Universal Broker 5.1.0 Reference Guide

Universal Broker

Universal Broker 5.1.0 Reference Guide

Universal Broker Overview

Universal Broker

Universal Broker manages Workload Automation 5 components.

This document provides operating system-specific detailed technical information for Universal Broker:

- Started task (z/OS)
- Configuration Options
- Component Definition options
- Universal Access Control List entries

Universal Broker Utilization

For information how Universal Broker is utilized, see the Indesca 5.1.0 User Guide or Infitran 5.1.0 User Guide.

Universal Broker for zOS

Started Task

The following figure illustrates the JCL procedure for the Universal Broker started task. **UBROKER** is the member name of this JCL procedure in the Workload Automation sample library (**SUNVSAMP**).

```
//UBROKER  PROC HLQ=#SHLQ.UNV,
//             DBHLQ=#PHLQ.UNV,
//             PHLQ=#PHLQ.UNV,
//             SAPRFC=USPRFC00,
//             RGN=50M,
//             UPARM=,
//             LEPARM=
//*
//S1       EXEC PGM=UBROKER,REGION=&RGN,
//             PARM='ENVAR(TZ=EST5EDT) &LEPARM/&UPARM'
//STEPLIB  DD  DSN=&HLQ..SUNVLOAD,
//             DISP=SHR
//UNVCONF  DD  DSN=&PHLQ..UNVCONF,
//             DISP=SHR
//UNVCOMP  DD  DSN=&PHLQ..UNVCOMP,
//             DISP=SHR
//UNVRFC   DD  DSN=&PHLQ..UNVCONF(&SAPRFC),
//             DISP=SHR
//UNVNLS   DD  DSN=&HLQ..SUNVNLS,
//             DISP=SHR
//UNVTMPL  DD  DSN=&HLQ..SUNVTMPL,
//             DISP=SHR
//UNVCREF  DD  DSN=&PHLQ..UNVCREF,
//             DISP=SHR
//UNVDB    DD  DSN=&DBHLQ..UNVDB,
//             DISP=SHR
//UNVSPOOL DD  DSN=&DBHLQ..UNVSPOOL,
//             DISP=SHR
//UNVTRACE DD  DSN=&PHLQ..UNVTRACE,
//             DISP=SHR
//UNVTRMDL DD  DSN=&PHLQ..MDL,
//             DISP=SHR
//UNVLOG   DD  SYSOUT=*,HOLD=YES
//SYSPRINT DD  SYSOUT=*,HOLD=YES    -- standard output
//SYSOUT   DD  SYSOUT=*,HOLD=YES    -- standard error
//CEEDUMP  DD  SYSOUT=*,HOLD=YES    -- LE dumps
//SYSUDUMP DD  SYSOUT=*,HOLD=YES    -- system dumps
//SYSIN    DD  DUMMY                -- standard input
```

DD Statements used in JCL Procedure

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

ddname	DCB Attributes	Mode	Description
STEPLIB	DSORG=PO, RECFM=U	input	Workload Automation load library containing the program

			being executed.
UNVCONF	DSORG=PO, RECFM=(F, FB, V, VB)	input	Configuration members for all Workload Automation components.
UNVCOMP	DSORG=PO, RECFM=(F, FB, V, VB)	input	Universal Broker component definition PDS.
UNVRFC	DSORG=PS, RECFM=(F, FB, V, VB)	input	SAP RFC file used by Universal Connector.
UNVNLS	DSORG=PO, RECFM=(F, FB, V, VB)	input	Workload Automation national language support library. Contains message catalogs and code page translation tables.
UNVTMPL	DSORG=PO, RECFM=(V, VB)	input	Workload Automation configuration template library.
UNVCREF	DSORG=PO, RECFM=(F, FB, V, VB)	input	Universal Command Server command reference PDS.
UNVDB	DSNTYPE=HFS	input, output	Universal Broker database. **Note** This ddname is not used if zFS data sets are used instead of HFS data sets.
UNVSPOOL	DSNTYPE=HFS	input, output	Workload Automation spool database. **Note** This ddname is not used if zFS data sets are used instead of HFS data sets.
UNVTRACE	DSORG=PO, RECFM=(F, FB, V, VB), LRECL=256 or above.	output	Workload Automation trace PDS. This ddname is used only if UNVTRMDL is not defined.
UNVTRMDL	DSORG=PS, RECFM=(F,FB,V,VB), LRECL=256 or above.	output	Workload Automation trace model data set. The data set name is used as the high-level qualifier of the dynamically allocated trace data sets.
UNVLOG	DSORG=PS, RECFM=(F,FB,V,VB), LRECL=256 or above.	output	Universal Broker message destination ddname when option MESSAGE_DESTINATION value is LOGFILE.
SYSPRINT	DSORG=PS, RECFM=(F, FB, V, VB)	output	Standard output file for the UBROKER program.

SYSOUT	DSORG=PS, RECFM=(F, FB, V, VB)	output	Standard error file for the UBROKER program.
SYSIN	DSORG=PS, RECFM=(F, FB, V, VB)	input	Standard input file for the UBROKER program.

Configuration

Universal Broker reads configuration options only from the Universal Broker configuration file, which is allocated to ddname **UNVCONF**.

Configuration Options

The following table identifies all of the Universal Broker 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.
CA_CERTIFICATES	Path to PEM formatted trusted CA X.509 certificates.
CERTIFICATE	Path to Broker’s PEM formatted X.509 certificate.
CERTIFICATE_REVOCATION_LIST	Path to PEM formatted CRL.
CODE_PAGE	Text translation code page.
CTL_SSL_CIPHER_LIST	SSL cipher list for the control sessions.
DNS_CACHE_TIMEOUT	Time-out for DNS cache.
EVENT_GENERATION	Events to be generated as persistent event records.
MESSAGE_DESTINATION	Location where messages are written.
MESSAGE_LANGUAGE	Language of written messages.
MESSAGE_LEVEL	Level of messages written.
MONITOR_EVENT_EXPIRATION	Duration of a monitoring event record in the Universal Broker local UES database.
MOUNT_POINT	HFS or zFS database mount directory.

