYPTION_KEY - UFTP configuration option

Description

The ENCRYPTION_KEY option specifies the key that is used to decrypt the file specified by the [ENCRYPTED_COMMAND_FILE](#) option.

This key is required only if a key was used to encrypt the user file with [Universal Encrypt](#).

If this option is not used, a default key established by the Universal FTP Client utility is used.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-k <i>key</i>			✓	✓	✓
Command Line, Long Form	n/a					
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

key is the key used to decrypt the data set / file.

EXTENDED_PASSIVE_MODE - UFTP configuration option

Description

The EXTENDED_PASSIVE_MODE option specifies that the data connection mode of the FTP server is FTP Extended Passive (ESPV).

If the FTP server does not support ESPV, the [FTP Passive Mode](#) (PASV) is used.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-epsv			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

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

HELP - UFTP configuration option

Description

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

Usage

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

Value

There are no values to be specified for this option.

HOST - UFTP configuration option

Description

The HOST option specifies the name of the remote host to connect to on which the [command\(s\)](#) are to be run.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-i name</code>			✓	✓	✓
Command Line, Long Form	<code>-host name</code>			✓	✓	✓
Environment Variable	<code>UFTPHOST=name</code>			✓	✓	✓
Configuration File Keyword	<code>n/a</code>					

Value

name is name of the remote host to connect to.

name can be:

- Fully qualified domain name of the host
- Local network name of the machine on your network
- IP address

For example:


- `-host dev-server`
- `-host 192.168.33.40`

Optionally, you can include a protocol identifier.

For example:

- `-host ftp://dev-server`

Note

 You also can specify a protocol by using the [PROTOCOL](#) configuration option.

Updates for Explicit FTPS (FTPES) Support

Explicit FTPS support was added to UFTP for Universal Agent 7.1.0.0. When an FTP server enables *explicit* TLS/SSL support, it allows clients to request encrypted sessions over the standard FTP port 21. This is different from *implicit* TLS/SSL support, which requires a connection to a unique, well-known port (990 by default) that ONLY accepts TLS/SSL-enabled client requests.


UFTP has always offered implicit FTPS support via the FTPS protocol value.

Because an FTP client may not always know whether the FTP server's port 21 is accepting SSL-enabled connections (thereby permitting unencrypted, plain-text FTP sessions), explicit FTPS is not really a distinct protocol per se. Therefore, UFTP offers explicit FTPS support by applying new options and supported values to the existing FTP [PROTOCOL](#) value.

Requesting an FTPES Session Via the HOST Option

One way that UFTP provides FTPES support is by allowing the `-host` value to be prefixed with `ftpes://`. UFTP will still use the FTP protocol, but when it will request a session using TLS/SSL with the FTP server. If the FTP server does not have TLS/SSL support enabled, an unencrypted, plain-text session will be created.

Universal Controller Task Support

 The `ftpes://` prefix also enables compatibility with current and previous versions of Universal Controller. The existing File Transfer and Remote File Monitor forms can be used to request an FTPES connection simply by choosing the FTP protocol and updating the FTP Server value with the `ftpes://` prefix.

Requesting an FTPES Session Via New UFTP Options

UFTP also provides explicit FTPS session support with the `ENABLE_SSL` option, new for Universal Agent 7.1.0.0. When this option is **yes**, the `-host` value does not require any special prefixes.

Setting `ENABLE_SSL` to **yes** AND adding the `ftpes://` prefix to the `-host` value is supported, but unnecessary. The `ftpes://` prefix causes UFTP to behave as though `ENABLE_SSL` were set to **yes**.

KEY_PASSPHRASE - UFTP configuration option

Description

The KEY_PASSPHRASE option specifies a passphrase which may have been used to secure a private key file used in SFTP transfers.

It is only used for SFTP transfers when the [SSH_KEY](#) configuration option is specified.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-key_passphrase <i>passphrase</i>			✓	✓	
Environment Variable	UFTPKEYPASSPHRASE= <i>passphrase</i>			✓	✓	
Configuration File Keyword	n/a					

Value

passphrase is the passphrase which may have been used to secure a private key file used in SFTP transfers.

LIST_PARSER - UFTP configuration option

Description

The LIST_PARSER option specifies the path to a file containing entries that describe the format of file listings received from FTP servers, based on operating system and/or file system.

UFTP contains built-in rules for parsing the file lists that it receives from commonly used FTP servers running on the specified platforms. Those built-in rules are intended to handle most FTP server responses. However, situations may arise when an FTP server returns a file list in a non-standard format. Keeping the rules for parsing that output in a separate file makes it possible to change UFTP's behavior without requiring a new version of the application.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-list_parser <i>ddname</i> or <i>filepath</i>			✓	✓	✓
Environment Variable	UFTPLISTPARSER= <i>filepath</i>			✓	✓	✓
Configuration File Keyword	list_parser <i>ddname</i> or <i>filepath</i>			✓	✓	✓

Values

UNIX	<i>filepath</i> is the path name of the file that contains FTP server parsing rules. Relative paths are relative to the current working directory. The default value is parser.txt .
Windows	<i>filepath</i> is the path name of the file that contains FTP server parsing rules. Relative paths are relative to the current working directory. The default value is parser.txt .
z/OS	<i>ddname</i> is the DD name used to point to a dataset that contains FTP server parsing rules. A sequential dataset with a fixed record length of 80 bytes must be allocated to <i>ddname</i> . <i>ddname</i> also can contain instream data with the FTP server parsing rules. The default value is dd:UFTPPARS .

MESSAGE_LEVEL - UFTP configuration option

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l <i>level</i>			✓	✓	✓
Command Line, Long Form	-level <i>level</i>			✓	✓	✓
Environment Variable	UFTPLEVEL= <i>level</i>			✓	✓	✓
Configuration File Keyword	message_level <i>level</i>			✓	✓	✓

Values

level indicates either of the following level of messages:

- **trace**
Writes trace messages used for diagnostic purposes (see [Trace Files](#), below).

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

UNIX	info
Windows	info
z/OS	info

Trace Files

UNIX	<p>If UFTP is executed from a Universal Controller File Transfer or FTP File Monitor task, and the UAGSRV LOGLVL configuration option value is set to T (Trace), a UFTP trace file will be generated with the name <code>uftp_execid.trc</code>, where <code>execid</code> is a generated value that uniquely identifies the task.</p> <p>If UFTP is executed directly from the command line, and its MESSAGE_LEVEL value is trace, a trace file with the name <code>uftp_compuid.trc</code> is created, where <code>compuid</code> is a unique component ID that UFTP receives when it registers with the local Universal Broker.</p>
Windows	<p>If UFTP is executed from a Universal Controller File Transfer or FTP File Monitor task, and the UAGSRV LOGLVL configuration option value is set to T (Trace), a UFTP trace file will be generated with the name <code>uftp_execid.trc</code>, where <code>execid</code> is a generated value that uniquely identifies the task.</p> <p>If UFTP is executed directly from the command line, and its MESSAGE_LEVEL value is trace, a trace file with the name <code>uftp_compuid.trc</code> is created, where <code>compuid</code> is a unique component ID that UFTP receives when it registers with the local Universal Broker.</p>
z/OS	Trace file is written to the data set referenced by the UNVTRACE ddname.

MIN_SSL_PROTOCOL - UFTP configuration option

Description

The MIN_SSL_PROTOCOL option specifies the minimum SSL/TLS protocol level that will be negotiated and used to secure a session for FTPS transfers.

This option is used only when the [transfer protocol](#) is FTPS.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-min_ssl_protocol <i>option</i>			✓	✓	
Environment Variable	UFTPMINSSLPROTOCOL <i>option</i>			✓	✓	
Configuration File Keyword	min_ssl_protocol <i>option</i>			✓	✓	

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.

MOVE - UFTP configuration option

Description

The MOVE option modifies the behavior of the [GET](#), [MGET](#), [PUT](#), and [MPUT](#) commands by deleting the [source](#) file after it has been transferred to the [destination](#).

GET and MGET with MOVE will delete files from the remote FTP server; PUT and MPUT with MOVE will delete files from the local FTP client.

Example:

```
$ uftp -host ftp://sb-123-x64 -user test -pwd test -cmd MGET -src r-tmit.(script|transaction) -regex -move
```

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-move			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

(There are no values used with this option.)

NO_SIZE - UFTP configuration option

Description

The NO_SIZE option is used with the [LIST](#) command; it disables the Get File Size operation.

By default, if the FTP server is running under an operating system that does not provide file sizes with lists, UFTP will calculate the size by transferring the file to the client and counting the bytes received. The file contents are not saved. (Currently, z/OS is the only FTP server operating system known to not provide file size.)

Although this provides an accurate size for files, it comes with a cost in terms of system overhead. Using NO_SIZE bypasses this operation. The file sizes will not be available, but the overhead will have been bypassed.

If NO_SIZE is used with a server that does provide sizes, NO_SIZE has no effect.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-nosize					✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

(There are no values used with this option.)

PASSIVE_MODE - UFTP configuration option

Description

The PASSIVE_MODE option specifies that the data connection mode of the FTP server is FTP Passive (PASV).

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-pasv			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

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

PASSWORD - UFTP configuration option

Description

The PASSWORD option specifies the password to use in authentication.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-w / -pwd <i>password</i></code>			✓	✓	✓
Command Line, Long Form	<code>-password <i>password</i></code>			✓	✓	✓
Environment Variable	<code>UFTPPWD=<i>password</i></code>			✓	✓	✓
Configuration File Keyword	n/a					

Value

password is the password to use in authentication.

PLF_DIRECTORY - UFTP 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 UFTP process to store manager process termination information for the Universal Broker.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-plf_directory <i>directory</i>			✓		
Environment Variable	UFTPPLFDIRECTORY= <i>directory</i>			✓		
Configuration File Keyword	n/a					

Values

directory is the name of the PLF directory.

Default is /var/opt/universal/tmp.

PORT - UFTP configuration option

Description

The PORT option specifies the port number of the remote port to connect to.

PORT, in conjunction with the [HOST](#) option, is used to define a connection to a remote host.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-p <i>port</i>					
Command Line, Long Form	-port <i>port</i>			✓	✓	✓
Environment Variable	UFTPSPORT= <i>port</i>			✓	✓	✓
Configuration File Keyword	n/a					

Value

port is the port number of the remote port to connect to.

PRIVATE_KEY - UFTP 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.

This option is needed only when a certificate is specified by [CERTIFICATE](#).

This option is used only when the [transfer protocol](#) is FTPS.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-private_key <i>file</i>			✓	✓	
Environment Variable	UFTPPRIVATEKEY= <i>file</i>			✓	✓	
Configuration File Keyword	private_key <i>file</i>			✓	✓	

Values

file is the path name of the PEM-formatted RSA private key file that corresponds to the X.509 certificate. Relative paths are relative to the current working directory.

PRIVATE_KEY_PWD - UFTP configuration option

Description

The PRIVATE_KEY_PWD option specifies the password or passphrase for the PEM-formatted RSA private key specified with the [PRIVATE_KEY](#) option.

This option is needed only when a private key file was protected with a password or passphrase upon creation.

This option is used only when the [transfer protocol](#) is FTPS.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	<code>-private_key_pwd <i>password</i></code>			✓	✓	
Environment Variable	<code>UFTPPRIVATEKEYPWD=<i>password</i></code>			✓	✓	
Configuration File Keyword	<code>private_key_password <i>password</i></code>			✓	✓	

Values

password is the password for the private key.

PROTOCOL - UFTP configuration option

Description

The PROTOCOL option specifies a protocol to be used for the transfer.

The protocol also can be specified directly in the [HOST](#) option URL definition.

If a protocol is specified in both the PROTOCOL option and the [HOST](#) option, the [HOST](#) option protocol is used.

If a protocol is not specified in either the PROTOCOL option or [HOST](#) option, UFTP will make a guess based on the host.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-proto <i>protocol</i>			✓	✓	✓
Environment Variable	UFTP_PROTOCOL= <i>protocol</i>			✓	✓	✓
Configuration File Keyword	n/a					

Value

protocol is the protocol to be used for the transfer if none is specified by the [HOST](#) option.

Valid values:

- FTP
- FTPS
 - UFTP offers support for *Implicit SSL/TLS*. This means that UFTP expects the FTP Server to be configured in a way that not only enables SSL/TLS, but also requires that an SSL/TLS handshake be completed before a connection attempt succeeds. FTP servers with implicit SSL/TLS support enabled typically listen for connections over port 990. After the client connects and the SSL/TLS handshake completes, the entire FTP session is encrypted and secured with SSL/TLS. FTP sessions secured with SSL/TLS can be configured using the [SSL_CIPHER_LIST](#), [MIN_SSL_PROTOCOL](#), [CERTIFICATE](#), [PRIVATE_KEY](#), [PRIVATE_KEY_PWD](#), and/or [CA_CERTIFICATES](#) options. Refer to the documentation for each option to understand how UFTP uses SSL/TLS to secure a transfer session.
- SFTP
 - FTP sessions over SSH will require an SSH private key file when key-based authentication is used (as opposed to password-based authentication). See the [SSH_KEY](#) and [KEY_PASSPHRASE](#) options for more information.
- TFTP

Notes for Explicit FTPS (FTPES) Support

Explicit FTPS support was added to UFTP for Universal Agent 7.1.0.0. When an FTP server enables *explicit* TLS/SSL support, it allows clients to request encrypted sessions over the standard FTP port 21. This is different from *implicit* TLS/SSL support described above.

Because an FTP client may not always know whether the FTP server's port 21 is accepting SSL-enabled connections (i.e., it may only accept unencrypted, plain-text FTP sessions), explicit FTPS is not really a distinct protocol per se. Therefore, UFTP offers explicit FTPS support by applying new options and supported values to the existing FTP [PROTOCOL](#) value.

UFTP can also request TLS/SSL support over FTP by prefixing the value specified for the HOST option with `ftpes://`.

QUOTE - UFTP configuration option

Description

The QUOTE option allows a protocol-specific command string to be sent directly to the host.

A QUOTE option command is not processed by the FTP Client. It is sent as-is to the FTP server, which then executes it as if it were entered in the FTP command shell.

You also can specify a command to be executed by the host with the [COMMAND](#) option.

- A QUOTE option command can be sent to the host with or without a [COMMAND](#) option command.
- A QUOTE option command always is executed before a [COMMAND](#) option command.

FTP/S Commands

The complete list of FTP/S server commands is documented in [RFC 5797 - FTP Command and Extension Registry](#).

SFTP Commands

The valid SFTP commands are (reference - [cURL Quote Explained](#)):

chgrp group file	The chgrp command sets the group ID of the file named by the file operand to the group ID specified by the group operand. The group operand is a decimal integer group ID.
chmod mode file	The chmod command modifies the file mode bits of the specified file. The mode operand is an octal integer mode number.
chown user file	The chown command sets the owner of the file named by the file operand to the user ID specified by the user operand. The user operand is a decimal integer user ID.
ln source_file target_file	The ln and symlink commands create a symbolic link at the target_file location pointing to the source_file location.
mkdir directory_name	The mkdir command creates the directory named by the directory_name operand.
pwd	The pwd command returns the absolute pathname of the current working directory.
rename source target	The rename command renames the file or directory named by the source operand to the destination path named by the target operand.
rm file	The rm command removes the file specified by the file operand.
rmdir directory	The rmdir command removes the directory entry specified by the directory operand, provided it is empty.
statvfs file	The statvfs command returns statistics on the file system in which specified file resides. (Added in 7.49.0)
symlink source_file target_file	See ln source_file target_file .

Example:

```
$ uftp -host ftp://SB-123-x64/ -file readme -user test -pwd test -f /home/build1/quote_CWD
```

Command file contents:

```
-quote "CWD //qa/durability_export/"
```

Note

[COMMAND](#) commands should always be used when available instead of any protocol-specific command.

"Prefix the command with an asterisk (*) to make libcurl continue even if the command fails as by default libcurl will stop at first failure." - [cURL Quote Explained](#)

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-quote <i>command</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

command is the name of the command to execute.

Valid values:

Command	Description	src	dst
GET	Retrieves a single file.	required	optional
PUT	Transfers a single file.	required	optional
MGET	Retrieves multiple files.	required	optional
MPUT	Transfers multiple files.	required	optional
DELETE	Deletes a single remote file.	required	
MDELETE	Deletes multiple remote files.	required	
MKDIR	Creates a directory on a remote computer.	required	
RMDIR	Removes a directory on a remote computer.	required	

Note



Do not use the [LIST](#) command with the QUOTE option; this will not work correctly with `libcurl`. Instead, use the [COMMAND](#) option.

REGEX - UFTP configuration option

Description

The REGEX option specifies the use of Regular Expression matching for the file name for the following commands:

- [LIST](#)
- [MGET](#)
- [MPUT](#)
- [MDELETE](#)

If the REGEX option is not specified, file name matching defaults to wildcard, if wildcard characters are present in the file pattern (a comma-delimited list of file names also is valid).

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-regex			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

There are no values to be specified for this option.

SITE - UFTP configuration option

Description

The SITE option specifies a command that is host-specific and is generally used to control file transfers.

SITE option commands are a requirement for z/OS FTP servers. They also are useful for IBM I and Windows servers to format directory listings ([LIST](#)) so that they can be easily parsed by the FTP Client.

For example:

```
-site "listfmt 1" -site "namefmt 1"
```

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-site <i>command</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

command is the host-specific command that is used to control file / data transfers.

SOURCE - UFTP configuration option

Description

The SOURCE option specifies the source directory / file / file pattern for the action defined by the command.

Source is required for all commands (except LIST, for which it is optional).

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-src <i>directory / file</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

directory / file is the source directory / file for the action defined by the command.

file can be either:

- File name
- File pattern using wildcards
- File pattern using regular expressions
- Comma-delimited list of file names

The following commands must specify a file name:

- GET
- PUT
- DELETE

The following commands can specify any of the four forms for *file*:

- MGET
- MPUT
- MDELETE


The following command can specify a directory or any of the four forms for *file*:

- LIST

Do not specify a *file* for the following commands:

- MKDIR
- RMDIR

Note

 For UFTP running under Windows, a SOURCE (or DESTINATION) option value cannot start with / (Windows XPS interprets this as the configuration option prefix: -), but it can start with \.