MOUNT_POINT_MODE	HFS or zFS permission mode for MOUNT_POINT.
PERSISTENT_EVENT_EXPIRATION	Duration of a persistent event record in the Universal Broker local UES database.
PRIVATE_KEY	Path to Broker’s PEM formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Broker’s PRIVATE_KEY.
RUNNING_MAX	Maximum number of simultaneous components.
SAF_KEY_RING	SAF certificate key ring name.
SAF_KEY_RING_LABEL	SAF certificate key ring label.
SERVICE_BACKLOG	Service interface backlog size for pending connection requests.
SERVICE_IP_ADDRESS	TCP/IP address on which the Broker listens.
SERVICE_PORT	TCP/IP port number on which the Broker listens.
SMF_EXIT_LOAD_LIBRARY	UNVACTRT SMF exit load library.
SSL_IMPLEMENTATION	SSL implementation.
SYSTEM_ID	Broker running on a system (O/S image).
TMP_DIRECTORY	z/OS UNIX directory name for temporary files.
TRACE_FILE_LINES	Maximum number of lines written to the trace file.
TRACE_TABLE	Memory trace table specification.
UCMD_STC_SUPPORT	Support for Universal Command started tasks.
UNIX_DB_DATA_SET	HFS or zFS data set used for the Universal Broker's databases.
UNIX_SPOOL_DATA_SET	HFS or zFS data set used for the Universal Broker's spool.

Component Management

Universal Broker is aware only of Workload Automation components that have been defined. It is the responsibility of Universal Broker to start, stop, and query these defined components.

One of the steps in the installation of a component is defining it to the local Universal Broker. These component definitions provide Universal Broker with the necessary information that it needs to manage the components.

Component Definitions

Component definitions are text files that define Workload Automation components to the Universal Broker. All z/OS component definition files are located in the Universal Broker component definition library **UNVCOMP** allocated to the **UNVCOMP** ddname.

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

The following table identifies all of the options that comprise Workload Automation for z/OS component definitions. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AUTOMATICALLY_START	Specification for whether the component automatically starts by the Universal Broker at start-up time or only on demand.
COMPONENT_NAME	Name by which clients know the component.
COMPONENT_TYPE	Type of component.
CONFIGURATION_FILE *	Component's configuration file name.
RESTART	Specification for whether or not the component should be restarted if it ends.
RESTART_CONDITIONS	Exit conditions criteria for which the server is considered eligible for restart.
RESTART_DELAY	Number of seconds to wait before restarting.
RESTART_MAX_FREQUENCY	Maximum frequency a server can be restarted.
RUNNING_MAXIMUM	Maximum number of this component that can run simultaneously.
START_COMMAND *	Component program member name.
WORKING_DIRECTORY *	Path used as the working directory of the component.

* These options are required in the component definitions.

Universal Access Control List

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

The Universal Broker reads in the UACL entries when the program is started. If the UACL file is changed, the new entries can be activated by recycling the Broker or by sending the Universal Broker a Universal Control REFRESH command that will instruct the Universal Broker to reread all its configuration files including the UACL file.

UACL Entries

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

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

UACL Entry Name	Description
UBROKER_ACCESS	Allows or denies access to Universal Broker services.
CERT_MAP	Maps a client X.509 certificate to a certificate identifier.
EVENT_ACCESS	Controls which Universal Enterprise Controller has read and delete access to the Universal Event Subsystem event data maintained by the Universal Broker.
REMOTE_CONFIG_ACCESS	Authorizes update access to the product configuration files and setting of the configuration managed mode of the Broker.

Universal Broker for Windows

Configuration

Universal Broker reads configuration options only from the Universal Broker configuration file.

The Universal Broker configuration file is named **ubroker.conf**. This file can be edited manually with any text editor.

Configuration Options

The following table identifies all of the Universal Broker 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.
CA_CERTIFICATES	Path to PEM formatted trusted CA X.509 certificates.
CERTIFICATE	Path to Broker's PEM formatted X.509 certificate.
CERTIFICATE_REVOCATION_LIST	Path to PEM formatted CRL.
CODE_PAGE	Text translation code page.
COMPONENT_PORT	TCP/IP port used for Broker-Component communications.
CTL_SSL_CIPHER_LIST	SSL cipher list for the control sessions.
DNS_CACHE_TIMEOUT	Time-out for DNS cache.
EVENT_GENERATION	Events to be generated as persistent event records.
INSTALLATION_DIRECTORY	Base directory where product is installed.
LOG_DIRECTORY	Directory where log files are created.

MESSAGE_DESTINATION	Location where messages are written.
MESSAGE_LANGUAGE	Language of written messages.
MESSAGE_LEVEL	Level of messages written.
MONITOR_EVENT_EXPIRATION	Duration of a monitoring event record in the Universal Broker local UES database.
NLS_DIRECTORY	Location of UMC and UTT files.
PERSISTENT_EVENT_EXPIRATION	Duration of a persistent event record in the Universal Broker local UES database.
PRIVATE_KEY	Path to Broker’s PEM formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Broker’s PRIVATE_KEY.
RUNNING_MAX	Maximum number of simultaneous components.
SERVICE_BACKLOG	Service interface backlog size for pending connection requests.
SERVICE_IP_ADDRESS	TCP/IP address on which the Broker listens.
SERVICE_PORT	TCP/IP port number on which the Broker listens.
SPOOL_DIRECTORY	Spool file directory.
TMP_DIRECTORY	Temporary file directory.
TRACE_DIRECTORY	Trace file directory.
TRACE_FILE_LINES	Maximum number of lines written to the trace file.
TRACE_TABLE	Memory trace table specification.
WORKING_DIRECTORY	Broker's working directory.

Component Management

Universal Broker is aware only of Workload Automation components that have been defined to it. It is the responsibility of Universal Broker to start, stop, and query these defined components.

One of the steps in the installation of a component is defining it to the local Universal Broker. These component definitions provide Universal Broker with the necessary information that it needs to manage the components.

Component Definitions

Component definitions are text files that define Workload Automation components to the Universal Broker.

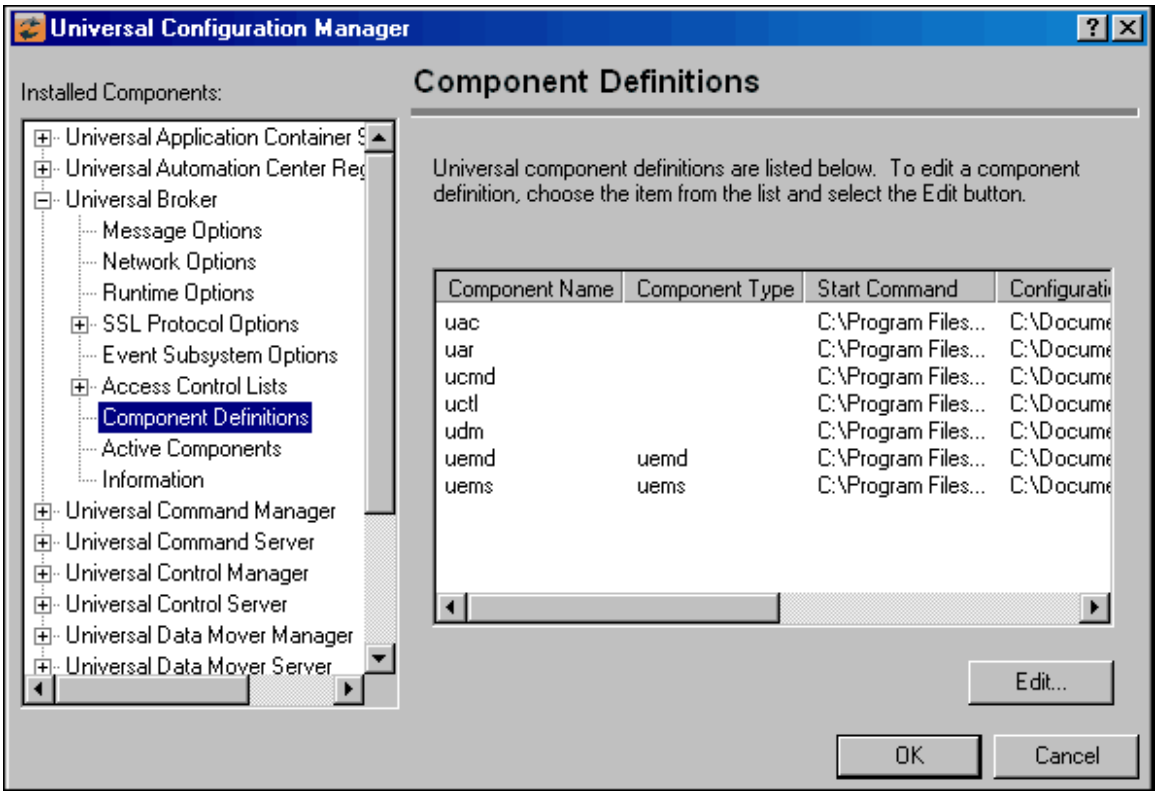
Component definition files reside in **`%ALLUSERSPROFILE%\Application Data\Universal\comp`**, where **`%ALLUSERSPROFILE%`** is an environment variable that resolves by default to:

- **`C:\Documents and Settings\All Users`** on Windows 2000/XP/Server 2003
- **`C:\ProgramData`** on Windows Vista/Server 2008

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

Although component definition files can be edited with any text editor (for example, Notepad), the Universal Configuration Manager application is the recommended way to edit component definitions for Windows.

> **Note**
> The component definitions for all Workload Automation are identified in the Component Definitions property page of the Universal Broker, as shown below.



The following identifies all of the options that comprise Workload Automation for Windows component definitions. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AUTOMATICALLY_START	Specification for whether the component automatically starts by the Universal Broker at start-up time or only on demand.

COMPONENT_NAME	Name by which clients know the component.
COMPONENT_TYPE	Type of component.
CONFIGURATION_FILE *	Component's configuration file name.
RESTART	Specification for whether or not the component should be restarted if it ends.
RESTART_CONDITIONS	Exit conditions criteria for which the server is considered eligible for restart.
RESTART_DELAY	Number of seconds to wait before restarting.
RESTART_MAX_FREQUENCY	Maximum frequency a server can be restarted.
RUNNING_MAXIMUM	Maximum number of this component that can run simultaneously.
START_COMMAND *	Command that starts the component.
WORKING_DIRECTORY *	Path used as the working directory of the component.

* These options are required in the component definitions.

Universal Access Control List

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

Universal Broker reads the UACL entries when the program is started. If the UACL file is changed, the new entries can be activated either by:

- Stopping and starting Universal Broker.
- Sending Universal Broker a Universal Control REFRESH command, which instructs Universal Broker to reread all of its configuration files, including the UACL file.

Note
Although the UACL file, like all 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 change UACL entries.

Via this method, a REFRESH command is sent to Universal Broker, and any new entries take effect immediately. There is no need to stop and restart the Broker in order for the changes to be applied.

UACL Entries

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

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

UACL Entry Name	Description

UBROKER_ACCESS	Allows or denies access to Universal Broker services
CERT_MAP	Maps a client X.509 certificate to a certificate identifier.
EVENT_ACCESS	Controls which Universal Enterprise Controller has read and delete access to the Universal Event Subsystem event data maintained by the Universal Broker.
REMOTE_CONFIG_ACCESS	Authorizes update access to the product configuration files and setting of the configuration managed mode of the Broker.

Universal Broker for UNIX

Configuration

Universal Broker reads configuration options only from the Universal Broker configuration file.

The Universal Broker configuration file is named **ubroker.conf**. This file can be edited manually with any text editor.

Configuration Options

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

Option Name	Description
ACTIVITY_MONITORING	Specification for generation of product activity monitoring events.
BIF_DIRECTORY	Broker Interface Directory where Universal Broker will create its broker interface file.
CA_CERTIFICATES	Path to PEM formatted trusted CA X.509 certificates.
CERTIFICATE	Path to Broker's PEM formatted X.509 certificate.
CERTIFICATE_REVOCATION_LIST	Path to PEM formatted CRL.
CODE_PAGE	Text translation code page.
COMPONENT_DIRECTORY	Component definition file directory.
CTL_SSL_CIPHER_LIST	SSL cipher list for the control sessions.
DNS_CACHE_TIMEOUT	Time-out for DNS cache.
EVENT_GENERATION	Events to be generated as persistent event records.
INSTALLATION_DIRECTORY	Base directory where product is installed.