SSH_KEY - UFTP configuration option

Description

The SSH_KEY option specifies the path to the SSH key file.

SSH_KEY is optional for SFTP protocol.

If SSH_KEY is not specified, SFTP uses password authorization.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-key <i>path</i>			✓	✓	✓
Environment Variable	UFTPSSHKEY= <i>path</i>			✓	✓	
Configuration File Keyword	n/a					

Value

path is the path to the SSH key file.

SSL_CIPHER_LIST - UFTP configuration option

Description

The SSL_CIPHER_LIST option specifies one or more SSL/TLS cipher suites that UFTP would like to use to secure the control and data sessions for an FTPS transfer.

This option is used only when the [transfer protocol](#) is FTPS.


Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-ssl_cipher_list <i>list</i>			✓	✓	
Environment Variable	UFTPSSLCIPHERLIST= <i>list</i>			✓	✓	
Configuration File Keyword	ssl_cipher_list <i>list</i>			✓	✓	

Values

list is a comma-separated list of SSL/TLS cipher suites. The following table identifies the 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	<p>128-bit DES encryption with SHA-1 message digest.</p> <p>Note  As of Universal Agent 6.7.0.0, DES-CBC-SHA is supported only on HP-UX.</p> <p>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.</p>

SYSTEM_ID - UFTP configuration option

Description

The SYSTEM_ID option identifies the local Universal Broker with which UFTP must register before it 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	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-system_id <i>id</i>					✓
Environment Variable	UFTPSYSTEMID= <i>id</i>					✓
Configuration File Keyword	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.

UNHIDE - UFTP configuration option

Description

The UNHIDE option, by its inclusion in UFTP configuration, specifies that UFTP will include any hidden files in files reported to the user.

UNHIDE is used with the [LIST](#) command. By default, FTP Client does not report hidden files from the List. In UNIX and Windows, hidden files '.' prefix indicates that a file is hidden.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-unhide			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

(None.)

USER - UFTP configuration option

Description

The USER option specifies the remote user name to use in authentication.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-u / -user <i>name</i>					
Command Line, Long Form	-userid <i>name</i>			✓	✓	✓
Environment Variable	UFTPUSERID= <i>name</i>			✓	✓	✓
Configuration File Keyword	n/a					

Value

name the remote user name to use in authentication.

VERIFY_HOST_NAME - UFTP configuration option

Description

The VERIFY_HOST_NAME option specifies whether UFTP will require identifying information in an FTP server's certificate that corresponds to the specified HOST value.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-verify_host_name <i>opt</i>			✓	✓	✓
Environment Variable	UFTPVERIFYHOSTNAME= <i>opt</i>			✓	✓	✓
Configuration File Keyword	verify_host_name <i>opt</i>			✓	✓	✓

Value

opt specifies whether UFTP will ensure that the FTP server is actually the one intended. UFTP verifies this using the Common Name or Subject Alternate Name in the certificate that the FTP server presents during TLS/SSL handshake.

opt can be:

- YES - the Common Name or Subject Alternate Name must contain a host name or IP address that matches the one specified with UFTP's HOST option.
- NO - UFTP will proceed with the TLS/SSL handshake regardless of the host information contained in the FTP server's certificate.

Default is no.

This option is not used when PROTOCOL is SSH, which doesn't use TLS/SSL. It is used when the PROTOCOL is FTPS ("implicit" FTPS) or when UFTP requests explicit FTPS support.

Notes for Explicit FTPS (FTPES) Support

Explicit FTPS support was added to UFTP for Universal Agent 7.1.0.0. When an FTP server enables *explicit* TLS/SSL support, it allows clients to request encrypted sessions over the standard FTP port 21. This is different from *implicit* TLS/SSL support, which requires a connection to a unique, well-known port (990 by default) that ONLY accepts TLS/SSL-enabled client requests.

UFTP has always offered implicit FTPS support via the FTPS PROTOCOL value.

Because an FTP client may not always know whether the FTP server's port 21 is accepting SSL-enabled connections (i.e., it may only accept unencrypted, plain-text FTP sessions), explicit FTPS is not really a distinct protocol per se. Therefore, UFTP offers explicit FTPS support by applying new options and supported values to the existing FTP PROTOCOL value.

UFTP requests an explicit FTPS session using the ENABLE_SSL option or by prefixing the HOST option's value with `ftpes://`.

The `-verify_host_name` option can be specified for explicit or implicit FTPS sessions.

VERSION - UFTP configuration option

Description

The VERSION option writes the program version information and copyright.

Usage

Specification Method	Parameter / Value	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-v			✓	✓	✓
Command Line, Long Form	-version			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

Value

There are no values to be specified for this option.

Universal Message Translator

Overview

Universal Message Translator (UMET) translates error messages into return (exit) codes based on a user-defined translation table.

Every command ends with a return code that indicates the success or failure of the command execution. Typically, a return code of 0 indicates success; all other codes indicate failure.

However, a small number of commands do not set their return code under failure conditions; instead, they issue error messages. Based on the user-defined translation table, Universal Message Translator translates these error messages into return codes.

Detailed Information

The following pages provide detailed information for Universal Message Translator:

- [Universal Message Translator - Usage](#)
- [Universal Message Translator for z/OS](#)
- [Universal Message Translator for Windows and UNIX](#)
- [Universal Message Translator for IBM i](#)
- [Universal Message Translator Configuration Options](#)

Message Translator Examples

See [Message Translation - Examples](#) for examples of how to use Universal Message Translator.

Universal Message Translator - Usage

- [Input](#)
- [Translation Steps](#)
- [Translation Table](#)
 - [Translation Table Format](#)
 - [Translation Table Fields](#)
- [Matching Algorithm](#)

Input

UMET requires two input files:

1. Message Input file (user-specified or standard input) containing the error messages that are to be translated into a return codes.
2. Translation Table file containing the user-defined translation table that controls the error message-to-return code translation process.

Translation Steps

To perform a translation, UMET:

1. Reads the messages in the input file.
2. Matches each line against the translation table entries.
3. Exits with an return code from the best match in the translation table.

If no match is found, UMET ends with return code 0.

UMET performs operations specified by the configuration options.

Translation Table

The translation table specifies:

- Text to search for.
- Return code associated with the text.
- Precedence when multiple matches are found.

Translation Table Format

The translation table consists of one or more lines.

Each line is either:

- Comment line (# in column one)
- Blank line (ignored)
- Translation table entry

Translation table entries consist of two fields separated by spaces or tabs. An entry cannot be continued onto multiple lines.

Translation Table Fields

Field	Description
Message Mask	<p>Selects which messages to match in the input file. The mask must be enclosed in double (") quotation marks.</p> <p>Mask characters include the asterisks (*) and the question mark (?). The asterisk matches 0 or more characters and the question mark matches one character.</p> <p>If an asterisk, question mark, or quotation mark is required in the message text, it must be preceded with a back slash (\). If a back slash is required in the message text, it must be preceded by another back slash.</p>
Exit Code	<p>Specifies an integer value that UMET exits with if this entry is the resulting match.</p> <p>The exit code is in the range of -99999 to 99999.</p>

Matching Algorithm

The input file is read line by line. For each line, the line is compared to each entry in the translation table. All the matching entries are saved.

After the entire input file is read, the matched entries from the translation table are sorted in ascending order by their line number in the translation table. The first entry in this sorted list is the resulting translation table entry. The exit code from the resulting translation table entry is used as the return code of UMET. If no matching entry is found, UMET exits with 0.

Universal Message Translator for zOS

- [JCL](#)
- [DD Statements used in JCL](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

JCL

The following figure illustrates the Universal Message Translator for z/OS JCL.

```
//UMET      EXEC PGM=UMET,PARM='-TABLE TABLE'
//STEPLIB  DD  DISP=SHR,DSN=UNV.SUNVLOAD
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
//TABLE    DD  DISP=SHR,MY.TRANS.TABLE
//SYSIN    DD  DISP=SHR,MY.MSG.FILE
```

UMET options are passed in with the PARM keyword on the EXEC statement.

DD Statements used in JCL

The following table describes the DD statements used in the Universal Message Translator for z/OS [#JCL](#), above.

ddname	Description
STEPLIB	Load library in which program UMET is located.
SYSPRINT	UMET standard output ddname.
SYSOUT	UMET standard error ddname.
TABLE	Translation table specified by the -table option on the PARM keyword.
SYSIN	Standard input ddname from which the message file is read.

Configuration Options

The following table identifies the UMET for z/OS configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
HELP	Writes a description of the configuration options and their format.
IGNORE_CASE	Specification that matching of message masks to the input file is not case sensitive.
MESSAGE_FILE	Input message file name.
MESSAGE_LEVEL	Level of messages that will be written.
TRANSLATION_TABLE	Translation table file name.
VERSION	Writes the program version and copyright information.

Command Line Syntax

The following figure illustrates the syntax - using the long form of command line options - of UMET for z/OS.

```
umet
-tabletable
[-filemessages]
[-ignorecase]
[-level {verbose|info|warn|error}]

umet
{ -help | -version }
```

Universal Message Translator for Windows and UNIX

Configuration Options

The following table identifies the UMET for Windows and UNIX configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
HELP	Writes a description of the configuration options and their format.
IGNORE_CASE	Specification that matching of message masks to the input file is not case sensitive.
MESSAGE_FILE	Input message file name.
MESSAGE_LEVEL	Level of messages that will be written.
TRANSLATION_TABLE	Translation table file name.
VERSION	Writes the program version and copyright information.

Command Line Syntax

The following figure illustrates the syntax — using the long form of command line options — of UMET for Windows and UNIX.

```
umet
-tabletable
[-filemessages]
[-ignorecase]
[-level {verbose|info|warn|error}]

umet
{ -help | -version }
```

Universal Message Translator for IBM i

- [Return Codes](#)
- [Workload Automation 5 for IBM i Commands](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

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

Return Codes

The resulting return code from the translation process is converted into an IBM i escape message.

The escape message ID and message severity depend on the return code value, as identified in the following table.

Return Code	Message ID	Message Severity
1 - 10	UNV0344	10
11 - 20	UNV0345	20
21 - 30	UNV0346	30
31 and higher	UNV0347	40

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 [Change Release Tag](#) program, **UCHGRLS**, lets you change the tagged command names in **QSYS** to the untagged command names in **UNVPRD511**.

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

Configuration Options

The following table identifies the UMET for IBM i configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
IGNORE_CASE	Specification that matching of message masks to the input file is not case sensitive.
MESSAGE_FILE	Input message file name.
MESSAGE_LEVEL	Level of messages that will be written.
TRANSLATION_TABLE	Translation table file name.

Command Line Syntax

The following figure illustrates the syntax — using the STRUME parameter form of command line options — of UMET for IBM i.

```
STRUME
TBL([library/]{umetbl|filename}) [TBLMBR(member)]
[MSGFILE([library/]{*stdin|filename}) [MSGMBR(member)] ]

**Additional Options**
[MSGLEVEL(*{verbose|info|warn|error})]
[IGNORECASE({yes|no})]
```

Universal Message Translator Configuration Options

Universal Message Translator Configuration Options

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

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

Configuration Options List

The following table identifies the Universal Message Translator configuration options.

Option Name	Description
HELP	Displays a description of the command line options and their format.
IGNORE_CASE	Specification that matching of message masks to the input file is not case sensitive.
MESSAGE_FILE	Input message file name.
MESSAGE_LEVEL	Level of messages that will be displayed.
TRANSLATION_TABLE	Translation table file name.
VERSION	Writes the program version and copyright information.

HELP - UMET configuration option

Description

The HELP option display a description of the Universal Message Translator 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		✓	✓	✓
STRUME Parameter	n/a				

Values

(There are no values for this option.)

IGNORE_CASE - UMET configuration option

Description

The IGNORE_CASE option specifies that the matching of message masks to the input file is not case sensitive.

(For example, if IGNORE_CASE is used, the word **Error** matches **ERROR**).

If this option is not used, the matching of message masks is case sensitive.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-i	✓	✓	✓	✓
Command Line, Long Form	-ignorecase	✓	✓	✓	✓
STRUME Parameter	IGNORECASE(* <i>option</i>)	✓			

Values

There are no values for this option.

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.

MESSAGE_FILE - UMET configuration option


Description

The MESSAGE_FILE option specifies the name of the input message file.

Note

 If this option is not used, Universal Message Translator reads its input from standard input.

z/OS

 If this option is not used, Universal Message Translator reads the input file from SYSIN ddname.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-f <i>messages</i>	✓	✓	✓	✓
Command Line, Long Form	-file <i>messages</i>	✓	✓	✓	✓
STRUME Parameter	MSGFILE(<i>messages</i> [<i>library</i>]) [MSGMBR(<i>member</i>)]	✓			

Values

messages is the name of the input message file.

IBM i	<p>Valid values for <i>messages</i> are:</p> <ul style="list-style-type: none"> • *stdin Standard input is used to specify the messages. 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. This value is not valid for the command line form. Simply do not specify the command line option to read from standard input. • filename Name of 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. A member name can be used for further qualification by specifying the MSGMBR parameter.
z/OS	<i>messages</i> is the ddname to which the input message file is allocated.

MESSAGE_LEVEL - UMET configuration option

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-l <i>level</i>	✓	✓	✓	✓
Command Line, Long Form	-level <i>level</i>	✓	✓	✓	✓
STRUME Parameter	MSGLEVEL (* <i>level</i>)	✓	✓	✓	✓

Values

level indicates either of the following level of messages:

- **verbose**
Writes messages that provide information on the message matching process. The messages are helpful for verifying or debugging a translation table.
(Information, warning, and error messages also are written.)
- **info**
Writes informational, warning, and error messages.
- **warn**
Writes warning and error messages.
- **error**
Writes error messages only.

Default

IBM i	info
UNIX	warn
Windows	warn
z/OS	info

TRANSLATION_TABLE - UMET configuration option

Description

The TRANSLATION_TABLE option specifies the name of the translation table file.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-t <i>table</i>	✓	✓	✓	✓
Command Line, Long Form	-table <i>table</i>	✓	✓	✓	✓
STRUME Parameter	TBL(<i>table</i> [<i>library</i>]) [TBLMBR(<i>member</i>)]	✓			

Values

table is the name of the translation table file.

IBM i	<p>Valid values for <i>table</i> are:</p> <ul style="list-style-type: none"> • umetbl Default file name for the STRUME parameter. The command line form does not default. An error is generated if no value is specified. • <i>filename</i> Translation table is read from a file. The file name can be qualified by a library name. Otherwise, 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 TBLMBR parameter.
z/OS	<i>table</i> is the ddname to which the translation table is allocated.

VERSION - UMET configuration option

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-v		✓	✓	✓
Command Line, Long Form	-version		✓	✓	✓
STRUME Parameter	n/a				

Values

(There are no values for this option.)

Universal Products Install Merge

- [Universal Products Install Merge](#)
- [Detailed Information](#)
- [Universal Products Install Merge Examples](#)

Universal Products Install Merge

The Universal Products Install Merge (UPIMERGE) utility merges options and values from one Universal Agent component configuration file or component definition file with another.

UPIMERGE runs automatically during Universal Agent installation upgrades on UNIX and Windows. During the install, UPIMERGE combines options and values from existing configuration and component definition files with the options and values in the most recent versions of those files (delivered with the distribution package).

The result of each merge is a single file, with preserved options and values residing alongside any new options and values that were introduced to support new Universal Agent features.

The Universal Agent (UNIX and Windows) and Universal Enterprise Controller (Windows only) distribution packages also install UPIMERGE. This makes UPIMERGE available at any time for recovering archived options and values and merging them with the most recent options and values.

When used to update a Universal Agent configuration or component definition file, UPIMERGE must run with a user account that has write access to the output file. This typically means administrative access (that is, root on UNIX, Administrator on Windows).

Detailed Information

The following pages provide detailed information for Universal Products Install Merge:

- [Universal Products Install Merge - Usage](#)
- [Universal Products Install Merge Configuration Options](#)

Universal Products Install Merge Examples

See [Merging Configuration Options](#) for examples of how to use Universal Products Install Merge.

Universal Products Install Merge - Usage

- [Usage](#)
- [Command Line Syntax](#)
- [Examples](#)

Usage

As input, UPIMERGE typically uses an archived configuration file or component definition file. However, it can use any file as input, provided that the file is in standard keyword / value format.


UPIMERGE output is a file containing the options and values from the input file, merged with those in the output file. For each option in the output file, UPIMERGE replaces its value with the value of a matching option in the input file. If the input file contains options not defined in the output file, UPIMERGE adds those options to the end of the output file.

UPIMERGE does not attempt to sequence the entries that it adds to the output file. Thus, the order of options in the output file may not match the order of the same options in the input file.

If the input file contains more than one entry for an option, UPIMERGE adds every entry to the output file. The application will use the value of the last entry that appears in the output file.

UPIMERGE does not update any comments in the output file. Options that reside only in the output file are either commented out or left as is, depending on the command line parameters specified.

Note

 UPIMERGE has no effect on a Universal Agent application's behavior if the local Universal Broker is in managed mode. In that environment, configurations and component definitions reside in a database file, not a text file. Use the [I-Management Console](#) application to manage configurations for managed installations.

Command Line Syntax

The following figure illustrates the syntax - using the long form of command line options - of Universal Products Install Merge.

```
upimerge
  -dest filename
  [-source filename ]
  [ -installdir dirname { -cfgtype type [-comptype type] | -comptype type [-cfgtype type] } ]
  [ -keep_nomatch {yes|no} ]
  [ -bkup_dest {yes|no} ]
  [ -file filename | -encryptedfile filename [-key key] [-keypath path] ]
  [ -level { trace|audit|info|warn|error }[, { time|notime } ] ]

upimerge
{ -help | -version }
```

Examples

For examples of how to run Universal Products Install Merge, see [Merging Configuration Options Examples](#).