LOG_DIRECTORY	Log file directory.
LOG_FILE_GENERATIONS	Total number of log files that will be saved within the log directory.
LOG_FILE_LINES	Total number of lines to be written to the log file before the log file is wrapped.
MESSAGE_DESTINATION	Location where messages are written.
MESSAGE_LANGUAGE	Language of written messages.
MESSAGE_LEVEL	Level of messages written.
MONITOR_EVENT_EXPIRATION	Duration of a monitoring event record in the Universal Broker local UES database.
NLS_DIRECTORY	UMC and UTT file directory.
PERSISTENT_EVENT_EXPIRATION	Duration of a persistent event record in the Universal Broker local UES database.
PID_FILE_DIRECTORY	PID file location.
PRIVATE_KEY	Path to Broker’s PEM formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Broker’s PRIVATE_KEY.
RUNNING_MAX	Maximum number of simultaneous components.
SERVICE_BACKLOG	Service interface backlog size for pending connection requests.
SERVICE_IP_ADDRESS	TCP/IP address on which the Broker listens.
SERVICE_PORT	TCP/IP port number on which the Broker listens.
SPOOL_DIRECTORY	Spool file directory.
TMP_DIRECTORY	Temporary file directory.
TRACE_DIRECTORY	Trace file directory.
TRACE_FILE_LINES	Maximum number of lines written to the trace file.

TRACE_TABLE	Memory trace table specification.
WORKING_DIRECTORY	Broker's working directory.

Component Management

Universal Broker is aware only of Workload Automation components that have been defined. It is the responsibility of Universal Broker to start, stop, and query these defined components.

One of the steps in the installation of a component is defining it to the local Universal Broker. These component definitions provide Universal Broker with the necessary information that it needs to manage the components.

Component Definitions

Component definitions are text files that define Workload Automation components to the Universal Broker. All UNIX component definition files are located in the Universal Broker component definition directory (specified with the COMPONENT_DIRECTORY configuration option).

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

The following table identifies all of the options that comprise Workload Automation for UNIX component definitions. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AUTOMATICALLY_START	Specification for whether the component automatically starts by the Universal Broker at start-up time or only on demand.
COMPONENT_NAME	Name by which clients know the component.
COMPONENT_TYPE	Type of component.
CONFIGURATION_FILE *	Component's configuration file name.
RESTART	Specification for whether or not the component should be restarted if it ends.
RESTART_CONDITIONS	Exit conditions criteria for which the server is considered eligible for restart.
RESTART_DELAY	Number of seconds to wait before restarting.
RESTART_MAX_FREQUENCY	Maximum frequency a server can be restarted.
RUNNING_MAXIMUM	Maximum number of this component that can run simultaneously.
START_COMMAND *	Command that starts the component.
WORKING_DIRECTORY *	Path used as the working directory of the component.

***** These options are required in the component definitions.

Universal Access Control List

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

Universal Broker reads in the UACL entries when the program is started. If the UACL file is changed, the new entries can be activated either by:

- Stopping and starting Universal Broker
- Sending Universal Broker a Universal Control REFRESH command, which instructs Universal Broker to reread all its configuration files, including the UACL file.

The UNIX REFRESH command is:

```
uctl –refresh –host BROKER-IPADDR
```

UACL Entries

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

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

UACL Entry Name	Description
UBROKER_ACCESS	Allows or denies access to Universal Broker services.
CERT_MAP	Maps a client X.509 certificate to a certificate identifier.
EVENT_ACCESS	Controls which Universal Enterprise Controller has read and delete access to the Universal Event Subsystem event data maintained by the Universal Broker.
REMOTE_CONFIG_ACCESS	Authorizes update access to the product configuration files and setting of the configuration managed mode of the Broker.

Universal Broker for IBM i

Configuration

Universal Broker reads configuration options only from the Universal Broker configuration file.

The Universal Broker configuration file is named **UNVPRD510/UNVCONF(UBROKER)**. File **UNVCONF** is a physical source file located in the **UNVPRD510** library. File member **UBROKER** contains the configuration options for the Universal Broker. File **UNVCONF** contains configuration members for the Workload Automation components. This file can be edited manually with any text editor.

Configuration Options

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

Option Name	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
CA_CERTIFICATES	Path to PEM formatted trusted CA X.509 certificates.
CERTIFICATE	Path to Broker's PEM formatted X.509 certificate.
CERTIFICATE_REVOCATION_LIST	Path to PEM formatted CRL.
CODE_PAGE	Text translation code page.
CTL_SSL_CIPHER_LIST	SSL cipher list for the control sessions.
DNS_CACHE_TIMEOUT	Time-out for DNS cache.
EVENT_GENERATION	Events to be generated as persistent events.
MESSAGE_DESTINATION	Location where messages are written.
MESSAGE_LANGUAGE	Language of written messages.
MESSAGE_LEVEL	Level of messages written.

MONITOR_EVENT_EXPIRATION	Duration of a monitoring event record in the Universal Broker local UES database.
PERSISTENT_EVENT_EXPIRATION	Duration of a persistent event record in the Universal Broker local UES database.
PRIVATE_KEY	Path to Broker's PEM formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Broker's PRIVATE_KEY.
RUNNING_MAX	Maximum number of simultaneous components.
SERVICE_BACKLOG	Service interface backlog size for pending connection requests.
SERVICE_IP_ADDRESS	TCP/IP address on which the Broker listens.
SERVICE_PORT	TCP/IP port number on which the Broker listens.
TRACE_FILE_LINES	Maximum number of lines written to the trace file.
TRACE_TABLE	Memory trace table specification.

Component Management

Universal Broker is aware only of Workload Automation components that have been defined. It is the responsibility of Universal Broker to start, stop, and query these defined components.

One of the steps in the installation of a component is defining it to the local Universal Broker. These component definitions provide Universal Broker with the necessary information that it needs to manage the components.

Component Definitions

Component definitions are text files that define Workload Automation components to the Universal Broker. All IBM i component definitions are located in the source physical file **UNVPRD510/UNVCOMP** as individual members.

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

The following table identifies all of the options that comprise Workload Automation for IBM i component definitions. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AUTOMATICALLY_START	Specification for whether or not the component automatically starts by the Universal Broker at start-up time or only on demand.
COMPONENT_NAME	Name by which clients know the component.

COMPONENT_TYPE	Type of component.
CONFIGURATION_FILE *	Component's configuration file name.
RESTART	Specification for whether or not the component should be restarted if it ends.
RESTART_CONDITIONS	Exit conditions criteria for which the server is considered eligible for restart.
RESTART_DELAY	Number of seconds to wait before restarting.
RESTART_MAX_FREQUENCY	Maximum frequency a server can be restarted.
RUNNING_MAXIMUM	Maximum number of this component that can run simultaneously.
START_COMMAND *	Component program name.
WORKING_DIRECTORY *	Path used as the working directory of the component.

* These options are required in the component definitions.

Universal Access Control List

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

Universal Broker reads in the UACL entries when the program is started. If the UACL file is changed, the new entries can be activated either by:

- Stopping and starting Universal Broker.
- Sending Universal Broker a Universal Control **REFRESH** command, which instructs Universal Broker to reread all its configuration files, including the UACL file.

The IBM i REFRESH command is:

```
STRUCT REFRESH(*YES) HOST(hostname)
```

UACL Entries

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

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

UACL Entry Name	Description
UBROKER_ACCESS	Allows or denies access to Universal Broker services.
CERT_MAP	Maps a client X.509 certificate to a certificate identifier.

EVENT_ACCESS	Controls which Universal Enterprise Controller has read and delete access to the Universal Event Subsystem event data maintained by the Universal Broker.
REMOTE_CONFIG_ACCESS	Authorizes update access to the product configuration files and setting of the configuration managed mode of the Universal Broker.

Universal Broker for HP NonStop

⚠ **Currently, HP NonStop runs Universal Broker 2.1.1. This page provides information for that version.**

Configuration

Universal Broker reads configuration options only from the Universal Broker configuration file.

The Universal Broker configuration file is named **UBRCFG**. This file can be edited manually with the EDIT TACL command.

Configuration Options

The following table summarizes all configuration options for Universal Broker for HP NonStop. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
CODE_PAGE	Text translation code page.
INSTALLATION_DIRECTORY	Base directory where product is installed.
MESSAGE_DESTINATION	Location where messages are written.
MESSAGE_LANGUAGE	Language of written messages.
MESSAGE_LEVEL	Level of messages written.
RUNNING_MAX	Maximum number of simultaneous components.
SERVICE_IP_ADDRESS	TCP/IP address on which the Broker listens.
SERVICE_PORT	TCP/IP port number on which the Broker listens.

Component Management

Universal Broker is aware only of Workload Automation components that have been defined. It is the responsibility of Universal Broker to start, stop, and query these defined components.

One of the steps in the installation of a component is defining it to the local Universal Broker. These component definitions provide Universal Broker with the necessary information that it needs to manage the components.

Component Definitions

Component definitions are text files that define Workload Automation components to the Universal Broker. All HP NonStop component definition files (EDIT files) are located in the component definition subvolume, **$SYSTEM.UNVCOMP**.

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

The following table identifies all of the options that comprise Workload Automation for HP NonStop component definitions. Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AUTOMATICALLY_START	Specification for whether or not the component automatically starts by the Universal Broker at start-up time or only on demand.
COMPONENT_NAME	Name by which clients know the component.
CONFIGURATION_FILE *	Component's configuration file name.
RUNNING_MAXIMUM	Maximum number of this component that can run simultaneously.
START_COMMAND *	Command that starts the component.
WORKING_DIRECTORY *	Path used as the working directory of the component.

***** These options are required in the component definitions.

Universal Access Control List

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

The Broker reads in the UACL entries when the program is started. If the UACL file is changed, the new entries can be activated by stopping and starting the Broker or by sending the Broker a Universal Control REFRESH command that will instruct the Broker to reread all its configuration files including the UACL file.

The HP NonStop REFRESH command is:

```
uctl –refresh –host BROKER-IPADDR
```

UACL Entries

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

The following table identifies all Universal Broker for HP NonStop UACL entries. Each **UACL Entry Name** is a link to detailed information about that option.

UACL Entry Name	Description
UBROKER_ACCESS	Allows or denies access to Universal Broker services.

Universal Broker Configuration Options

Universal Broker Configuration Options - Overview

Universal Broker Configuration Options

This page provides links to detailed information on the configuration options available for use with the Universal Broker. Information on how these options are used is documented in the operating system-specific pages of this document.

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

Configuration Options Information

For each configuration option, the following information is provided.

Description

Describes the configuration option and how it is used.

Usage

Provides a table of the following information:

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<Format / Value>					

Method

Identifies the method used to specify Universal Broker configuration options:

- Configuration File Keyword

Syntax

Identifies the syntax of the method used to specify the option:

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

(Operating System)

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

- IBM i
- HP NonStop
- UNIX
- Windows
- z/OS

Values

Identifies all possible values for the specified value type.

Defaults are identified in **bold type**.

<Additional Information>

Identifies any additional information specific to the option.

Configuration Options List

The following table identifies all Universal Broker configuration options.

Option	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
BIF_DIRECTORY	Broker Interface File directory that specifies where Universal Broker will create its interface file.
CA_CERTIFICATES	Path to PEM-formatted trusted CA X.509 certificates.
CERTIFICATE	Path to Broker's PEM-formatted X.509 certificate.
CERTIFICATE_REVOCATION_LIST	Path to PEM-formatted CRL.
CODE_PAGE	Text translation code page.
COMPONENT_DIRECTORY	Component definition file directory.
COMPONENT_PORT	TCP/IP port used for Broker-Component communications.
CTL_SSL_CIPHER_LIST	SSL cipher list for the control sessions.
DNS_CACHE_TIMEOUT	Time-out for DNS cache.
EVENT_GENERATION	Events to be generated as persistent events.
INSTALLATION_DIRECTORY	Base directory where product is installed.
LOG_DIRECTORY	Log file directory.
LOG_FILE_GENERATIONS	Total number of log files that will be saved within the log directory.