Universal Products Install Merge Configuration Options

Overview

This page provides links to detailed information on the configuration options available for use with Universal Products Install Merge (UPIMERGE).

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

Configuration Options List

The following table identifies the Universal Products Install Merge configuration options.

Option Name	Description
BACKUP_DESTINATION	Creates a copy of the original DESTINATION_FILE prior to the merge.
COMMAND_FILE_ENCRYPTED	Name of a file encrypted with Universal Encrypt that contains command options.
COMMAND_FILE_PLAIN	Name of a plain text file that contains command options.
COMPONENT_TYPE	<p>Notifies UPIMERGE that the SOURCE_FILE is a component definition file that contains settings for the specified Universal Agent server component.</p> <p>You cannot use this option with CONFIGURATION_TYPE.</p> <p>UPIMERGE ignores this option if INSTALLATION_DIRECTORY is omitted.</p>
CONFIGURATION_TYPE	<p>Notifies UPIMERGE that the SOURCE_FILE is a configuration file that contains settings for the specified Universal Agent application.</p> <p>You cannot use this option with COMPONENT_TYPE.</p> <p>UPIMERGE ignores this option if INSTALLATION_DIRECTORY is omitted.</p>
DESTINATION_FILE	Name of a file used to store the result of the merge.
ENCRYPTION_KEY	Key used to encrypt the file specified by COMMAND_FILE_ENCRYPTED .
HELP	Displays a description of the command line options and their format.
INSTALLATION_DIRECTORY	Primary location in which the Universal Agent server component identified with COMPONENT_TYPE or the Universal Agent application identified by CONFIGURATION_TYPE resides.
KEEP_NOMATCH	Controls merge behavior when an option in DESTINATION_FILE has no match in SOURCE_FILE .
KEYSTORE_PATH	Path to a local or remote Universal Broker service interface from which an encryption key can be obtained.
MESSAGE_LEVEL	Level of messages to write.
SOURCE_FILE	<p>Name of a file used as input to the merge.</p> <p>If this parameter is omitted, UPIMERGE assumes input is redirected via stdin.</p>
VERSION	Writes the program version and copyright information.

BACKUP_DESTINATION - UPIMERGE configuration option

Description

The BACKUP_DESTINATION option instructs UPIMERGE to create a copy of the original configuration or component definition file specified by [DESTINATION_FILE](#) before merging the contents of [SOURCE_FILE](#) into it.

UPIMERGE creates the backup file in the same directory as the original configuration or component definition file.

UNIX	<p>The format of the file name UPIMERGE generates is implementation-dependent.</p> <p>On most systems, the format is xxxxxxxx, where:</p> <ul style="list-style-type: none"> xxx is a prefix based on the name of the Universal Agent configuration or component definition file specified for DESTINATION_FILE. (UPIMERGE uses a default of uim if it does not recognize the file name.) nnnnn is a unique alpha-numeric identifier. <p>On some older UNIX systems, the format is undefined, but the name is still unique.</p>
Windows	<p>UPIMERGE generates a file name with a format of xxxxnnn.tmp, where:</p> <ul style="list-style-type: none"> xxx is a prefix based on the name of the Universal Agent configuration or component definition file specified for DESTINATION_FILE. (UPIMERGE uses a default of uim if it does not recognize the file name.) nnnn is a unique alpha-numeric identifier.

UPIMERGE reports the backup file it generates with message ID UNV44121.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-bkup_dest <i>option</i>			✔	✔	
Environment Variable	UIMBKUPDEST= <i>option</i>			✔	✔	

Values

option specifies whether UPIMERGE creates a copy of the original configuration or component definition file.

Valid values for *option* are:

- yes**
Create a backup of the file specified with [DESTINATION_FILE](#) before merging the contents of [SOURCE_FILE](#) into it.
- no**
Merge the contents of [SOURCE_FILE](#) into [DESTINATION_FILE](#), without saving a copy of the original [DESTINATION_FILE](#).

Default is no.

COMMAND_FILE_ENCRYPTED - UPIMERGE configuration option

Description

The `COMMAND_FILE_ENCRYPTED` option specifies the file containing encrypted values for command line option parameters.

Command files specify an additional source of command line options. Storing options in a file is useful in situations where it is not desirable to specify them on the command line. The application reads the file and processes the options exactly like those specified on the command line. The options must be in their respective command line formats.

UPIMERGE can process encrypted or plain text command files (see the [COMMAND_FILE_PLAIN](#) option). Encrypted command files are an excellent place to store sensitive data that you may want to hide from the command line. As an added measure of security, Stonebranch, Inc. recommends that you configure the file system's security to protect any command file that contains sensitive data from unauthorized read access.

Use the [Universal Encrypt](#) utility provided with Universal Agent to encrypt a plain text command file. If Universal Encrypt used a key to encrypt the file, use that same key to read the file using the [ENCRYPTION_KEY](#) option.

Note



If UPIMERGE encounters the `COMMAND_FILE_ENCRYPTED` and [COMMAND_FILE_PLAIN](#) options on its command line, it uses the file specified for [COMMAND_FILE_PLAIN](#).

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-x filename</code>			✓	✓	
Command Line, Long Form	<code>-encryptedfile filename</code>			✓	✓	
Environment Variable	n/a					

Values

filename is the name of the encrypted file that contains the command line options and their values.

COMMAND_FILE_PLAIN - UPIMERGE configuration option


Description

The `COMMAND_FILE_PLAIN` option specifies the file containing plain text values for command line option parameters.

Command files specify an additional source of command line options. Storing options in a file is useful in situations where it is not desirable to specify them on the command line. The application reads the file and processes the options exactly like those specified on the command line. The options must be in their respective command line formats.

UPIMERGE can process encrypted or plain text command files (see the [COMMAND_FILE_ENCRYPTED](#) option). Stonebranch, Inc. recommends that you configure the file system's security to protect command files from unauthorized access.

Note

 If UPIMERGE encounters the [COMMAND_FILE_ENCRYPTED](#) and `COMMAND_FILE_PLAIN` options on its command line, it uses the file specified for `COMMAND_FILE_PLAIN`.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-f filename</code>			✓	✓	
Command Line, Long Form	<code>-file filename</code>			✓	✓	
Environment Variable	n/a					

Values

filename is the name of the file that contains the command line options and their values.

COMPONENT_TYPE - UPIMERGE configuration option


Description

The COMPONENT_TYPE option notifies UPIMERGE that SOURCE_FILE is a component definition file that contains settings for the specified Universal server component.

UPIMERGE uses this value together with the INSTALLATION_DIRECTORY value to set options in a component definition file that depend on the Universal Agent server component's installed location.

Although COMPONENT_TYPE is not required to merge component definition files, using it ensures that UPIMERGE executes any custom merge logic necessary for a particular component's definition file.

Note

 You cannot use COMPONENT_TYPE with the CONFIGURATION_TYPE option. UPIMERGE ignores COMPONENT_TYPE if the INSTALLATION_DIRECTORY is omitted.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-comptype <i>type</i>			✓	✓	
Environment Variable	UIMCOMPTYPE= <i>type</i>			✓	✓	

Values

type identifies a Universal Agent server component.

Valid values for *type*, and the Universal Agent server component that each value represents, are:

oms	Universal Message Service Server
uac	Universal Application Container Server
uag	Universal Automation Center Agent
ucmd	Universal Command Server
uctl	Universal Control Server
udm	Universal Data Mover Server
uem d	Universal Event Monitor Server (demand-driven)
uems	Universal Event Monitor Server (event-driven)

There is no default.

CONFIGURATION_TYPE - UPIMERGE configuration option

Description

The CONFIGURATION_TYPE option notifies UPIMERGE that the file specified with the SOURCE_FILE option is a configuration file that contains settings for the specified Universal Agent application.

UPIMERGE uses this value together with the INSTALLATION_DIRECTORY value to set options in a configuration file that depend on the Universal Agent application's installed location.

Although CONFIGURATION_TYPE is not required to merge configuration files, using it ensures that UPIMERGE executes any custom merge logic necessary for a particular application's configuration file.

Note



You cannot use CONFIGURATION_TYPE with the COMPONENT_TYPE option. UPIMERGE ignores CONFIGURATION_TYPE if the INSTALLATION_DIRECTORY option is omitted.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-cfgtype <i>type</i>			✓	✓	
Environment Variable	UIMCFGTYPE= <i>type</i>			✓	✓	

Values

type specifies the Universal Agent configuration file to merge.

Valid values for *type*, and the Universal Agent component that each value represents, are shown in the following table:

omss	Universal Message Service Server
uac	Universal Access Control List
uacs	Universal Application Container Server
uags	Universal Automation Center Agent Server
ubroker	Universal Broker
ucmd	Universal Command Manager
ucmds	Universal Command Server
uctl	Universal Control Manager
uctls	Universal Control Server
udm	Universal Data Mover Manager
udms	Universal Data Mover Server
uec	Universal Enterprise Controller
ueld	Universal Event Log Dump Utility
uem	Universal Event Monitor Manager
uems	Universal Event Monitor Server
upps	Universal Connector for PeopleSoft
uquery	Universal Query
usap	Universal Connector for SAP

DESTINATION_FILE - UPIMERGE configuration option

Description

The `DESTINATION_FILE` option identifies the configuration or component definition file that UPIMERGE uses to store the results of the merge with `SOURCE_FILE`.

During the merge process, UPIMERGE replaces all values in `DESTINATION_FILE` that have a match in `SOURCE_FILE`. If `SOURCE_FILE` contains configuration or component definition options not defined in `DESTINATION_FILE`, UPIMERGE adds those options to the end of the output file.

UPIMERGE does not attempt to sequence the entries it adds to `DESTINATION_FILE`, which means the order of options in the output file may not match the order of the same options specified in `SOURCE_FILE`. In addition, UPIMERGE does not update any comments in the output file. Finally, if the input file contains more than one entry for a given option, UPIMERGE adds every occurrence of that entry to `DESTINATION_FILE`. The application uses the value of the last entry that appears in the output file.

Options that reside only in `DESTINATION_FILE` are either commented out or left as-is, depending on the value of the `KEEP_NOMATCH` option.

The file name specified for `DESTINATION_FILE` must exist (even if it is empty) before executing UPIMERGE.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	<code>-dest filename</code>			✓	✓	
Environment Variable	<code>UIMDESTFILE=filename</code>			✓	✓	

Values

filename is the name of the file that contains the results of the merge with `SOURCE_FILE`.

filename can contain a complete path to the output file or a path relative to the current directory.

ENCRYPTION_KEY - UPIMERGE configuration option

Description

The ENCRYPTION_KEY option specifies the key used to encrypt the file specified with the [COMMAND_FILE_ENCRYPTED](#) option.

This key acts much like a password, preventing unauthorized users from decrypting the encrypted command file.

If [Universal Encrypt](#) used a key to encrypt the file, UPIMERGE requires the same key to decrypt the file.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-K <i>key</i>			✓	✓	
Command Line, Long Form	-key <i>key</i>			✓	✓	
Environment Variable	UIMKEY= <i>key</i>			✓	✓	

Values

key is the key used to encrypt the command file.

HELP - UPIMERGE configuration option

Description

The HELP option displays a description of the Universal Products Install Merge command line options and their format.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-h			✓	✓	
Command Line, Long Form	-help			✓	✓	
Environment Variable	n/a					

Values

(There are no values used with this option.)

INSTALLATION_DIRECTORY - UPIMERGE configuration option

Description

The `INSTALLATION_DIRECTORY` option specifies the installed location of the Universal Agent application or server component specified with the `CONFIGURATION_TYPE` or `COMPONENT_TYPE` option, respectively.

`INSTALLATION_DIRECTORY` is a command line-only parameter that UPIMERGE uses to set values for options in `DESTINATION_FILE` that depend upon the installed location of a specified Universal Agent application or server component.

It does NOT specify the installed location of the UPIMERGE utility.

Note



`INSTALLATION_DIRECTORY` requires the `CONFIGURATION_TYPE` or the `COMPONENT_TYPE` option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-installdir <i>dirname</i>			✓	✓	
Environment Variable	UIMSTALLDIR= <i>dirname</i>			✓	✓	

Values

dirname is the primary directory into which the Universal Agent application or server component specified by `CONFIGURATION_TYPE` or `COMPONENT_TYPE` resides.

KEEP_NOMATCH - UPIMERGE configuration option

Description

The KEEP_NOMATCH option instructs UPIMERGE to comment out any option in [DESTINATION_FILE](#) that does not exist in [SOURCE_FILE](#).

The primary purpose of KEEP_NOMATCH is to prevent the introduction of new values for existing configuration options during a product upgrade.

For example, the typical sequence of events for an upgrade is as follows:

1. The install program archives a product's current configuration file to a well-known location.
2. The install program places a new configuration file from the distribution package into the well-known Universal Agent configuration file directory.
3. The install program passes the name of the archive file to UPIMERGE as [SOURCE_FILE](#).
4. The install program passes the name of the newly-installed configuration file to UPIMERGE as [DESTINATION_FILE](#).

If the archive file contains no value for a given configuration option - [MESSAGE_LEVEL](#), for example - that means the corresponding program currently runs with the application-defined default. If the newly-installed configuration file were to contain an entry that sets [MESSAGE_LEVEL](#) to **audit**, and UPIMERGE kept this option by default, the result would be a change to the application's behavior (assuming the application-defined default for the option is something other than **audit**).

Note



As an additional precaution against a product upgrade changing an application's behavior, packaged configuration files have most options commented out. This approach allows product upgrades to "announce" the availability of new options while preserving an existing configuration.

On the other hand, a situation may arise after the install where it is desirable to introduce new options and/or values into a product's configuration. In this case, simply execute UPIMERGE with KEEP_NOMATCH set to **yes**.

While the discussion above focused on configuration files, UPIMERGE uses the same approach with component definition files.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-keep_nomatch <i>option</i>			✔	✔	
Environment Variable	UIMKEEPNOMATCH= <i>option</i>			✔	✔	

Values

option specifies whether UPIMERGE should "keep" or comment out any setting contained in [DESTINATION_FILE](#) that has no match in [SOURCE_FILE](#).

Valid values for *option* are:

- **yes**
Keep any option in [DESTINATION_FILE](#) without a match in [SOURCE_FILE](#).
- **no**
Comment out any option in [DESTINATION_FILE](#) with no match in [SOURCE_FILE](#).

Default is no.

KEYSTORE_PATH - UPIMERGE configuration option

Description

The KEYSTORE_PATH option specifies the local or remote Universal Broker service interface from which an encryption key can be obtained.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-p <i>path</i>			✓	✓	✓
Command Line, Long Form	-keypath <i>path</i>			✓	✓	✓
Environment Variable	n/a					

Value

path is the path to the local or remote Universal Broker service interface.

MESSAGE_LEVEL - UPIMERGE configuration option

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage


Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l <i>level</i>			✓	✓	
Command Line, Long Form	-level <i>level</i>			✓	✓	
Environment Variable	UIMLEVEL= <i>level</i>			✓	✓	

Values

level indicates either of the following level of messages:

- **trace**
Writes trace messages used for diagnostic purposes.

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.

SOURCE_FILE - UPIMERGE configuration option

Description

The SOURCE_FILE option identifies the file that UPIMERGE merges into the configuration or component definition file specified by the DESTINATION_FILE option.

During the merge process, UPIMERGE replaces all values in DESTINATION_FILE that have a match in SOURCE_FILE.

If SOURCE_FILE contains configuration or component definition options not defined in DESTINATION_FILE, UPIMERGE adds those options to the end of the output file. If SOURCE_FILE contains more than one entry for a given option, UPIMERGE adds every occurrence of that entry to DESTINATION_FILE. The application uses the value of the last entry that appears in the output file.

Use of SOURCE_FILE is optional; it also is possible to provide input options and values to UPIMERGE using standard input (that is, stdin) redirection.

The following illustrates an example of stdin redirection.

```
upimerge -dest outfile.txt
```

Note



It is possible to execute UPIMERGE without the redirected input file. However, the program may appear unresponsive. In this situation, UPIMERGE is actually waiting for an end-of-file indicator to signal the end of the redirected input.

Windows	Supply the end-of-file indicator by pressing <Ctrl+Z> <Enter>.
UNIX	Supply the end-of-file indicator by pressing <Ctrl+D>.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-source <i>filename</i>			✓	✓	
Environment Variable	UIMSOURCE= <i>filename</i>			✓	✓	

Values

filename is the name of the file that contains options and values that UPIMERGE merges into DESTINATION_FILE.

filename can contain a complete path to the input file or a path relative to the current directory.

VERSION - UPIMERGE configuration option

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-v			✓	✓	
Command Line, Long Form	-version			✓	✓	
Environment Variable	n/a					

Values

(There are no values for this option.)

Universal Query

- [Overview](#)
- [Usage](#)
- [Network Alias Look-Up](#)
- [Examples](#)
- [Detailed Information](#)
- [Universal Query Examples](#)

Overview

The Universal Query (UQUERY) utility queries any Universal Broker for Broker-related and active component-related information.

Usage

UQUERY returns information for a Universal Broker that is installed on the host, as specified by configuration options on the command line or in a configuration file. Information regarding the components managed by a particular Broker also can be requested.

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

Network Alias Look-Up

When a Linux/Unix or Windows task that uses the [Agent Cluster](#) Distribution type [Network Alias](#) is launched, the task first will confirm if a non-expired cached Universal Agent name is available, and if so, this name will be used.

If the name is not available, a UQUERY look-up will be performed and the returned IP address will be used to look up the required Universal Agent name to be used for the task. The name will be cached based upon the Agent Cluster cache retention definition. If no active or registered Agent is found, the task will enter a status of Undeliverable and the UQUERY look-up will be retried after a period (in minutes) specified by the [Agent Cluster Network Alias Retry Interval](#).

Examples

For an example of Universal Query output, and examples of how to run Universal Query, see [Examples of Querying for Job Status and Activity](#).