LOG_FILE_LINES	Total number of lines to be written to the log file before the log file is wrapped.
MESSAGE_DESTINATION	Location where messages are written.
MESSAGE_LANGUAGE	Language of messages written.
MESSAGE_LEVEL	Level of messages written.
MONITOR_EVENT_EXPIRATION	Duration of a monitoring event record in the Universal Broker local UES database.
MONITOR_EVENT_EXPIRATION	HFS or zFS database mount directory.
MOUNT_POINT_MODE	HFS or zFS permission mode for MOUNT_POINT.
NLS_DIRECTORY	UMC and UTT file directory.
PERSISTENT_EVENT_EXPIRATION	Duration of a persistent event record in the Universal Broker local UES database.
PID_FILE_DIRECTORY	PID file location.
PRIVATE_KEY	Path to Broker's PEM formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Broker's PRIVATE_KEY.
RUNNING_MAX	Maximum number of simultaneous components.
SAF_KEY_RING	SAF certificate key ring name.
SAF_KEY_RING_LABEL	SAF certificate key ring label.
SERVICE_BACKLOG	Service interface backlog size for pending connection requests.
SERVICE_IP_ADDRESS	TCP/IP address on which the Broker listens.
SERVICE_PORT	TCP/IP port number on which the Broker listens.
SMF_EXIT_LOAD_LIBRARY	UNVACTRT SMF exit load library.
SPOOL_DIRECTORY	Spool file directory.

SSL_IMPLEMENTATION	SSL implementation to be used for network configuration.
SYSTEM_ID	Universal Broker running on a system (O/S image).
TMP_DIRECTORY	Directory for temporary files.
TRACE_DIRECTORY	Directory for trace files.
TRACE_FILE_LINES	Maximum number of lines written to the trace file.
TRACE_TABLE	Memory trace table specification.
UCMD_STC_SUPPORT	Support for Universal Command started tasks.
UNIX_DB_DATA_SET	HFS or zFS data set used for the Universal Broker's databases.
UNIX_SPOOL_DATA_SET	HFS or zFS data set used for the Universal Broker's spool.
WORKING_DIRECTORY	Universal Broker’s working directory.

ACTIVITY_MONITORING - UBROKER configuration option

Description

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

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	activity_monitoring *option*	✅		✅	✅	✅

Values

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

Valid values for *option* are:

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

Default is yes.

BIF_DIRECTORY - UBROKER configuration option

Description

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

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	bif_directory *directory*			✓		

Values

directory is the name of the BIF directory.

Default is `/var/opt/universal`.

CA_CERTIFICATES - UBROKER 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 certificate authentication and verification is desired.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	ca_certificates *ddname* or *file*	✓		✓	✓	✓

Values

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

CERTIFICATE - UBROKER configuration option

Description

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

A UCMD Manager X.509 certificate is required if clients require Universal Broker authentication.

> **Note**
> If the CERTIFICATE option is used, the PRIVATE_KEY option also is required.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	certificate *ddname* or *file*	✓		✓	✓	✓

Values

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

CERTIFICATE_REVOCATION_LIST - UBROKER 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	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	crl *file* or *ddname*	✓		✓	✓	✓

Values

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

CODE_PAGE - UBROKER configuration option

Description

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

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

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	codepage *codepage*	✅	✅	✅	✅	✅

Value

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

codepage references a Universal Translate Table (UTT) file provided with the product. 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 Workload Automation components.

COMPONENT_DIRECTORY - UBROKER configuration option

Description

The COMPONENT_DIRECTORY option specifies the name of the directory where component definitions are stored.

All files located in the component directory are read and processed as component definitions. The name of each file found represents the component name.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	component_directory *directory*					

Value

directory is the name of the directory.

Relative path names are relative to the installation directory.

Default is `/etc/universal/comp`.

COMPONENT_PORT - UBROKER configuration option

Description

The COMPONENT_PORT option specifies the IP port on which components communicate with the Universal Broker.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	component_port *port*				✓	

Value

port is the IP port.

Valid values for *port* are:

- Numbers (for example, 7000)
- Service name (for example, **ubrokercmp**)

Default is 7987.

> **Note**
> It is recommended to use the default (**7987**) whenever possible.

CTL_SSL_CIPHER_LIST - UBROKER configuration option

Description

The CTL_SSL_CIPHER_LIST option specifies one or more SSL 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	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	ctl_ssl_cipher_list *cipherlist*	✓		✓	✓	✓

Values

cipherlist is a comma-separated list of SSL cipher suites. The list should be ordered with the most preferred suite first and the least preferred suite last.

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

Cipher Suite	Description
RC4-SHA	128-bit RC4 encryption and SHA-1 message digest.
RC4-MD5	128-bit RC4 encryption and MD5 message digest.
AES256-SHA	256-bit AES encryption and SHA-1 message digest.
AES128-SHA	128-bit AES encryption and SHA-1 message digest.
DES-CBC3-SHA	128-bit Triple-DES encryption and SHA-1 message digest.
DES-CBC-SHA	128-bit DES encryption and SHA-1 message digest.

Default is RC4-SHA,RC4-MD5,AES256-SHA,AES128-SHA,DES-CBC3-SHA, DES-CBC-SHA.

DNS_CACHE_TIMEOUT - UBROKER configuration option

Description

The DNS_CACHE_TIMEOUT option specifies the number of seconds that a DNS cached host entry remains valid.

When the host name resolver is asked to resolve a host name into an IP address, the host entry returned is saved in the DNS cache. The next call to resolve a host name will return the cached entry and not go back to the resolve. The cached entry is considered valid until the cache time-out period is reached.

The DNS cache provides a performance improvement as the resolution of a host name can take some time depending on the environment.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	dns_cache_timeout *seconds*	✓		✓	✓	✓

Value

seconds is the number of seconds that a DNS cached host entry remains valid.

A value of **0** disables caching of host entries.

Default is 120.

EVENT_GENERATION - UBROKER configuration option

Description

The EVENT_GENERATION option specifies which events are to be generated and processed as persistent events.

A persistent event record is saved in a Universal Enterprise Controller (UEC) database for long-term storage.

(For a list of all event types for all Workload Automation components, see Universal Event Subsystem 5.1.0 Event Definitions.)

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	event_generation *types*	✓		✓	✓	✓

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,!300
Generate all event types except for 200 through 250 and 300.

Default is X (no event types).*

INSTALLATION_DIRECTORY - UBROKER configuration option

Description

The INSTALLATION_DIRECTORY option specifies the Universal Broker base installation directory.

> **Note**
> This is a required option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	installation_directory *directory*		✓	✓	✓	

Value

directory is the name of the Universal Broker base installation directory.

A full path name is required.

HP NonStop	Universal Broker is installed in **`$SYSTEM.UNVBIN`**; this path should be specified.
UNIX	If Universal Broker is installed in **`/opt/universal/ubroker`**, specify that entire path name: **`/opt/universal/ubroker`**.
Windows	The default is set in the **`ubroker.conf`** file at installation time.

LOG_DIRECTORY - UBROKER configuration option

Description

The LOG_DIRECTORY option specifies the name of the directory 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
Configuration File Keyword	log_directory *directory*					

Value

directory is the name of the directory where log files are created.

Relative directory paths are relative to the Universal Broker installation directory. Fully qualified path names are recommended.

UNIX	**Default is `/var/opt/universal/log`.**
Windows	**Default is `log`.**

LOG_FILE_GENERATIONS - UBROKER configuration option

Description

The LOG_FILE_GENERATIONS option specifies the total number of log files that will be saved within the log directory.

Log file creation is specified by the MESSAGE_DESTINATION option (value = **logfile**).

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	log_file_generations *generations*					

Value

generations is the number of log files that will be saved within the log directory.

The maximum number of generations of log files that can be saved is 999.

Default is 5.

Note
If the value is decreased, only the specified number of generations will be maintained. The "excess" log files are not cleaned up immediately, but as the log files rotate, this "excess" will be cleaned up and reused.

LOG_FILE_LINES - UBROKER configuration option

Description

The LOG_FILE_LINES option specifies the total number of lines to be written to the Universal Broker log file (**unv.log**) before the log file is archived and a new log file is created.

Log file creation is specified by the MESSAGE_DESTINATION option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	log_file_lines *lines*					

Value

lines is the total number of lines to be written to the log file before the log file is wrapped. The average number of bytes per line is approximately 50.

The maximum number of lines that can be written is 2,147,483,647.

Default is 2000.

MESSAGE_DESTINATION - UBROKER configuration option

Description

The MESSAGE_DESTINATION option specifies the location where messages are written.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	message_dest *destination*	✔	✔	✔	✔	✔

Value

destination is the location where messages are written.

Valid values for *destination* are:

z/OS	• **logfile** Writes the messages to ddname **UNVLOG**. • **system** Writes the messages to the console as WTO messages. **Default for a console process is system.**
Windows	• **stderr** Writes the messages to the console. **stderr** is a valid value only if Universal Broker is running as a console application. • **system** Writes the messages to the Windows Application Event Log. **system** is the only allowable value if Universal Broker is running as a Windows service. **Default for a console application is stderr.**
UNIX	• **stderr** Writes the messages to the console. **stderr** is a valid value only if Universal Broker is running as a console application. • **logfile** Writes the messages to a log file. The log file location is specified by the Log Directory option. The current log file name is **unv.log**. Past generation log files are named **unvNNNN.log**, where **NNNN** equals the generation number. By default, five generations are kept. • **system** Writes the messages to the **syslog** daemon. **Default depends on how Universal Broker is started:** • Default for a console process is stderr. • Default for a daemon process is logfile.

IBM i	• **stderr** Writes the messages to the STDERR file. A batch job's STDERR file is allocated to the print file **QPRINT**. • **logfile** Writes the messages to the job's job log. • **system** Writes the messages to the system operator message queue **QSYSOPR**. The product is delivered with a value of **logfile**. If a value of **system** is preferred, you may want to reduce the number of messages written to the message queue by specifying a MESSAGE_LEVEL of **warn**. **Default is stderr.**

HP NonStop	• **stderr**
	Writes the messages to the console. **stderr** is a valid value only if Universal Broker is running as a console application.
	• **logfile**
	Writes the messages to a log file. The log file is located in the **$SYSTEM.UNVLOG** subvolume. The current log file name is **UNVLOG**. Past generation log files are named **unvlogNN**, where **NN** equals the generation number. Currently, default five generations are kept.
	• **system**
	Writes the messages to the **syslog** daemon.
	
	Default depends on how Universal Broker is started:
	  • Default for a console process is stderr.
	  • Default for a daemon processs is logfile.

MESSAGE_LANGUAGE - UBROKER configuration option

Description

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

There is a message catalog for each language. The first three characters of the language are used as a three-character suffix of the member name. All UMC files have a **.UMC** extension.

> **Note**
> Currently, the only message catalog provided is for English.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	language *language*	✓	✓	✓	✓	✓

Values

language is the name of the UMC file.

z/OS	*language* translates to a member name of the library allocated on the **UNVNLS** DD statement. Universal Broker message catalog member names start with characters **USSMC**.
UNIX	The location of the UMC file is specified by the NLS_DIRECTORY option.
IBM i	UMC file members are located in the physical source file **UNVPRD510/UNVNLS**.
HP NonStop	UMC files are located in subvolume **$SYSTEM.UNVNLS**

Default is *ENGLISH* (UMC member USSMCENG is used).

MESSAGE_LEVEL - UBROKER 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
Configuration File Keyword	message_level *level*	✓	✓	✓	✓	✓

Values

level is the level of messages to write.

Valid values for *level* are:

- **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	The trace file name is **`UNVTMP510/UNVTRCUBR)`**.
HP NonStop	The trace file name is **`UBRTRC`**. It is created in the **`$SYSTEM.UNVTRACE`** subvolume.
UNIX	The trace file name depends on how it is started: • If running as a console application, the file name is **`ubroker.trc`**. • If running as a daemon, the file name is **`ubrokerd.trc`**. The trace file is created in the directory **`/var/opt/universal/trace`**.