Detailed Information

The following pages provide detailed information for Universal Query:

- [Universal Query for z/OS](#)
- [Universal Query for Windows and UNIX](#)
- [Universal Query for IBM i](#)
- [Universal Query Configuration Options](#)

Universal Query Examples

See [Monitoring and Alerting](#) for examples of how to use Universal Query.

Universal Query for zOS

- [JCL Procedure](#)
- [DD Statements used in JCL Procedure](#)
- [JCL](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

JCL Procedure

The following figure illustrates the Universal Query for z/OS JCL procedure (**UQRYPRC**, located in the **SUNVSAMP** library) that is provided to simplify the execution JCL and future maintenance.

```
//UQRYPRC  PROC UPARAM=,           -- UQUERY options
//          UCMDPRE=#SHLQ.UNV
// *
//PS1      EXEC PGM=UQUERY , PARM= ' ENVAR(TZ=EST5EDT) /&UPARM '
//STEPLIB  DD  DISP=SHR,DSN=&UCMDPRE..SUNVLOAD
// *
//UNVNLS   DD  DISP=SHR,DSN=&UCMDPRE..SUNVNLS
//UNVTRACE DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
```

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

(See the section on [z/OS Installation](#) in the [Universal Agent 7.1.x Installation, Upgrade, and Applying Maintenance](#) for information regarding the customization of Language Environment parameters.)

DD Statements used in JCL Procedure

The following table describes the DD statements used in the Universal Query for z/OS JCL procedure illustrated above.

ddname	Description
STEPLIB	Load library in which program UQUERY is located.
UNVNLS	UQUERY national language support ddname.
UNVTRACE	UQUERY trace ddname.
SYSPRINT	UQUERY standard output ddname.
SYSOUT	UQUERY standard error ddname.

JCL

The following figure illustrates the Universal Query for z/OS JCL using the **UQRYPRC** JCL procedure, above.

```
//jobname  JOB CLASS=A,MSGCLASS=X
//STEP1    EXEC UQRYPRC
//SYSIN    DD  *
-i dallas
/*
```

Job step STEP1 executes the procedure **UQRYPRC**.

The command options are specified on the **SYSIN DD**.

Configuration Options

UQUERY for z/OS operations are controlled by the configuration options, which are specified either on the command line (via the PARM keyword of the EXEC statement or in the SYSIN ddname) or the configuration file.

The following table identifies the UQUERY configuration options for z/OS. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
CODE_PAGE	Character code page used to translate text data received and transmitted over the network.
COMMAND_ID	Requests that Universal Query return information for all records that match the specified command ID.
COMPONENT_ID	Requests that Universal Query return information only for the specified component ID.
HELP	Writes a description of the configuration options and their format.
MANAGERS	Specification for whether or not Universal Query requests manager component information from the queried Broker.
MESSAGE_LANGUAGE	Universal Message Catalog (UMC) file used to write messages.
MESSAGE_LEVEL	Level of messages to write.
OUTBOUND_IP	Sets the host or IP address that UQUERY binds to when initiating outgoing connections.
PING	Information Universal Query requests from Universal Broker.
REMOTE_HOST	IP address of the remote computer.
REMOTE_PORT	TCP port number on the remote computer on which Universal Broker is accepting connections.
REPORT	Format in which Universal Broker information is written.
SYSTEM_ID	Local Universal Broker with which the Universal Query must register.
VERSION	Writes the program version and copyright information.

Command Line Syntax

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

```

uquery
-host hostaddress
[-port port]
[-system_id ID]
[-ping {yes|no}]
[-report {normal|fixed}]
[-codepage codepage]
[-level {trace|audit|info|warn|error}]
[-lang language]
[-managers {yes|no}]
[-outboundip host]
[-cmdid ID]
[-component ID]

uquery
{ -help | -version }

```

Universal Query for Windows and UNIX

- [Configuration Options](#)
- [Command Line Syntax](#)

Configuration Options

The following table identifies the UQUERY configuration options for Windows and UNIX. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
BIF_DIRECTORY *	Broker Interface File (BIF) directory where the Universal Broker interface file is located.
CODE_PAGE	Character code page used to translate text data received and transmitted over the network.
COMMAND_ID	Requests that Universal Query return information for all records that match the specified command ID.
COMPONENT_ID	Requests that Universal Query return information only for the specified component ID.
HELP	Writes a description of the configuration options and their format.
MANAGERS	Specification for whether or not Universal Query requests manager component information from the queried Broker.
MESSAGE_LANGUAGE	Universal Message Catalog (UMC) file used to write messages.
MESSAGE_LEVEL	Level of messages to write.
NLS_DIRECTORY	Directory where the Universal Query message catalog and code page tables are located.
OUTBOUND_IP	Sets the host or IP address that UQUERY binds to when initiating outgoing connections.
PING	Information Universal Query requests from Universal Broker.
PLF_DIRECTORY *	Program Lock File (PLF) directory where the program lock files are located.
REMOTE_HOST	IP address of the remote computer.
REMOTE_PORT	TCP port number on the remote computer on which Universal Broker is accepting connections.
REPORT	Format in which Universal Broker information is written.
VERSION	Writes the program version and copyright information.
* Valid for UNIX only.	

Command Line Syntax

The following figure illustrates the command line syntax — using the command line, long form of the configuration options — of UQUERY for Windows and UNIX.

```
uquery
-host hostaddress
[-port port]
[-ping {yes|no}]
[-report {normal|fixed}]
[-bif_directory directory] (NOTE: This option is valid only for UNIX.)
[-plf_directory directory] (NOTE: This option is valid only for UNIX.)
[-codepage codepage]
[-level {trace|audit|info|warn|error}]
[-lang language]
[-managers {yes|no}]
[-outboundip host]
[-cmdid ID]
[-component ID]

uquery
{ -help | -version }
```

Windows



The Universal Configuration Manager also can be used to obtain the same information reported by the command line version of Universal Query.

Universal Query for IBM i

- [Workload Automation 5 for IBM i Commands](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

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

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 [Change Release Tag](#) program, **UCHGRLS**, lets you change the tagged command names in **QSYS** to the untagged command names in **UNVPRD511**.

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

Configuration Options

The following table identifies the UQUERY for IBM i configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
CODE_PAGE	Character code page used to translate text data received and transmitted over the network.
COMMAND_ID	Requests that Universal Query return information for all records that match the specified command ID.
COMPONENT_ID	Requests that Universal Query return information only for the specified component ID.
HELP	Writes a description of the configuration options and their format.
MANAGERS	Specification for whether or not Universal Query requests manager component information from the queried Broker.
MESSAGE_LANGUAGE	Universal Message Catalog (UMC) file used to write messages.
MESSAGE_LEVEL	Level of messages to write.
PING	Information Universal Query requests from Universal Broker.
PLF_DIRECTORY	Program Lock File (PLF) directory where the program lock files are located.
REMOTE_HOST	IP address of the remote computer.
REMOTE_PORT	TCP port number on the remote computer on which Universal Broker is accepting connections.
REPORT	Format in which Universal Broker information is written.
VERSION	Writes the program version and copyright information.

Command Line Syntax

The following figure illustrates the command line syntax — using the STRUQR parameter form of command line options — of UQUERY for IBM i.

```
STRUQR
HOST(hostaddress)
[PORT(port)]
[PING(*{yes|no})]
[REPORT(*{normal|fixed})]
[CODEPAGE(codepage)]
[MSGLANG(language)]
[MSGLEVEL(*{trace|audit|info|warn|error})]
      (NOTE: trace turns on the Trace function.)
[OUTBOUNDIP(host|blank_line)]
[CMDID(ID)]
[COMPID(ID)]
[MANAGERS(*{yes|no})]
[PLFDIR(ifs_directory)]

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

Universal Query Configuration Options

Overview

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

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

Configuration Options List

The following table identifies the Universal Query configuration options.

Option Name	Description
BIF_DIRECTORY	Broker Interface Directory that specifies the location of the Universal Broker interface file.
COMMAND_ID	Requests that Universal Query return information for all records that match the specified command ID.
COMPONENT_ID	Requests that Universal Query return information only for the specified component ID.
CODE_PAGE	Character code page used to translate text data received and transmitted over the network.
HELP	Displays a description of the command line options and their format.
MANAGERS	Specification for whether or not Universal Query requests manager component information from the queried Broker.
MESSAGE_LANGUAGE	Universal Message Catalog (UMC) file used to write messages.
MESSAGE_LEVEL	Level of messages to write.
NLS_DIRECTORY	Directory where the Universal Query message catalog and code page tables are located.
OUTBOUND_IP	Sets the host or IP address that UQUERY binds to when initiating outgoing connections.
PING	Information Universal Query requests from Universal Broker.
PLF_DIRECTORY	Program Lock File directory that specifies the location of the Universal Query program lock file.
REMOTE_HOST	IP address of the remote computer.
REMOTE_PORT	TCP port number on the remote computer on which Universal Broker is accepting connections.
REPORT	Format in which Universal Broker information is written.
SYSTEM_ID	Local Universal Broker with which the Universal Query must register.
VERSION	Writes the program version and copyright information.

BIF_DIRECTORY - UQUERY 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 <i>directory</i>		✓		
Environment Variable	UQUERYBIFDIRECTORY= <i>directory</i>		✓		
Configuration File Keyword	n/a				
STRUQR Parameter	n/a				

Values

directory is the name of the BIF directory.

Default is `/var/opt/universal`.

CODE_PAGE - UQUERY configuration option

Description

The CODE_PAGE option specifies the character code page used to translate text data.

Usage

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

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.**)

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

See [UTT Files](#) for information on UTT files.

Default

The 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_ID - UQUERY configuration option

Description

The COMMAND_ID option specifies a command ID that should be used by Universal Query when searching for component records.

When COMMAND_ID is used, Universal Query will return Broker-specific information and component information for all records that match the specified command ID.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-C <i>ID</i>		✓	✓	✓
Command Line, Long Form	-cmdid <i>ID</i>		✓	✓	✓
Environment Variable	UQRYCMDID= <i>ID</i>	✓	✓	✓	✓
Configuration File Keyword	n/a				
STRUQR Parameter	CMDID(<i>ID</i>)	✓			

Values

ID is the command ID used by Universal Query.

There is no default.

COMPONENT_ID - UQUERY configuration option

Description

The COMPONENT_ID option specifies a component ID that should be used by Universal Query when searching for component records.

When COMPONENT_ID is used, Universal Query will return Broker-specific information and component information only for the specified component ID.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-c <i>ID</i>		✓	✓	✓
Command Line, Long Form	-component <i>ID</i>		✓	✓	✓
Environment Variable	UQRYCOMPONENT= <i>ID</i>	✓	✓	✓	✓
Configuration File Keyword	n/a				
STRUQR Parameter	COMPID(<i>ID</i>)	✓			

Values

ID is the component ID used by Universal Query.

There is no default.

HELP - UQUERY configuration option

Description

The HELP option displays a description of the Universal Query 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				
STRUQR Parameter	n/a				

Values

(There are no values for this option.)

MANAGERS - UQUERY configuration option

Description

The MANAGERS option specifies whether or not Universal Query requests manager component information from the Broker being queried.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-managers <i>option</i>		✓	✓	✓
Environment Variable	UQRYMANAGERS= <i>option</i>	✓	✓	✓	✓
Configuration File Keyword	managers <i>option</i>	✓	✓	✓	✓
STRUQR Parameter	MANAGERS(* <i>option</i>)	✓			

Values

option is the specification for whether or not Universal Query requests manager component information.

Valid values for option are:

- **yes**
Request manager component information.
- **no**
Do not request manager component information.

Default is yes.

MESSAGE_LANGUAGE - UQUERY configuration option

Description

The MESSAGE_LANGUAGE option specifies the Universal Message Catalog (UMC) file that will be used to write messages.

Each UMC file contains messages for a specific language.

Usage

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

Values

language is the UMC file that will be used to write messages.

The first three characters of the language name are used as a three-character suffix in the UMC file base name (for example, **uelmceng.umc**). All UMC files have a **.umc** extension.

IBM i	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 .
z/OS	The first three characters of the language name are used as a three-character suffix in the UMC member name. UMC files are read from the partitioned data set allocated on ddname UNVNLS. Universal Query message catalog member names start with UQRCM.

Default is uelmceng.umc.

MESSAGE_LEVEL - UQUERY configuration option

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-l <i>level</i>	✓	✓	✓	✓
Command Line, Long Form	-level <i>level</i>	✓	✓	✓	✓
Environment Variable	UQRYLEVEL= <i>level</i>	✓	✓	✓	
Configuration File Keyword	message_level <i>level</i>	✓	✓	✓	✓
STRUQR Parameter	MSGLEVEL(* <i>level</i>)	✓			

Values

level indicates either of the following level of messages:

- **trace**
Writes trace messages used for diagnostic purposes.

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

IBM i	info
UNIX	warn
Windows	warn
z/OS	info

Trace Files

IBM i	Trace file name is *CURLIB/UNVTRCUQR(UQRxxxxxx), where xxxxxx is the job number of the job invoking Universal Command.
UNIX	Trace file name is uquery.trc . It is created in the working directory of Universal Query.
Windows	Trace file name is uquery.trc . It is created in the working directory of Universal Query.
z/OS	Trace file name is uquery.trc . It is created in the working directory of Universal Query.

NLS_DIRECTORY - UQUERY configuration option

Description

The NLS_DIRECTORY option specifies the name of the directory where the Universal Query 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 <i>directory</i>		✓	✓	
STRUQR 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.

Default

UNIX	<code>/opt/universal/nls</code>
Windows	<code>..\nls</code>

OUTBOUND_IP - UQUERY configuration option

Description

The OUTBOUND_IP option specifies the host or IP address that Universal Query binds to when initiating outgoing connections.

By default, no host or IP address is specified for this option.

Usage

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

Values

host is the host or IP address that Universal Query binds to when initiating outgoing connections.

IBM i



Valid values for *host* are:

- *host*
Host or IP address that Universal Query binds to when initiating outgoing connections.
- (blank line)
No value is used.

PING - UQUERY configuration option

Description

The PING option specifies the type of information that Universal Query requests from Universal Broker.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-P <i>option</i>	✓	✓	✓	✓
Command Line, Long Form	-ping <i>option</i>	✓	✓	✓	✓
Environment Variable	UQRYPING= <i>option</i>	✓	✓	✓	
Configuration File Keyword	ping <i>option</i>	✓	✓	✓	✓
STRUQR Parameter	PING(* <i>option</i>)	✓			

Values

option is the specification for the type of information that Universal Query requests.

Valid values for *option* are:

- **yes**
Universal Broker information only is returned.
- **no**
Universal Broker information and Universal Broker active component information is returned.

Default is no.

PLF_DIRECTORY - UQUERY 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 Universal Query process to store manager process termination information for the Universal Broker.

IBM i



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

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	n/a				
Command Line, Long Form	-plf_directory <i>directory</i>		✓		
Environment Variable	UQRYPLFDIRECTORY= <i>directory</i>	✓	✓		
Configuration File Keyword	n/a				
STRUQR Parameter	PLFDIR(<i>directory</i>)	✓			

Values

directory is the name of the PLF directory.

A full path name must be specified.


UNIX	<code>/var/opt/universal/tmp</code>
IBM i	<code>/tmp</code>

REMOTE_HOST - UQUERY configuration option

Description

The REMOTE_HOST option specifies the IP address of the remote computer.

Note

 The remote computer must have a Universal Broker running and accepting connections.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-i <i>hostaddress</i>	✓	✓	✓	✓
Command Line, Long Form	-host <i>hostaddress</i>	✓	✓	✓	✓
Environment Variable	UQRYHOST= <i>hostaddress</i>	✓	✓	✓	
Configuration File Keyword	host <i>hostaddress</i>	✓	✓	✓	✓
STRUQR Parameter	HOST(<i>hostaddress</i>)	✓			

Values

hostaddress is the IP address of the remote computer.

The format of *hostaddress* can be either:

- IP address in dotted form (for example, 1.2.3.4)
- Host name (for example, dallas).

REMOTE_PORT - UQUERY configuration option

Description

The REMOTE_PORT option specifies the TCP port on the remote computer on which Universal Broker is accepting connections.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-p <i>port</i>	✓	✓	✓	✓
Command Line, Long Form	-port <i>port</i>	✓	✓	✓	✓
Environment Variable	UQRYPORT= <i>port</i>	✓	✓	✓	
Configuration File Keyword	port <i>port</i>	✓	✓	✓	✓
STRUQR Parameter	PORT(<i>port</i>)	✓			

Values

port is the TCP port on which Universal Broker is accepting connections.

The format of *port* can be either:

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

Default is 7887.

REPORT - UQUERY configuration option

Description

The REPORT option specifies the format in which the Universal Broker information is written.

Usage

Method	Syntax	IBM i	UNIX	Windows	z/OS
Command Line, Short Form	-R <i>format</i>	✓	✓	✓	✓
Command Line, Long Form	-report <i>format</i>	✓	✓	✓	✓
Environment Variable	UQRYREPORT= <i>format</i>	✓	✓	✓	
Configuration File Keyword	report <i>format</i>	✓	✓	✓	✓
STRUQR Parameter	REPORT(* <i>format</i>)	✓			

Values

format is the format in which the Universal Broker information is written.

Valid values for *report* are:

- **normal**
Universal Broker information is written one field per line.
- **fixed**
Universal Broker information is written in a table format with limited number of columns.

Default is normal.

SYSTEM_ID - UQUERY configuration option

Description

The SYSTEM_ID option identifies the local Universal Broker with which Universal Query must register before Universal Query 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 <i>ID</i>				✓
Environment Variable	UQRYSYSTEMID= <i>ID</i>				✓
Configuration File Keyword	n/a				
STRUQR 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.)

VERSION - UQUERY configuration option

Description

The VERSION option writes the program version and copyright information.

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				
STRUQR Parameter	VERSION(* <i>option</i>)	✓			

Values

There are no values for this option.

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 Return Code

- [Overview](#)
- [Usage](#)
- [Command Line Syntax](#)
- [Configuration Options](#)

Overview

Note



As of Windows 2000, the Universal Return Code command is no longer necessary in Windows batch files. Microsoft added the ability to specify a script return code as an argument to the EXIT command in Windows 2000 and above.

The Universal Return Code (URC) utility is a Windows utility that performs the function of ending a process with a return code that is equal to its command line argument.

The return code of a Windows batch script is the return code of the last command executed. Universal Return Code can be used as the last command to set the return code of the batch script.

Usage

The Universal Return Code program is **urc.exe**.

It exits with its integer command line argument as its return code.

Command Line Syntax

The following figure illustrates the syntax - using the command line, long form of the configuration options - of Universal Return Code.

```
urc
return_code

urc
{ -help | -version }
```

Configuration Options

The command line arguments to Universal Return Code are:

Argument	Description
<i>return_code</i>	Integer-value return code of Universal Return Code.
-? -h -help	Write command instructions.
-v -version	Write version information.

Universal Spool List

- [Universal Spool List](#)
- [Usage](#)
- [Detailed Information](#)

Universal Spool List

The Universal Spool List (USLIST) utility provides the ability to list Universal Spool database records. Universal Spool List must be executed on the system on which the database is located.

The functions that Universal Spool List provide are required for possible database clean-up or problem resolution by Stonebranch, Inc. Customer Support.

Usage

The Universal Spool List utility reads requested records from a specified database. The selected records are written to standard output.

Universal Spool List performs operations specified by the command options.

Detailed Information

The following pages provide detailed information for Universal Spool List:

- [Universal Spool List for z/OS](#)
- [Universal Spool List for Windows and UNIX](#)
- [Universal Spool List for IBM i](#)
- [Universal Spool List Output](#)
- [Universal Spool List Configuration Options](#)

Universal Spool List for zOS

- [Databases](#)
 - [HFS Data Sets](#)
 - [zFS Data Sets](#)
- [JCL Procedure](#)
- [DD Statements used in JCL Procedure](#)
- [JCL](#)
- [Configuration Options](#)

Databases

Universal Spool databases are implemented as HFS or zFS data sets.

HFS Data Sets

Universal Spool List (**USLIST**) and [Universal Spool Remove \(USLRM\)](#) obtain the HFS data set names from **UNVDB** and **UNVSPPOOL** ddnames.

The following table identifies the database files and the HFS data sets in which they reside.


Database Name	Data Set	File Name
Universal Broker Component Database	UNVDB	bcomponent.db
Universal Server Component Database	UNVDB	scomponent.db
Universal Server Spool Databases *	UNVSPPOOL	spool.stdin. <i>COMPID</i> .db spool.stdout. <i>COMPID</i> .db spool.stderr. <i>COMPID</i> .db

* The *COMPID* in the Universal Server Spool Databases file names is the component ID assigned to the Server instance.

zFS Data Sets

zFS data sets cannot be provided via ddnames. To obtain the zFS data set names, **USLIST** and **USLRM** allocate and parse the Universal Broker's configuration member, **UBRCFG00**, for the [UNIX_DB_DATA_SET](#) and [UNIX_SPOOL_DATA_SET](#) options. If these options are not found, **USLIST** and **USLRM** assume that HFS data sets are being used and refer to the **UNVDB** and **UNVSPPOOL** ddnames for the HFS data set names.

Note

 You can use these configuration options to specify HFS data sets as well if this method is preferred over specifying them as a ddname allocation.

USLIST and **USLRM** mount the HFS and zFS data sets if they are not already mounted.

The Universal Agent configuration PDSE is allocated to **UNVCONF** in the **USLLSPRC** and **USLRMPRC** JCL procedures in the **SUNVSAMP** library. The only other changes to **USLLSPRC** and **USLRMPRC** is the addition of a comment that the **UNVDB** and **UNVSPPOOL** ddnames must be commented out to use zFS data sets.

USLIST and **USLRM** write messages UNV2264I and UNV2265I, which provide information on the Broker's database and spool, respectively. The messages also provide the file system type, data set name, and mount point.

JCL Procedure

The following figure illustrates the Universal Spool List for z/OS JCL procedure (**USLLSPRC**, located in the **SUNVSAMP** library) that is provided to simplify the execution JCL and future maintenance.

```
//USLLSPRC PROC UPARM=,          -- USLIST options
//          UNVPRE=#SHLQ.UNV,
//          CFGPRE=#PHLQ.UNV,
//          DBPRE=#PHLQ.UNV
// *
//PS1      EXEC PGM=USLIST, PARM=' ENVAR(TZ=EST5EDT) /&UPARM '
//STEPLIB DD  DISP=SHR, DSN=&UNVPRE..SUNVLOAD
// *
//UNVNLS  DD  DISP=SHR, DSN=&UNVPRE..SUNVNLS
//UNVCONF DD  DISP=SHR, DSN=&CFGPRE..UNVCONF
//UNVDB   DD  DISP=SHR, DSN=&DBPRE..UNVDB
//UNVSPOOL DD DISP=SHR, DSN=&DBPRE..UNVSPOOL
//UNVTRACE DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSOUT  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

Note



If zFS data sets are being used instead of HFS data sets, the UNVDB and UNVSPOOL ddnames cannot be used to allocate those zFS data sets. They must be removed, along with the DBPRE procedure parameter.

DD Statements used in JCL Procedure

The following table describes the DD statements used in the Universal Spool List for z/OS [JCL procedure](#), above.

ddname	Description
STEPLIB	Load library in which program USLIST program is located.
UNVNLS	Universal National Language Support library.
UNVCONF	Universal Agent configuration library.
UNVDB	Universal Broker Database data set (HFS only).
UNVSPOOL	Universal Spool Database data set (HFS only).
UNVTRACE	Application trace ddname.
SYSOUT	USLIST standard error ddname.
SYSPRINT	USLIST standard output ddname.

JCL

The following figure illustrates the Universal Spool List for z/OS JCL using the **USLLSPRC** [JCL procedure](#), above (for HFS data sets).

```
//STEP1 EXEC PGM=USLIST
//STEPLIB DD DISP=SHR,DSN=UNV.SUNVLOAD
//UNVNLS DD DISP=SHR,DSN=UNV.SUNVNLS
//UNVCONF DD DISP=SHR,DSN=&CFGPRE..UNVCONF
//UNVDB DD DISP=SHR,DSN=UNV.UNVDB
//UNVSPool DD DISP=SHR,DSN=UNV.UNVSPool
//UNVTRACE DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
command options
/*
```

Configuration Options

The following table identifies the Universal Spool List for z/OS configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
COMPONENT	Component identifier for which records will be selected to write.
HELP	Writes a description of the command options and their format.
LIST	Type of database from which to select record to write.
MESSAGE_LEVEL	Level of messages that will be written.
MOUNT_POINT	HFS directory in which the HFS databases allocated to ddnames UNVDB and UNVSPool are mounted.
VERSION	Writes the program version and copyright information.

```
uslist
[-list {ubroker|ucmd|urmtcfg|stdin|stderr|stdout}]
[-component cid]
[-mount_point directory]
[-level {audit|info|warn|error}]
```

```
uslist
{ -help | -version }
```

Universal Spool List for Windows and UNIX

- [Databases](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

Databases

The Universal Spool List utility can be used to read the databases listed in the following table.

Database Name	Database File Name
Universal Broker Component Database	bcomponent.db
Universal Server Component Database	scomponent.db
Universal Server Spool Databases *	spool.stdin. <i>COMPID</i> .db spool.stdout. <i>COMPID</i> .db spool.stderr. <i>COMPID</i> .db
Universal Event Monitor Event Definition Database	ueme.db
Universal Event Monitor Event Handler Database	uemh.db
Universal Event Monitor Spool Database	uems.db

* The *COMPID* in the Universal Server Spool Databases file names is the component ID assigned to the Server instance.

UNIX	By default, the database files are located in the <code>/var/opt/universal/spool</code> directory. The program file is located in the Universal Spool installation directory <code>bin</code> directory, which defaults to <code>/opt/universal/uspool-5.2.0/bin</code> .
Windows	By default, the database files are stored in the <code>C:\Program Files\Universal\spool</code> directory. The Universal Spool List program file, <code>uslist.exe</code> , is located in the <code>{bin}</code> subdirectory of the Universal Spool installation directory, which defaults to <code>C:\Program Files\Universal\uspool</code> .

Configuration Options

The following table identifies the Universal Spool List for Windows and UNIX configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
BROKER_SPOOL_DIR	Directory location in which the Universal Broker Component database is located.
COMPONENT	Component identifier for which records will be selected to write.
HELP	Writes a description of the command options and their format.
ID	Lists the contents of a specific record from the Universal Event Monitor event definition, event handler, or spool databases.
LIST	Type of database from which to select record to write.
MESSAGE_LEVEL	Level of messages that will be written.
UCMD_SPOOL_DIR	Directory location in which the Universal Server Component database is located.
VERSION	Writes the program version and copyright information.

Command Line Syntax

The following figure illustrates the command line syntax — using the command line, long form of the configuration options — of Universal Spool List for Windows and UNIX.

```
uslist
[-list {ubroker|ucmd|urmtcfg|stdin|stderr|stdout}]
[-component cid]
[-id id]
[-brokerspooldir directory]
[-ucmdspooldir directory]
[-level {audit|info|warn|error}]

uslist
{ -help | -version }
```

Universal Spool List for IBM i

- [Databases](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

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

Databases

The Universal Spool List utility can be used to read the databases listed in the following table.

Database Name	Database File Name
Universal Broker Component Database	UBR_CMP_DB
Universal Server Component Database	SRV_CMP_DB
Universal Server Spool Databases *	Slcompid (STDIN) SOcompid (STDIN) SEcompid (STERR)
I-Management Console Remote Configuration Database	UNVCFG_DB
* The compid in the Universal Server Spool Databases file names is the component ID (in hexadecimal) assigned to the Server instance.	

The spool files are located in library 1.

Configuration Options

The following table identifies the Universal Spool List for IBM i configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
COMPONENT	Component identifier for which records will be selected to write.
ID	Lists the contents of a specific record from the Universal Event Monitor event definition, event handler, or spool databases.
LIST	Type of database from which to select record to write.
MESSAGE_LEVEL	Level of messages that will be written.
VERSION	Writes the program version and copyright information.

Command Line Syntax

The following figure illustrates the command line syntax — using the ULSTSE parameter form of command line options — of Universal Spool List for IBM i.

```

ULSTSE
[LIST(*{ubroker|ucmd|urmtcfg|stdin|stderr|stdout})]
[COMPONENT(cid)]
[ID(id)]
[LEVEL(*{audit|info|warn|error})]

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

Universal Spool List Output

Universal Spool List Output

Universal Spool List lists output for the following Universal Agent components:

- [Universal Broker](#)
- [Universal Command](#)
- [Universal Event Monitor](#)

Universal Broker Output

- [Universal Broker Component](#)
- [Universal Broker Component List](#)

Universal Broker Component

The following table identifies the information written for a requested Universal Broker component.

Field Name	Description
ID	Component identifier.
Name	Component definition name. Either ucmd or uctl .
Desc	Component description field from the component definition.
Version	Component version and build level.
State	Component state.
Cmd ID	Command identifier provided by the manager.
Comm State	Component communication state.
Comp State Time	Date and time the component entered the communication state.
Restartable	Specification for whether or not the component is restartable (manager fault tolerant).
Srv PID	Component's process identifier.
Srv Start Time	Components start date and time.
Srv End Time	Component's end date and time.
Srv Exit Code	Component's exit code if its status is not RUNNING.
Srv Exit Status	Component's execution status.
Mgr UID	Manager's user identifier.
Mgr Work ID	Manager's work identifier.
Mgr Host Name	Manager's TCP/IP host name on which it's executing.
Mgr Port	Manager's TCP/IP port number from which it connected.

Universal Broker Component List

The following table identifies the columns of data in a Universal Broker component list.

Column Name	Description
ID	Component Identifier
NAME	Component name. Either ucmd or uctl .

CST	<p>Component's communication state. Communication state values are:</p> <ul style="list-style-type: none"> • COM Component is completed. • DIS Communication link between the component and the manager is disconnected. The status of the manager is unknown. The network fault tolerant protocol is being used. • EST Communication link between the component and the manager is established. This is the normal operating mode. • ORH Component is executing, but the manager has terminated. The manager is orphaned. The component was started with manager fault tolerance and is waiting for a manager restart. The user process is still executing. • PEN Component has completed its work and is waiting for a manager to restart to receive the user process spool files and exit status. The component was started with manager fault tolerance. • RCG Component is in the middle of reconnecting the manager. The network fault tolerant protocol is being used. • RSG Component is in the middle of restarting with a manager. The component was started with manager fault tolerance. • RSA Manager restart request has been accepted. The manager and the component will be reestablishing their communication links. • STR Component is starting. The component usually remains in this state for a short period of time unless they are executing with manager fault tolerance and the manager is redirecting a large stdin file.
MGR-WORK-ID	<p>Manager's work identifier. The work ID format depends on the system type on which the manager is executing.</p>
COMMAND-ID	<p>Command identifier specified by the manager.</p>

Universal Command Output

- [Universal Command Server Component](#)
- [Universal Command Server Component List](#)

Universal Command Server Component

The following table identifies the information in a requested Universal Command Server component.

Field Name	Description
ID	Component identifier.
Cmd Line	User command the manager requested to be executed.
User ID	User identifier with which the user command is executing.
Desc	Component description.
Comm State	Component communication state.
Comp State Time	Date and time the component entered the communication state.
Restartable	Specification for whether or not the component is restartable (manager fault tolerant).
Spool Retention	Number of days to retain the spool files after the component goes into completed state.
Comp Retention	Number of days to retain the component record after the component goes into completed state.
PID	User command's process identifier.
Start Time	User process's start date and time.
End Time	User process's end date and time.
Exit Code	User process's exit code if it is not RUNNING.
Exit Status	User process's exit status.
Script File	Script file being executed by the Server.

Universal Command Server Component List

The following table identifies the columns of data in a Universal Command Server component record.

Column Name	Description
ID	Component Identifier.
CST	Component's communication state. (See CST in Universal Broker Component List .)
USER-ID	Local user account with which the user process is being executed.
COMMAND	Command which the manager requested to be executed.

Universal Event Monitor Output

- [Universal Event Monitor Event Definition](#)
- [Event Type-Specific Fields](#)
 - [FILE Event Definitions](#)
- [Universal Event Monitor Event Definitions List](#)
- [Universal Event Monitor Event Handler](#)
- [Universal Event Monitor Event Handler List](#)
- [Universal Event Monitor Spool List](#)
- [Universal Event Monitor Spool Record](#)

Universal Event Monitor Event Definition

The following table identifies the information that is displayed for a requested Universal Event Monitor event definition.

Field Name	Description
Event ID	Event definition identifier.
Event Type	Type of system event that the event definition is responsible for detecting and monitoring. The following event types are supported: FILE Instructs UEM to detect the creation of a file and track its progress until it meets its specified completion criteria.
Component Name	Name of the event-driven Universal Event Monitor Server component to which the event definition is assigned. This is the UEM Server component responsible for monitoring the event.
Enabled	Indicates whether the event definition is currently recognized by its assigned UEM Server. An enabled event definition will be monitored as long as the current date and time fall within the activation and inactivation times. A disabled event definition will never become active and will never be monitored, unless it is explicitly enabled.
Active	Indicates whether the event definition is currently being monitored by its assigned UEM Server. An event definition must be enabled before it will be made active. The event will be made inactive once its inactivation time elapses.
Activation Time	Date and time at which the assigned UEM Server component will begin monitoring this event.
Inactivation Time	Date and time at which the assigned UEM Server component will stop monitoring this event.
Tracking Interval	Frequency, in seconds, with which UEM will test for the completion of any system occurrence detected for this event.
Triggered Handler	ID of a record stored in the event handler database that should be executed whenever the processing state for an occurrence of this event is set to TRIGGERED.
Expired Handler	ID of a record stored in the event handler database that should be executed whenever the processing state for this event is set to EXPIRED.
Rejected Handler	ID of a record stored in the event handler database that should be executed whenever the processing state for an occurrence of this event is set to REJECTED.
Handler Options	Parameters that UEM adds to the command line used to execute the event handler process. The event handler process receives these parameters as command line options.
Last Modified On	Date and time the event definition record was last updated.
Last Modified By	Name of the user account that last updated the event definition record.

Event Type-Specific Fields

This section describes the event definition fields that vary depending on the value of the Event Type parameter.

FILE Event Definitions

The following table identifies the fields that are displayed for events with an event type of **FILE**.

Field Name	Description
File Specification	File whose creation should be detected and whose progress should be tracked by UEM.
Minimum File Size	Smallest size a file must be before it is considered complete by UEM.
Rename File	Indication of whether or not the file will be renamed by UEM whenever the processing state of the tracked event occurrence is set to TRIGGERED.
Rename Specification	Format that UEM should use when renaming a file whose event occurrence has been set to the TRIGGERED state.

Universal Event Monitor Event Definitions List

The following table identifies the items for which values are defined in a Universal Event Monitor Event Definitions list.

Column Name	Description
EVENT ID	Event Definition Identifier.
TYPE	Type of system event that the event definition is responsible for detecting and monitoring. For a complete list of supported event types, see Event Type in #Universal Event Monitor Event Definition .
ENABLED	Indication of whether or not the event definition currently is being processed by its assigned UEM Server.
ACTIVE	Indication of whether or not the event definition currently is being monitored by its assigned UEM Server.

Universal Event Monitor Event Handler

The following table identifies the information displayed for a requested Universal Event Monitor event handler.

Field Name	Description
Handler ID	Event Handler Identifier.
Handler Type	Process which is executed on behalf of the event handler. The following process types are supported: <ul style="list-style-type: none"> CMD Indicates the record contains the name of an application, along with all of its required command line parameters, that is to be executed on behalf of the event handler. SCRIPT Indicates the record contains a set of one or more system commands that are to be executed as a single script on behalf of the event handler.
Max Acceptable Return Code	Highest value an event handler process may return to still be considered as having executed successfully.
User ID	ID of the user account in whose security context the event handler process will be executed.
Command	If the value of the Handler Type parameter is CMD , this field shows the command to execute. This field will not be shown if the value of the Handler Type parameter is SCRIPT .
Script Statements	If the value of the Handler Type parameter is SCRIPT , this field marks the beginning of the system commands that will be executed as a script. This field will not be shown if the value of the Handler Type parameter is CMD .
Script Type	Type of script statements to execute when the value of the Handler Type parameter is SCRIPT . This field will not be shown if the value of the Handler Type parameter is CMD .
Last Modified On	Date and time the event handler record was last updated.
Last Modified By	Name of the user account that last updated the event handler record.

Universal Event Monitor Event Handler List

The following table identifies the items for which values are defined in a Universal Event Monitor Event Handlers list.

Column Name	Description
HANDLER ID	Event Handler Identifier.
TYPE	Describes the process which is executed on behalf of the event handler. For a complete list of supported process types, see Handler Type in #Universal Event Monitor Event Handler .

Universal Event Monitor Spool List

The following table identifies the items for which values are listed in Universal Event Monitor Spool List.

Column Name	Description
SERIAL NO	Sequential number that is assigned to each record as it is added to the database. This number serves to uniquely identify each occurrence of a given event definition.
EVENT ID	ID of the event definition responsible for the detection and monitoring of the event occurrence recorded by this spool record.
PRC STATE	Processing state of the event occurrence. For a complete list of possible values, see the description of the Processing State parameter, below.
HANDLER ID	ID of an event handler executed whenever the processing state of an event or an event occurrence enters the TRIGGERED, REJECTED, or EXPIRED state.
EXIT CODE	Value returned by the process executed on behalf of an event handler.
EXIT STATUS	Indicates whether the event handler process ended normally or whether it was terminated unexpectedly.
HANDLER STATUS	Indicates the outcome of event handler processing. For a list of possible values, see Handler Status in #Universal Event Monitor Spool Record .

Universal Event Monitor Spool Record

The following table identifies the information displayed for a requested Universal Event Monitor spool record.

Field Name	Description
Serial No	Sequential number that is assigned to each record as it is added to the database. This number serves to uniquely identify each occurrence of a given event definition.
Event ID	ID of the event definition responsible for the detection and monitoring of the event occurrence recorded by this spool record.
Component Name	Name of the event-driven Universal Event Monitor Server component to which the event definition is assigned. This is the UEM Server component responsible for monitoring and processing the event.
Component Description	Description of the UEM Server component identified by Component Name.
Component Version	Version of the UEM Server component identified by Component Name.
Component ID	Value that uniquely identifies the instance of the UEM Server component that processed the event occurrence.
Event Type	Type of system event that the event definition is responsible for detecting and monitoring.
System Object	System event detected and monitored by the event occurrence.
Processing State	<p>Processing state of the event occurrence.</p> <p>The following values are used:</p> <ul style="list-style-type: none"> • TRACKING Indicates that an occurrence of a system event described by an event definition was detected, but has not yet met the completion criteria set forth by the event definition and by UEM's application logic. • TRIGGERED Indicates that an occurrence of a system event described by an event definition was detected and has completed. If a triggered event handler was specified in the event definition, that handler's process will be executed and the event handler's ID will be shown in the Handler ID field. • REJECTED Indicates that an occurrence of a system event described by an event definition was detected, but failed to complete before the date and time specified in the event definition's Inactivation Time parameter. If a rejected event handler was specified in the event definition, that handler's process will be executed and the event handler's ID will be shown in the Handler ID field. • EXPIRED Indicates that no occurrence of the system event described by an event definition was detected before the event's Inactivation Time elapsed. If an expired event handler was specified in the event definition, that handler's process will be executed and the event handler's ID will be shown in the Handler ID field. • ERROR Indicates an error occurred while processing the event occurrence.

Handler ID	ID of an event handler executed whenever the processing state of an event or an event occurrence enters the TRIGGERED, REJECTED, or EXPIRED state.
User Command	Command executed on behalf of the event handler when the Handler Type is CMD. If the Handler Type is SCRIPT, this field contains no value.
Process ID	ID of the process executed by UEM on behalf of the event handler.
User ID	Name of the user account in whose security context the event handler process was executed.
Start Time	Date and time the event handler process started.
End Time	Date and time the event handler process ended.
Exit Code	Value returned by the event handler process.
Exit Status	Indicates whether the event handler process ended normally or whether it terminated unexpectedly.
Handler Status	<p>Indicates the outcome of event handler processing.</p> <p>The following values are used:</p> <ul style="list-style-type: none"> • FAILED The event handler process finished abnormally, or ended normally with an exit code greater than the maximum acceptable return code specified in the event handler record. • NO HANDLER No event handler was specified for the event's processing state. • NOT AUTHORIZED An attempt to execute the event handler process failed because the user ID or password specified for the event handler was incorrect. • NOT FOUND The ID of an event handler record specified for a particular processing state was not found in the event handler database. • SHUTDOWN The Universal Event Monitor Server was stopped while the event handler process was running. • SUCCESSFUL The event handler process completed normally and exited with a value that was less than or equal to the maximum acceptable return code specified in the event handler record. • UNRECOVERABLE Information for the event handler process could not be recovered.
Last Modified On	Date and time the spool record was last updated.

Universal Spool List Configuration Options

Overview

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

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

Configuration Options List

The following table identifies the Universal Spool List configuration options.

Option Name	Description
BROKER_SPOOL_DIR	Directory location in which the Universal Broker Component database is located.
COMPONENT	Component identifier for which records will be selected to write.
HELP	Displays a description of command line options and their format.
ID	Lists the contents of a specific record from the Universal Event Monitor event definition, event handler, or spool databases.
LIST	Type of database from which to select records to write.
MESSAGE_LEVEL	Level of messages to be written.
MOUNT_POINT	HFS directory in which the HFS databases allocated to ddnames UNVDB and UNVSPool are mounted.
UCMD_SPOOL_DIR	Directory location in which the Universal Server Component database is located.
VERSION	Writes the program version and copyright information.

BROKER_SPOOL_DIR - USLIST configuration option

Description

The BROKER_SPOOL_DIR option specifies the directory in which the Universal Broker component database (**bcomponent.db**) is located.

If this option is not used to specify the directory, the directory is read from the Universal Broker configuration file.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-b <i>directory</i>			✓	✓	
Command Line, Long Form	-brokerspooldir <i>directory</i>			✓	✓	
ULSTSE Parameter	n/a					

Values

directory is the directory in which the Universal Broker component database is located.

COMPONENT - USLIST configuration option

Description

The COMPONENT option specifies the ID of a single component (job) for which records will be selected to be written.

The LIST option identifies the database of the component.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-c cid</code>			✓	✓	✓
Command Line, Long Form	<code>-component cid</code>			✓	✓	✓
ULSTSE Parameter	COMPONENT(<i>cid</i>)	✓				

Values

cid is the component ID.

HELP - USLIST configuration option

Description

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

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-h			✓	✓	✓
Command Line, Long Form	-help			✓	✓	✓
ULSTSE Parameter	n/a					

Values

(There are no values for this option.)

ID - USLIST configuration option

Description

The ID option specifies the ID of a single record (from the Universal Event Monitor event definition, event handler, or spool databases) to be written.

The [LIST](#) option specifies the spool database.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-i <i>id</i>			✓	✓	
Command Line, Long Form	-id <i>id</i>			✓	✓	
ULSTSE Parameter	ID(<i>id</i>)	✓				

Values

id is the ID of a specific record.

- For an event definition or event handler record, *id* is the event ID or event handler ID, respectively.
- For a spool database record, *id* is the sequential serial number automatically assigned to the record.

LIST - USLIST configuration option

Description

The LIST option specifies the database from which to select records to write.

- Universal Broker
- Universal Command Server
- Universal Event Monitor
- Spool

The **COMPONENT** option is used to select individual component (job) records from the database.

The **ID** option is used to select a single record from a Universal Event Monitor database.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a			✓	✓	✓
Command Line, Long Form	-list <i>option</i>			✓	✓	✓
ULSTSE Parameter	LIST(* <i>option</i>)	✓				

Values

option is the database from which to select records.

Valid values for *option* are:

ubroker	List the contents of the Universal Broker Component database. A summary of all records is written.
ucmd	List the contents of the Universal Command Server Component database. A summary of all records is written.
ueme	UNIX and Windows only: List the contents of the Universal Event Monitor Event Definition database. A summary of all records is written.
uemh	UNIX and Windows only: List the contents of the Universal Event Monitor Event Handler database. A summary of all records is written. Use the ID option to list the complete contents of a single record.
uems	UNIX and Windows only: List the contents of the Universal Event Monitor Spool database. A summary of all records is written. Use the ID option to list the complete contents of a single record.
urmtcfg	List the contents of the I-Management Console Remote Configuration database. A summary of all records is written. Use the ID option to list the complete contents of a single record.
stdin	List the standard input spool file for a specified component.
stderr	List the standard error spool file for a specified component.
stdout	List the standard output spool file for a specified component.

Default is ubroker.

MESSAGE_LEVEL - USLIST configuration option

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l <i>level</i>			✓	✓	✓
Command Line, Long Form	-level <i>level</i>			✓	✓	✓
ULSTSE Parameter	LEVEL(* <i>level</i>)	✓				

Values

level indicates either of the following level of messages:

- **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

IBM i	info
UNIX	warn
Windows	warn
z/OS	info

MOUNT_POINT - USLIST configuration option

Description

The MOUNT_POINT option specifies the HFS directory in which the HFS database allocated to ddnames UNVDB and UNVSPool are mounted.

The actual mount points will be subdirectories named after the HFS data set names being mounted.

If the mount points do not exist, they are created by Universal Spool List.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-mount_point <i>directory</i>					✓
ULSTSE Parameter	n/a					

Values

directory is the HFS directory in which the HFS databases are mounted.

Default is /tmp.

UCMD_SPOOL_DIR - USLIST configuration option

Description

The UCMD_SPOOL_DIR option specifies the directory in which the Universal Server component database (**scomponent.db**) is located.

If this option is not used to specify the directory, the directory of the Universal Broker component database is used (see the [BROKER_SPOOL_DIR](#) option).

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-u <i>directory</i>			✓	✓	
Command Line, Long Form	-ucmdspooldir <i>directory</i>			✓	✓	
ULSTSE Parameter	n/a					

Values

directory is the directory of the Universal Server component database.

VERSION - USLIST configuration option

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-v			✓	✓	✓
Command Line, Long Form	-version			✓	✓	✓
ULSTSE Parameter	VERSION(* <i>option</i>)	✓				

Values

There are no values for this option.

IBM i



Valid values for *option* are:

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

Universal Spool Remove

- [Universal Spool Remove](#)
- [Prerequisite to Running Universal Spool Remove](#)
- [Usage](#)
- [Detailed Information](#)

Universal Spool Remove

Universal Spool Remove (USLRM) utility provides the ability to remove component records from the Universal Command and Universal Event Monitor (UNIX and Windows only) Spool databases. Universal Spool Remove must be executed on the system upon which the database is located.

By default, spool records are not retained after they no longer are needed. Accordingly, it is not anticipated that the spool databases will become too large. However, on occasion, some records may not be cleaned up, making it necessary to remove them with the Universal Spool Remove utility.

Prerequisite to Running Universal Spool Remove

Before attempting to remove any records using Universal Spool Remove, ensure that the Universal Broker is not running on the local system.

While it is active, the Universal Broker, in its role as a local database administrator, actually "owns" and maintains an open reference to the spool databases. Any changes made to these databases outside of the Broker are not committed to the database while this reference is open. If Universal Spool Remove removes a spool record while the Broker is running, that same record will "reappear" the next time that the Broker commits an update to the database (for example, a new server component is started and recorded in the spool).

All spool records that are deleted as part of the regular component clean-up are permanent because those deletions are done via the Broker.

The functions that the Universal Spool List program provide are required for possible database clean-up (see [Universal Spool List Configuration Options](#)).

Usage

The Universal Spool Remove utility removes all records for a specified component ID from the Universal Spool databases.

Any errors encountered while records are being removed from a database will be reported, but will not result in the program being stopped.

Universal Spool Remove removes records as specified by the command options.

Detailed Information

The following pages provide additional detailed information for Universal Spool Remove:

- [Universal Spool Remove for z/OS](#)
- [Universal Spool Remove for Windows and UNIX](#)
- [Universal Spool Remove for IBM i](#)
- [Universal Spool Remove Configuration Options](#)

Universal Spool Remove for zOS

- [Databases](#)
 - [HFS Data Sets](#)
 - [zFS Data Sets](#)
- [JCL Procedure](#)
- [DD Statements used in JCL Procedure](#)
- [JCL](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

Databases

Universal Spool databases are implemented as HFS or zFS data sets.

HFS Data Sets

Universal Spool Remove (**USLRM**) and [Universal Spool List \(USLIST\)](#) obtain the HFS data set names from **UNVDB** and **UNVSPPOOL** ddnames.

The following table identifies the database files and the HFS data sets in which they reside.

Database Name	Data Set	File Name
Universal Broker Component Database	UNVDB	bcomponent.db
Universal Server Component Database	UNVDB	scomponent.db
Universal Server Spool Databases *	UNVSPPOOL	spool.stdin. <i>COMPID</i> .db spool.stdout. <i>COMPID</i> .db spool.stderr. <i>COMPID</i> .db

* The *COMPID* in the Universal Server Spool Databases file names is the component ID assigned to the Server instance.

zFS Data Sets

zFS data sets cannot be provided via ddnames. To obtain the zFS data set names, **USLRM** and **USLIST** allocate and parse the Universal Broker's configuration member, **UBRCFG00**, for the [UNIX_DB_DATA_SET](#) and [UNIX_SPOOL_DATA_SET](#) options. If these options are not found, **USLRM** and **USLIST** assume that HFS data sets are being used and refer to the **UNVDB** and **UNVSPPOOL** ddnames for the HFS data set names.

Note



You can use these configuration options to specify HFS data sets as well if this method is preferred over specifying them as a ddname allocation.

USLRM and **USLIST** mount the HFS and zFS data sets if they are not already mounted.

The Universal Agent configuration PDSE is allocated to **UNVCONF** in the **USLLSPRC** and **USLRMPRC** JCL procedures in the **SUNVSAMP** library. The only other changes to **USLLSPRC** and **USLRMPRC** is the addition of a comment that the **UNVDB** and **UNVSPPOOL** ddnames must be commented out to use zFS data sets.

USLRM and **USLIST** write messages UNV2264I and UNV2265I, which provide information on the Broker's database and spool, respectively. The messages also provide the file system type, data set name, and mount point.

JCL Procedure

The following figure illustrates the Universal Spool Remove for z/OS JCL procedure (**USLRMPRC**, located in the **SUNVSAMP** library) that is provided to simplify the execution JCL and future maintenance.

```
//USLRMPRC PROC UPARM=,          -- USLRM options
//          UNVPRE=#SHLQ.UNV,
//          CFGPRE=#PHLQ.UNV,
//          DBPRE=#PHLQ.UNV
// *
//PS1      EXEC PGM=USLRM, PARM= ' ENVAR ( TZ=EST5EDT ) / &UPARM '
//STEPLIB DD DISP=SHR, DSN=&UNVPRE . . SUNVLOAD
// *
//UNVNLS  DD DISP=SHR, DSN=&UNVPRE . . SUNVNLS
//UNVCONF DD DISP=SHR, DSN=&CFGPRE . . UNVCONF
//UNVDB   DD DISP=SHR, DSN=&DBPRE . . UNVDB
//UNVSPool DD DISP=SHR, DSN=&DBPRE . . UNVSPool
//UNVTRACE DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSOUT  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

Note



If zFS data sets are being used instead of HFS data sets, the UNVDB and UNVSPool ddnames cannot be used to allocate those zFS data sets. They must be removed, along with the DBPRE procedure parameter.

DD Statements used in JCL Procedure

The following table describes the DD statements used in the Universal Spool Remove for z/OS [JCL procedure](#), above.

ddname	Description
STEPLIB	Load library in which program USLRM program is located.
UNVNLS	Universal National Language Support library.
UNVCONF	Universal Agent configuration library.
UNVDB	Universal Broker Database HFS data set.
UNVSPool	Universal Spool Database HFS data set.
UNVTRACE	Application trace ddname.
SYSOUT	USLRM standard error ddname.
SYSPRINT	USLRM standard output ddname.

JCL

The following figure illustrates the Universal Spool Remove for z/OS JCL using the **USLRMPRC** [JCL procedure](#), above.

```
//STEP1 EXEC PGM=USLRM
//STEPLIB DD DISP=SHR, DSN=UNV . SUNVLOAD
//UNVNLS DD DISP=SHR, DSN=UNV . SUNVNLS
//UNVCONF DD DISP=SHR, DSN=&CFGPRE . . UNVCONF
//UNVDB DD DISP=SHR, DSN=UNV . UNVDB
//UNVSPool DD DISP=SHR, DSN=UNV . UNVSPool
//UNVTRACE DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
command options
/*
```

If zFS data sets are being used instead of the default HFS data sets, the UNVDB and UNVSPool ddnames cannot be used to allocate the zFS data sets. The ddnames must be removed along with the DBPRE procedure parameter.

Configuration Options

The following table identifies the Universal Spool Remove for z/OS configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
COMPONENT	Component identifier for which records will be removed.
HELP	Writes a description of the command options and their format.
MESSAGE_LEVEL	Level of messages that will be written.
MOUNT_POINT	HFS directory in which the HFS databases allocated to ddnames UNVDB and UNVSPool are mounted.
VERSION	Writes the program version and copyright information.

Command Line Syntax

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

```

uslrm
  -component cid
  [-mount_point dir]
  [-level {audit|info|warn|error}]

uslrm
{ -help | -version }
```

Universal Spool Remove for Windows and UNIX

- [Databases](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

Databases

The following table identifies the databases from which Universal Spool Remove can remove records.

Database Name	Default File Name
Universal Broker Component Database	bcomponent.db
Universal Server Component Database	scomponent.db
Universal Server Spool Databases *	spool.stdin. <i>COMPID</i> .db spool.stdout. <i>COMPID</i> .db spool.stderr. <i>COMPID</i> .db
Universal Event Monitor Spool Database	uems.db

* In the Universal Server Spool Databases file names, *COMPID* is the component ID assigned to the Server instance.

UNIX	By default, the database files are located in the <code>/var/opt/universal/spool</code> directory. The program file is located in the Universal Spool installation directory <code>bin</code> directory, which defaults to <code>/opt/universal/uspool/bin</code> .
Windows	By default, the database files are stored in the <code>C:\Program Files\Universal\spool</code> directory. The Universal Spool Remove program file, <code>uslrm.exe</code> , is located in the <code>bin</code> subdirectory of the Universal Spool installation directory, which defaults to <code>C:\Program Files\Universal\uspool</code> .

Configuration Options

The following table identifies the Universal Spool Remove for Windows and UNIX configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
BROKER_SPOOL_DIR	Directory location in which the Universal Broker Component database is located.
COMPONENT	Component identifier for which records will be removed.
HELP	Writes a description of the command options and their format.
MESSAGE_LEVEL	Level of messages that will be written.
UCMD_SPOOL_DIR	Directory location in which the Universal Server Component database is located.
UEM_SERIALNO	Serial number of the Universal Event Monitor spool database record to remove.
VERSION	Writes the program version and copyright information.

Command Line Syntax

The following figure illustrates the command line syntax — using the command line, long form of the configuration options — of Universal Spool Remove for Windows and UNIX.

```
uslrm
-component cid | -UEM_SERIALNO cid
[-brokerspooldir dir]
[-ucmdspooldir dir]
[-level {audit|info|warn|error}]

uslrm
{ -help | -version }
```

Universal Spool Remove for IBM i

- [Databases](#)
- [Configuration Options](#)
- [Command Line Syntax](#)

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

Databases

The Universal Spool Remove utility can be used to read the databases listed in the following table.

Database Name	Database File Name
Universal Broker Component Database	UBR_CMP_DB
Universal Server Component Database	SRV_CMP_DB

The spool files are located in library **UNVUSL511**.

Configuration Options

The following table identifies the Universal Spool Remove for IBM i configuration options. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
COMPONENT	Component identifier for which records will be removed.
MESSAGE_LEVEL	Level of messages that will be written.
VERSION	Writes the program version and copyright information.

Command Line Syntax

The following figure illustrates the command line syntax — using the URMVSE parameter form of command line options — of Universal Spool Remove for IBM i.

```

URMVSE
[COMPONENT(cid)]
[LEVEL(*{audit|info|warn|error})]

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

Universal Spool Remove Configuration Options

Overview

This page provides detailed information on the configuration options available for use with the Universal Spool Remove utility.

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

Configuration Options List

The following table identifies the Universal Spool Remove configuration options.

Option Name	Description
BROKER_SPOOL_DIR	Directory location in which the Universal Broker Component database is located.
COMPONENT	Component identifier for which records will be removed.
HELP	Displays a description of the command line options and their format.
MESSAGE_LEVEL	Level of messages that will be written.
MOUNT_POINT	HFS directory in which the HFS databases allocated to ddnames UNVDB and UNVSPOOL are mounted.
UCMD_SPOOL_DIR	Directory location in which the Universal Command Server Component database is located.
UEM_SERIALNO	Serial number of the Universal Event Monitor spool database record to remove.
VERSION	Writes the program version and copyright information.

BROKER_SPOOL_DIR - USLRM configuration option

Description

The BROKER_SPOOL_DIR option specifies the directory in which the Universal Broker component database (**bcomponent.db**) is located.

If this option is not used to specify the directory, the directory is read from the Universal Broker configuration file.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-b <i>directory</i>			✓	✓	
Command Line, Long Form	-brokerspooldir <i>directory</i>			✓	✓	
URMVSE Parameter	n/a					

Values

directory is the directory in which the Universal Broker component database is located.

COMPONENT - USLRM configuration option

Description

The COMPONENT option specifies the ID of a component for which records will be removed from all databases.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-c cid</code>			✓	✓	✓
Command Line, Long Form	<code>-component cid</code>			✓	✓	✓
URMVSE Parameter	COMPONENT (<i>cid</i>)	✓				

Values

cid is the component ID.

HELP - USLRM configuration option

Description

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

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-h			✓	✓	✓
Command Line, Long Form	-help			✓	✓	✓
URMVSE Parameter	n/a					

Values

(There are no values for this option.)

MESSAGE_LEVEL - USLRM configuration option

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l <i>level</i>			✓	✓	✓
Command Line, Long Form	-level <i>level</i>			✓	✓	✓
URMVSE Parameter	LEVEL (* <i>level</i>)	✓				

Values

level indicates either of the following level of messages:

- **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

IBM i	info
UNIX	warn
Windows	warn
z/OS	info

MOUNT_POINT - USLRM configuration option

Description

The MOUNT_POINT option specifies the HFS directory in which the HFS database allocated to ddnames UNVDB and UNVSPool are mounted.

The actual mount points will be subdirectories named after the HFS data set names being mounted.

If the mount points do not exist, they are created by Universal Spool List.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-mount_point <i>dir</i>					✓
URMVSE Parameter	n/a					

Values

dir is the HFS directory in which the HFS databases are mounted.

Default is /tmp.

UCMD_SPOOL_DIR - USLRM configuration option

Description

The UCMD_SPOOL_DIR option specifies the directory in which the Universal Command Server component database (**scomponent.db**) is located.

If this option is not used to specify the directory, the directory of the Universal Broker component database is used (see the [BROKER_SPOOL_DIR](#) option).

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-u <i>dir</i>			✓	✓	
Command Line, Long Form	-ucmdspooldir <i>dir</i>			✓	✓	
URMVSE Parameter	n/a					

Values

dir is the directory of the Universal Command Server component database.

UEM_SERIALNO - USLRM configuration option

Description

The UEM_SERIALNO option specifies the serial number of the Universal Event Monitor spool database record to remove.

Use the [Universal Spool List](#) utility to generate a complete list of all UEM spool records.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-s <i>serno</i>			✓	✓	
Command Line, Long Form	-uem_serialno <i>serno</i>			✓	✓	
URMVSE Parameter	n/a					

Values

serno is the serial number of the database record to remove.

VERSION - USLRM configuration option

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-v			✓	✓	✓
Command Line, Long Form	-version			✓	✓	✓
URMVSE Parameter	VERSION(* <i>option</i>)	✓				

Values

There are no values for this option.

IBM i



Valid values for *option* are:

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

Universal Submit Job

- [Universal Submit Job](#)
- [Functions](#)
- [Detailed Information](#)

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

Universal Submit Job

The Universal Submit Job (USBMJOB) utility is a command for the IBM i environment that encapsulates the IBM Submit Job (SBMJOB) command.

USBMJOB builds on the functionality of SBJJOB by providing a job submission command that better suits the needs of a remote user issuing IBM i commands via Universal Command.

Note



Users never should call USBMJOB directly.

Functions

USBMJOB performs four main functions:

1. Submits commands in their own easily customized job environment.
2. Monitors submitted jobs to completion and sets a return code based on the end code and message severity codes of the submitted job.
3. Provides a facility for the remote handling of inquiry messages generated by the submitted job. Inquiry messages are received and replied to from the z/OS console via the Universal Write-to-Operator utility (available only for z/OS managers).
4. Returns spooled output generated by the submitted job on standard output. The joblog generated by the submitted job is written to standard error and, optionally, to a job log output queue.

Detailed Information

The following pages provide detailed information for Universal Submit Job:

- [Universal Submit Job Usage](#)
- [Universal Submit Job Configuration Options](#)
- [Universal Submit Job Command Line Syntax](#)
- [Universal Submit Job Remote Reply Facility](#)
- [Universal Submit Job Return Codes](#)

Universal Submit Job Usage

- [Usage](#)
 - [Universal Command Server Options Affecting USBMJOB](#)
 - [Output](#)

Usage

Universal Submit Job (USBMJOB) submits a user command in its own job. USBMJOB supports all of the command parameters offered by the SBMJOB command that make sense for an unscheduled job submitted from a batch environment. This enables the user to fully customize the job environment for the user command. Internally, the SBMJOB command is called to submit the user job.

USBMJOB remains active for the duration of the submitted job. USBMJOB continuously monitors the state of the submitted job at a user-defined polling interval. In addition to monitoring for job completion, USBMJOB can detect when the submitted job is waiting for a reply to an inquiry message.

USBMJOB provides the option for inquiry messages generated from the submitted job to be sent to a remote z/OS console. Replies received from the z/OS console are sent as reply messages to the corresponding inquiry message.

When the submitted job completes, USBMJOB writes the joblog for the submitted job to standard error and, optionally, to a job log output queue. The spooled output generated by the submitted job is written to standard output.

After USBMJOB has finished processing the submitted job, it completes by issuing an escape message to the external message queue. The escape message sets the return code for the USBMJOB command. The severity code of the escape message indicates the return code.

If the user job submitted by USBMJOB completes normally (end code < 20), the severity code for the USBMJOB escape message will be 0. If the user job submitted by USBMJOB completes abnormally (end code > 10), the severity code for the USBMJOB escape message will be set to the highest severity code generated by the submitted job.

By issuing an escape message with a severity code correlated with the submitted job's end code/highest severity code, USBMJOB allows the Universal Command Server to pick up the severity code. This, in turn, allows the USBMJOB command to propagate its return code to the Universal Command Server.

Two helper commands - supplied by Stonebranch, Inc. - are called internally by USBMJOB:

- Universal Job initializer (UJOBINIT)
- Universal Message Handler (UMSGHNDLR)

Universal Job initializer (UJOBINIT) is called from within the job submitted by USBMJOB. UJOBINIT performs initialization that allows USBMJOB to redirect the joblog of the submitted job and then issues the user command.

Universal Command Server Options Affecting USBMJOB

The Universal Command Server `JOBLOG_COPY_KEEP` configuration option controls the keeping a copy of the job log returned via standard output on the local iSeries system.

Output

In addition to joblog redirection, USBMJOB returns the jobs spooled output to the Universal Command Manager via standard output.

When the submitted job user name and the user profile name passed to the Universal Command Manager differ, USBMJOB requires *SPLCTL authority to retrieve the spooled output. USBMJOB receives this *SPLCTL authority from the **UNVUBR511** user profile.

If *SPLCTL special authority is removed from the **UNVUBR511** user profile, USBMJOB will fail if:

1. Security is set to DEFAULT via the UNVCONF(UCMDS) configuration file and the submitted job user name and the user profile name passed to the Universal Command Manager differ.

Example:

```
ucmd -c "usbmjob cmd(dsplib abc) user(abc)" -i rmtsys -u myuser -w mypwd
```

2. Security is set to NONE via the UNVCONF(UCMDS) configuration file and a user name is specified for the submitted job.

Example:

```
ucmd -c "usbmjob cmd(dsplib abc) user(abc)" -i rmtsys
```

Universal Submit Job Configuration Options

Universal Submit Job Configuration Options

The Universal Submit Job command performs operations specified by configuration options. The options have associated values that describe the actions to take.

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

Configuration Options Categories

The USBMJOB configuration options are separated into two categories:

1. [USBMJOB-Specific Configuration Options](#)
2. [SBMJOB Encapsulated Configuration Options](#)

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

USBMJOB-Specific Configuration Options

USBMJOB-Specific Configuration Options

The following table identifies the USBMJOB-specific configuration options, which control the way that the submitted job is monitored and administered. Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
COMMAND	Command that runs in the submitted batch job.
COPY_SPOOL_FILES	Specification for whether or not spooled output files generated by the submitted job be copied to standard output.
ENCRYPTED_COMMAND_FILE	Name of an encrypted command file.
JOB_LOG_LIBRARY	Library into which the job log will be placed.
JOB_STATUS_POLLING_INTERVAL	Number of seconds that USBMJOB will sleep between calls to check the status of the submitted job.
KEY	Encryption key used to encrypt the encrypted command file.
REMOTE_MESSAGE_PREFIX	Text string that prefixes any remote message sent by USBMJOB.
REMOTE_REFRESH_INTERVAL	Time that a remote reply message will remain on a remote console without being replied to before it times out.
REMOTE_REPLY_COMMAND_PATH	Path (including the executable name) to the remote message handler (uwto).
REMOTE_REPLY_HOST	Host name of the remote system on which the uwto command is executed.
REMOTE_REPLY_PORT	Port of the Universal Broker on the remote system on which the Universal WTO command is executed.
REMOTE_REPLY_USER_ID	User ID for the remote system where the uwto command resides.
REMOTE_REPLY_USER_PWD	Password for user on the remote system where uwto resides.
SPECIFY_PRINT_CONTROL_CHARS	Print control characters (if any) that are to replace the spooled file's internal print control characters.
TRACE	Specification for whether or not trace information will be written to standard error.
USE_REMOTE_REPLY_FACILITY	Specification for whether or not USBMJOB will use the remote reply facility.

COMMAND - USBMJOB configuration option

The COMMAND option specifies a command that runs in the submitted batch job.

The command can be a maximum of 3000 characters.

USBMJOB Parameter: CMD(command)

COPY_SPOOL_FILES - USBMJOB configuration option

The CPYSPLF option specifies whether or not spooled output files generated by the submitted job are copied to standard output.

USBMJOB Parameter: CPYSPLF({ *yes | *no })

- **yes* will cause spooled output files generated by the submitted job to be written to standard output.
- **no* will prevent spooled output files generated by the submitted job from being written to standard output.

ENCRYPTED_COMMAND_FILE - USBMJOB configuration option

The ENCRYPTED COMMAND FILE option specifies the name of an encrypted command file.

The Encrypted Command File option is used when the Remote Reply Facility is used. In order to issue inquiry commands remotely, Universal Submit Job must log on to the remote system. This is accomplished by internally issuing a command to the Universal Command Manager. Therefore, the Encrypted Command File for Universal Submit Job serves the same security purposes as the Encrypted Command File for Universal Command Manager. However, encrypted command files for Universal Submit Job should not contain commands. The command will be issued by Universal Submit Job.

Use the Universal Encrypt utility provided with Universal Command to encrypt a plain text command file. If a key was used to encrypt the file, the same key must be supplied using the KEY option.

Command files (encrypted or not) that contain sensitive data should be protected from unauthorized read access with file level security.

USBMJOB Parameter: ECMFILE(cmd_file) [ECMMBR(member)]

JOB_LOG_LIBRARY - USBMJOB configuration option

The JOB LOG LIBRARY option specifies a library into which the job log will be placed.

The job log will be sent to files **USJPnnnnnn** and **USJSnnnnnn** (nnnnnn is the job number):

- **USJPnnnnnn** contains the primary messages of the job log.
- **USJSnnnnnn** contains the secondary messages of the job log.

The member name for both primary and secondary joblog files is of the form **Cnnnnnn**, where **nnnnnn** is the job number of the job that USBMJOB is running under.

USBMJOB Parameter: JOBLOGLIB(library)

JOB_STATUS_POLLING_INTERVAL - USBMJOB configuration option

The JOB STATUS POLLING INTERVAL option specifies the number of seconds that USBMJOB will sleep between calls to check the status of the submitted job.

USBMJOB Parameter: POLL(seconds)

KEY - USBMJOB configuration option

The KEY option specifies the encryption key used to encrypt the encrypted command file specified by ENCRYPTED_COMMAND_FILE. If no encryption key is specified, a default key is used.

USBMJOB Parameter: KEY(key)

REMOTE_REFRESH_INTERVAL - USBMJOB configuration option

The REMOTE_REFRESH_INTERVAL option specifies a time interval (in seconds) that controls how long a remote reply message will remain on a remote console without being replied to before it will time out.

If the remote reply message times out, the message will be removed from the remote console. Universal Submit Job then will determine if the user job still is waiting for a reply. If it is, the remote reply message will be re-sent to the remote console.

USBMJOB Parameter: RMTREFRESH(seconds)

[Default is 0 (wait indefinitely for a reply).]

REMOTE_MESSAGE_PREFIX - USBMJOB configuration option

The REMOTE MESSAGE PREFIX option allows the user to specify a text string up to 12 characters in length that will prefix any remote messages sent by USBMJOB.

This prefix can make it easier to relate a remote message to its associated job.

USBMJOB Parameter: RMTMSGPRFX(prefix)

REMOTE_REPLY_COMMAND_PATH - USBMJOB configuration option

The REMOTE REPLY COMMAND PATH option specifies the path (including the executable name) to the remote message handler (**uwto**). (See [Universal Write-to-Operator](#) for information on the Universal WTO utility.)

USBMJOB Parameter: MSGCMDPATH(command_path)

REMOTE_REPLY_HOST - USBMJOB configuration option

The REMOTE_REPLY_HOST option specifies the host name of the Universal Broker on the remote system on which the Universal WTO command is executed. (See [Universal Write-to-Operator](#) for information on the Universal WTO utility.)

USBMJOB Parameter: RMTHOST(host)

The format of *host* is either:

- Host name (for example, *homer*)
- Numeric address (for example, *10.20.30.40*)

REMOTE_REPLY_PORT - USBMJOB configuration option

The REMOTE REPLY PORT option specifies the port of the Universal Broker on the remote system on which the Universal WTO command is executed (see [REMOTE_REPLY_HOST](#)). (See [Universal Write-to-Operator](#) for information on the Universal WTO utility.)

USBMJOB Parameter: RMTPORT(port)

Valid values for *port* are:

- Port number
- Service name

[Default is **7887**.]

REMOTE_REPLY_USER_ID - USBMJOB configuration option

The REMOTE REPLY USER ID option specifies the user ID for the remote system where the Universal WTO command resides. (See [Universal Write-to-Operator](#) for information on the Universal WTO utility.)

USBMJOB Parameter: RMTUSER(userid)

REMOTE_REPLY_USER_PWD - USBMJOB configuration option

The REMOTE REPLY USER PASSWORD option specifies the password for the user on the remote system where the Universal WTO command resides (see [Universal Write-to-Operator](#) utility).

USBMJOB Parameter: RMPWD(password)

SPECIFY_PRINT_CONTROL_CHARS - USBMJOB configuration option

Specifies which print control characters (if any) are to replace the spooled file's internal print control characters.

USBMJOB Parameter: SPLFCTLCHR(*characters)

Valid values for *characters* are:

- *NONE*
- *FCFC*
- *PRTCTL*
- *S36FMT*

See the CL Reference, SC41-5722 for more information on these values.

TRACE - USBMJOB configuration option

The TRACE option specifies whether or not trace information will be written to standard error.

USBMJOB Parameter: TRACE({*yes|*no})

- **yes* will cause trace information to be written to standard error.
- **no* will prevent trace information from being written to standard error.

Use TRACE only as directed by Stonebranch, Inc. Customer Support.

USE_REMOTE_REPLY_FACILITY - USBMJOB configuration option

The USER REMOTE REPLY FACILITY option specifies whether or not Universal Submit Job will use the remote reply facility.

The remote reply facility will detect messages, issued by the submitted job, that require a reply. The message then will be passed on to a remote system for a reply. When the reply is received, the reply will be sent to the message queue that is waiting for the reply.

USBMJOB Parameter: RMTRPY({ *yes | *no })

- *yes will cause USBMJOB to use the remote reply facility.
- *no will cause USBMJOB to ignore message wait conditions for the submitted job.

[Default is *no.]

SBMJOB Encapsulated Configuration Options

The SBJOB encapsulated configuration options, shown in the table below, have a one-to-one relationship with the IBM SBJOB command parameters of the same name.

Option Name	USBJOB Parameter
ALLOW DISPLAY BY WRKSBMJOB	DSPSBMJOB({ *yes *no })
CODED CHARACTER SET ID	CCSID({ *current *sysval *usrprf *hex coded_character_set_identifier })
COPY ENVIRONMENT VARIABLES	CPYENVVAR({ *yes *no })
COUNTRY ID	CNTRYID({ *current *sysval *usrprf country_id })
CURRENT LIBRARY	CURLIB({ *current *usrprf *crt dft current_library_name })
HOLD ON JOB QUEUE	HOLD({ *jobd *no *yes })
INITIAL LIBRARY LIST	INLLIBL({ *current *jobd *sysval *none library_name... })
INLASPGRP	INLASPGRP({ *current *jobd *none })
INQUIRY MESSAGE REPLY	INQMSGRPY({ *jobd *rqd *dft *sysrpyl })
JOB DATE	DATE({ *jobd *sysval job_date })
JOB DESCRIPTION	JOB({ *usrprf [library/] job_description })
JOB MESSAGE QUEUE FULL ACTION	JOBMSGQFL({ *jobd *sysval *nowrap *wrap *prtwrap })
JOB MESSAGE QUEUE MAXIMUM SIZE	JOBMSGQMX({ *jobd *sysval maximum_size_of_job_message_queue })
JOB NAME	JOB({ *jobd job_name })
JOB PRIORITY	JOBPTY(priority)
JOB QUEUE	JOBQ({ *jobd [library/] job_queue })
JOB SWITCHES	SWS({ *jobd switch_settings })
LANGUAGE ID	LANGID({ *current *sysval *usrprf language_id })
LOG CL PROGRAM COMMANDS	LOGCLPGM({ *jobd *no *yes })
LOGOUTPUT	LOGOUTPUT({ *jobd *sysval *joblogsvr *jobend *pnd })
OUTPUT PRIORITY	OUTPTY(priority)
OUTPUT QUEUE	OUTQ({ *current *usrprf *dev *jobd [library/] output_queue })
PRINT DEVICE	PRTDEV({ *current *usrprf *sysval *jobd printer_device_name })
PRINT TEXT	PRTTXT(text)
SORT SEQUENCE	SRTSEQ({ *current *sysval *usrprf *hex *langidunq *langidshr* [{ *libl *curlib library_name }] table_name })
SUBMITTED FOR	SBMFOR(job_number / user / job_name)
SYSTEM LIBRARY LIST	SYSLIBL({ *current *sysval })
USER	USER({ *current *jobd user_name })

Universal Submit Job Command Line Syntax

- [Command Line Syntax](#)
- [Command Line Syntax Rules](#)

Command Line Syntax

The following figure illustrates the command line syntax of Universal Submit Job.

The command line name of every USBMJOB-specific configuration option is a link to detailed information about that option.

```

USBMJOB
[CMD(command)]
[JOBLOGLIB(library)]
[POLL(seconds)]
[RMTRPY( { *yes|*no } )]
[RMTRFRESH(seconds)]
[RMTMSGPRX(prefix)]
[RMTUSER(userid)]
[RMPWD(password)]
[ECMFILE(cmd_file) [ECMMBR(member)] [KEY(key)] ]
[RMTHOST(host)]
[RMTPORT (port)]
[MSGCMDPATH(path)]
[JOB( { *jobd | job_name } )]
[JOB( { *usrprf | [library/ ] job_description } )]
[JOBMSGQFL( { *jobd|*sysval|*nowrap|*wrap|*prtwrap } )]
[JOBQ( { *jobd | [ library/ ] job_queue } )]
[JOBPTY(priority)]
[OUTPTY(priority)]
[PRTDEV( { *current | *usrprf | *sysval | *jobd | printer_device_name } )]
[OUTQ( { *current | *usrprf | *dev | *jobd | [ library/ ] output_queue } )]
[CPYSPLF( { *yes|*no } )]
[SPLFCTLCHR ( { *none|*fcfc|*prtctl|*s36fmt } )]
[TRACE( { *yes|*no } )]
[USER( { *current | *jobd | user_name } )]
[PRTTXT(text)]
[SYSLIB( { *current|*sysval } )]
[CURLIB( { *current | *usrprf | *crt dft | current_library_name } )]
[INLLIBL( { *current | *jobd | *sysval | *none |library_name... } )]
[LOGCLPGM( { *jobd|*no|*yes } )]
[INQMSGRPY( { *jobd|*rqd|*dft|*sysrpyl } )]
[INLASPRP( { *current | *jobd | *none } )]
[HOLD( { *jobd|*no|*yes } )]
[DATE( { *jobd|*sysval|job_date } )]
[SWS( { *jobd|switch_settings } )]
[DSPSBMJOB( { *yes|*no } )]
[SRTSEQ( { *current|*sysval|*usrprf|*hex|*langidung|*langidshr | [ *libl | *curlib | library/ ] table_name } )]
[LANGID( { *current|*usrprf|*sysval|language_id } )]
[CNTRYID( { *current|*usrprf|*sysval|country_id } )]
[CCSID( { *current|*usrprf|*sysval|*hex|coded_character_set_identifier } )]
[SBMFOR(job_number/user/job_name)]
[JOBMSGQMX( { *jobd|*sysval|*maximum_size_of_job_message_queue } )]
[CPYENVVAR( { *yes|*no } )]

```

Command Line Syntax Rules

Values for configuration options that contain special characters require:

- Double (") quotation marks when executed from an MVS Universal Command Manager.
- Single (') quotation marks when executed from an IBM i Universal Command Manager.

For example, the following is correct when executed from a z/OS Universal Command Manager:

```
MSGCMDPATH("/usr/local/universal/bin/uwto")
```

However, the following is incorrect when executed from a z/OS Universal Command Manager; it will create a syntax error:

```
MSGCMDPATH('/usr/local/universal/bin/uwto')
```

Universal Submit Job Remote Reply Facility

Remote Reply Facility

Universal Submit Job can detect when messages sent by the submitted job require a reply. If the Remote Reply Facility is turned on (RMTRPY(*yes)), USBMJOB will send all messages requiring a reply to a remote z/OS console. Replies to the inquiry messages are received from the z/OS console and sent to the IBM i message queue waiting for the reply.

The Remote Reply Facility requires an installation of Universal Command on the IBM i system where the USBMJOB command will run and an installation of Universal Command on the remote z/OS system where the inquiry messages will be sent to for reply.

- Universal Command on the IBM i must be at Universal Command 1.2.1 level 7 or greater.
- Universal Command on the remote z/OS system, where the inquiry messages will be sent, must be at level 12 or greater.

Remote Reply Facility Utilities

The Remote Reply Facility used by USBMJOB is comprised of a group of Stonebranch Inc. utilities that work together, as shown in the following table.

Utility	Platform
Universal Submit Job	IBM i
Universal Message Handler	IBM i
Universal Command	IBM i and z/OS
Universal UWTO	z/OS Unix System Services (USS)

Universal Submit Job and Universal Message Handler are part of the Universal Command for IBM i licensed product. These utilities were added to the product in Universal Command 1.2.1 level 7).

Universal WTO (UWTO) is a command line utility for the z/OS Unix System Services (USS) environment, as of level 12. The path to the UWTO command is specified on the **MSGCMDPATH** parameter of the USBMJOB command.

See [Universal Write-to-Operator](#) for more information on the UWTO command.

Universal Submit Job Return Codes

Return Codes

The Universal Submit Job command completes by sending an escape message to the external message queue. The severity code of this escape message is the USBMJOB return code.

The USBMJOB return code is determined in the following way:

When the submitted job completes, USBMJOB scans the job log, examining the severity code of all *ESCAPE, *NOTIFY, *STATUS, and function check messages.

- If the submitted job completed abnormally (end code > 10), the USBMJOB return code will be set to the highest severity code examined in the submitted job's job log.
- If the submitted job completes normally (end code < 20), the examined severity codes will be ignored and the USBMJOB return code will be set to 0.

Setting the return code with an escape message allows the Universal Command Server to pick up the return code for use in its return code processing.

The range of possible severity code values is 0 through 99.

If USBMJOB encounters a processing error, a severity code of 99 will be used, regardless of severity codes that may have been examined from the submitted job.

Universal Write-to-Operator

- [Universal Write-to-Operator](#)
- [Return Codes](#)
- [Command Line Syntax](#)
- [Detailed Information](#)
- [Universal Write-to-Operator Examples](#)

Universal Write-to-Operator

The Universal WTO (UWTO) utility is a command line utility for the z/OS UNIX System Services (USS) environment.

Universal WTO issues two types of messages to z/OS consoles:

1. Write-To-Operator (WTO) messages
2. Write-To-Operator-with-Reply (WTOR) messages

Universal WTO either:

- Writes a message to the z/OS console, and then ends (WTO).
- Writes a message to the z/OS console and waits for a requested reply (WTOR).

The type of message to be written (WTO or WTOR) is specified via the REPLY option. If WTOR is specified, the message is written to the console as a WTOR message and Universal WTO waits for a reply. The message reply is written to stdout.

Return Codes

The UWTO command ends with specific return codes indicating the success of the requested action.

The following table describes these return codes.

Return Code	Description
0	Process was successful.
1	Message was written to the console, but a warning was issued regarding a requested option. A detailed message is written to standard error.
2	WTOR request timed out waiting for an operator reply.
3	Error occurred when attempting to write the message. No message was written to the console.
10	Error in the command line options was detected. No processing occurred.

Command Line Syntax

The following figure illustrates the syntax - using the long form of command line options - of Universal WTO.

```

uwto
[

-msg

  message]
[

-reply

  {yes|no} [-timeout seconds] ]
[

-consoleid

  id | -consolename name ]
[

-level

  {trace|audit|info|warn|error} ]
[

-key

  keyname]
uwto
{

-help

  |

-version

}

```

Detailed Information

The following pages provide additional detailed information for Universal Write-to-Operator:

- [Universal Write-to-Operator Configuration Options](#)

Universal Write-to-Operator Examples

For examples of how to use Universal Write-to-Operator, see [Creating Write-to-Operator Messages](#).

Universal Write-to-Operator Configuration Options

Universal Write-to-Operator Configuration Options

The following table identifies the configuration options available for use with Universal Write-to-Operator (Universal WTO). Each **Option Name** is a link to detailed information about that option.

Option Name	Description
CONSOLE_ID	ID of the console to which the message is routed.
CONSOLE_NAME	Name of the console to which the message is routed.
HELP	Displays a description of the command line options and their format.
KEY	Key to associate with the message.
MESSAGE	Text to write to the z/OS operator console.
MESSAGE_LEVEL	Level of messages to write.
REPLY	Directs UWTO to issue a WTOR message and wait for an operator reply to the message.
TIMEOUT	Number of seconds to wait for a WTOR operator reply.
VERSION	Writes the program version and copyright information.

CONSOLE_ID - UWTO configuration option

Description

The CONSOLE_ID option specifies the console ID which the message is routed.

If CONSOLE_ID specifies an invalid console ID, the message is written to the default consoles. Universal WTO will write a warning message to standard error and end with exit code 1.

CONSOLE_ID and [CONSOLE_NAME](#) are mutually exclusive.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-consoleid <i>id</i>					✓
Environment Variable	UWTOCONSOLEID= <i>id</i>					✓

Values

id is the ID of the console to which the messages is routed.

Valid values for *id* are numeric values.

CONSOLE_NAME - UWTO configuration option

Description

The CONSOLE_NAME option specifies the console name which the message is routed.

It provides a method to route messages based on console name.

If an invalid console name is specified, the message is written to the default consoles. Universal WTO will write a warning message to standard error and end with exit code 1.

CONSOLE_NAME and [CONSOLE_ID](#) are mutually exclusive.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-consolename <i>name</i>					✓
Environment Variable	UWTOCONSOLENAME= <i>name</i>					✓

Values

name specifies the name of the console to which the message is routed.

The format of *name* is 1 to 8 characters.

HELP - UWTO configuration option

Description

The HELP option displays a description of the Universal Write-to-Operator command line options and their format.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-h					✓
Command Line, Long Form	-help					✓
Environment Variable	n/a					

Values

There are no values for this option.

KEY - UWTO configuration option

Description

The KEY option specifies a key to associate with the message.

The key provides a way to identify messages to operators. The z/OS DISPLAY console commands can list messages key values.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-key <i>keyname</i>					
Environment Variable	UWTOKEY= <i>keyname</i>					✓

Values

keyname is the message key.

The format of *keyname* is 1 to 8 characters.

MESSAGE - UWTO configuration option

Description

The MESSAGE option specifies the text to write to the z/OS operator console.

The text is written as a WTO or WTOR message, as specified by the [REPLY](#) option.

Note



Even though Universal WTO executes in the z/OS Unix System Services environment, not all USS supported characters are supported in the console character set. Refer to IBM MCS Console documentation for character set support. Unsupported characters are translated to blanks when written to the console.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-m message</code>					✓
Command Line, Long Form	<code>-msg message</code>					✓
Environment Variable	<code>UWTOMSG=message</code>					✓

Values

message is the text message to write to the z/OS operator console.

The maximum length of *message* depends on the type of message specified by REPLY:

- WTO: 770 characters
- WTOR: 121 characters

Note



A WTO message that exceeds 125 characters is written as a multi-line WTO.

MESSAGE_LEVEL - UWTO configuration option

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l <i>level</i>					✓
Command Line, Long Form	-level <i>level</i>					✓
Environment Variable	UWTOLEVEL= <i>level</i>					✓

Values

level indicates either of the following level of messages:

- **trace**
Writes traces messages used for debugging. The trace file, named **uwto.trc**, is created in the working directory of Universal WTO.

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 warn.

REPLY - UWTO configuration option

Description

The REPLY option specifies the type of message to be issued by Universal WTO:

- WTO (do not request message reply)
- WTOR (request and wait for message reply)

For a WTOR message, the length of time to wait for a reply can be limited with the [TIMEOUT](#) option. The maximum reply length is 119 characters. The reply is written to Universal WTO's standard output file.

Note



A valid operator reply to a WTOR message can be zero characters. In this case, nothing is written to standard output.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-reply <i>option</i>					✓
Environment Variable	UWTREPLY= <i>option</i>					✓

Values

option specifies the type of message to be issued.

Valid values for option are:

- **no**
Issue a WTO message.
- **yes**
Issues a WTOR message.

Default is no.

TIMEOUT - UWTO configuration option

Description

The TIMEOUT option specifies the number of seconds to wait for a reply to a WTOR message (see the [REPLY](#) option).

If a reply is not received within the specified time, the WTOR message is deleted and Universal WTO ends with exit code 2.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-timeout <i>seconds</i>					✓
Environment Variable	UWTOTIMEOUT= <i>seconds</i>					✓

Values

seconds is the length of time to wait for a reply.

Default is 0 seconds (wait indefinitely for a reply).

VERSION - UWTO configuration option

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-v			✓	✓	✓
Command Line, Long Form	-version			✓	✓	✓
Environment Variable	n/a					

Values

There are no values for this option.

Universal Agent Utilities - Additional Information

Universal Agent Utilities - Additional Information

The following table identifies and provides links to additional information related to Universal Agent Utilities.

Information	Description
SSL/TLS Cipher Suites	SSL/TLS cipher suites for use with Universal Command.
Character Code Pages	Character code pages provided by Stonebranch Inc. for use with Universal Agent on each supported operating system.
UTT Files	Universal Translate Table (UTT) files are used to translate between Unicode and the local single-byte code page.

Character Code Pages - UTIL


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 - UTIL

The following table identifies all SSL/TLS cipher suites provided by Stonebranch, Inc. for use with Universal Control.

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 and 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 and SHA-1 message digest.
RC4-SHA	128-bit RC4 encryption and SHA-1 message digest.
RC4-MD5	128-bit RC4 encryption and MD5 message digest.
DES-CBC3-SHA	128-bit Triple-DES encryption and SHA-1 message digest.
DES-CBC-SHA	<p>128-bit DES encryption with SHA-1 message digest.</p> <p>Note  As of Universal Agent 6.7.0.0, DES-CBC-SHA is supported only on HP-UX.</p> <p>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.</p>

UTT Files - UTIL

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 . <i>codepage</i> is the member name of the UTT file.
z/OS	UTT files are located in the library allocated to the UNVNLS ddname. <i>codepage</i> is the member name of the UTT file.
UNIX	UTT files are located in the nls subdirectory of the installation directory. <i>codepage</i> 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. <i>codepage</i> 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. <i>codepage</i> is the base file name of the UTT file.