Windows	• The trace file name depends on how it was started:
	  ◦ If running as a console application, the file name is **`ubroker.trc`**.
	  ◦ If running as a service, the file name is **`ubrsvc.trc`**.
	
	The trace file is created in the installation directory of Universal Broker, which defaults to:
	`C:\Program Files\Universal\Ubroker`

z/OS	There are two possible destinations of the trace data: 1. If ddname **UNVTRMDL** is defined in the UBROKER started task procedure, a sequential data set is created using the data set allocated to UNVTRMDL as a model. The dynamically allocated trace data set name is **`#HLQ.UBR.Dyymmdd.Thhmmss`**, where: • **#HLQ** is the data set name allocated on the UNVTRMDL ddname. • **yymmdd** is the year, month, and day. • **hhmmss** is the hour, minute, second the data set was allocated. The amount of space allocated for trace data sets modeled after **UNVTRMDL** is based upon the TRACE_FILE_LINES configuration option and the record format of the model data set. If the model data set is fixed record format, the total amount of space measured in bytes is TRACE_FILE_LINES *LRECL. If the model data set is variable record format, the total amount of space measured in bytes is TRACE_FILE_LINES * 50 (50 is considered the average length of a trace file record). The number of cylinders is calculated from the total amount of space in bytes. The total number of cylinders is calculated base on a total of 16 extents being allocated. The formula is cylCount = (totalSize / 16) / 750000. The allocation unit is set to cylinders and the primary and secondary space allocation is set to cylCount (that is, SPACE=(CYL,(cylCount,cylCount),RLSE)). 2. If ddname **UNVTRMDL** is not defined in the UBROKER started task procedure, member name **UBROKER** is created in the PDS or PDS/E allocated to the UNVTRACE ddname.

Depending on the error condition being diagnosed, it is possible that the member name of the **UNVTRACE** PDS or PDS/E is not created. If this occurs, the **UNVTRMDL** ddname must be used to create a sequential data set name.

The records written to PDS and PDS/E members cannot be wrapped, so the TRACE_FILE_LINES limit has no effect on the maximum number of trace records written to the member.

MONITOR_EVENT_EXPIRATION - UBROKER configuration option

Description

The MONITOR_EVENT_EXPIRATION option specifies the duration of an event record, for an event used for product activity monitoring, in the Universal Broker local UES database.

If a monitoring event record is not delivered to UEC within this time period, Universal Broker will delete the record from the local UES database. (A monitoring event record is not saved in a UEC database for long-term storage.)

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	monitor_event_expiration *seconds*	✓		✓	✓	✓

Values

seconds is the amount of time (in seconds) that a monitoring event record will remain in the database.

Default is 600 (10 minutes).

MOUNT_POINT - UBROKER configuration option

Description

The MOUNT_POINT option specifies the z/OS UNIX directory in which the HFS or zFS data sets are mounted. The actual mount points will be subdirectories named after the HFS or zFS data set names being mounted.

HFS data sets are specified by either of the following:

- UNIX_DB_DATA_SET and UNIX_SPOOL_DATA_SET options.
- **UNVDB** and **UNVSPOOL** ddnames.

zFS data sets are specified only by the UNIX_DB_DATA_SET and UNIX_SPOOL_DATA_SET options. zFS data set names cannot be specified by ddname.

The mount points are created by Universal Broker if they do not exist. The z/OS UNIX permission mode is set to the value specified by the MOUNT_POINT_MODE option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	mount_point *directory*					

Values

directory is the z/OS UNIX directory in which the HFS or zFS data sets are mounted.

Default is /tmp.

MOUNT_POINT_MODE - UBROKER configuration option

Description

The MOUNT_POINT_MODE option specifies the z/OS UNIX access permission mode value with which the mounted database file system's root directory is set.

The z/OS UNIX database file system (HFS or zFS) is initialized only if the file **.inited** is not found in the root directory. When initialization is performed, **.inited** is created; initialization will not be performed again.

If you need to customize the directory ownership or permissions, define the file **.inited** in the file system's root directory; the Broker will not perform its initialization.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	mount_point_mode *mode*					

Values

mode is the z/OS UNIX permission mode value, which is a sum of the permission modes to be granted.

The following table describes each mode.

Mode	Description
100	User execute permission.
200	User write permission.
400	User read permission.
010	Group execute permission.
020	Group write permission.
040	Group read permission.
001	Other execute permission.
002	Other write permission.
004	Other read permission.

The format of *mode* is the same as the “change mode” USS command **chmod**. It is an octal number that specifies the permission mode value corresponding to the user, group, and other permission mode fields.

Refer to the IBM *UNIX System Services Command Reference* for complete details on the **chmod** command.

Default is 750, which specifies:

- Read-write-execute access for the user
- Read-execute access for the group
- No access for other

NLS_DIRECTORY - UBROKER configuration option

Description

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

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	nls_directory *directory*			✓	✓	

Values

directory is the name of the directory where the files are located.

Full path names are recommended.

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

Defaults

UNIX
Default is **`/opt/universal/nls`**.

Windows
Default is **`\nls`**.

PERSISTENT_EVENT_EXPIRATION - UBROKER configuration option

Description

The PERSISTENT_EVENT_EXPIRATION option specifies the duration of an event record, for an event identified as a persistent event, in the Universal Broker local UES database.

If a persistent event record is not delivered to Universal Enterprise Controller (UEC) within this time period, Universal Broker will delete the record from the local UES database. (A persistent event record is saved in a UEC database for long-term storage.)

> **Note**
> Events are identified as persistent events via the EVENT_GENERATION option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	persistent_event_expiration *seconds*	✓		✓	✓	✓

Values

seconds is the amount of time (in seconds) that a persistent event record will remain in the database.

Default is 172800 (2 days).

PID_FILE_DIRECTORY - UBROKER configuration option

Description

The PID_FILE_DIRECTORY option specifies the name of the directory that Universal Broker uses for its PID file.

The PID file is used by Universal Broker to ensure that only one instance is executing at any one time.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	pid_file_directory *directory*					

Values

directory is the name of the directory for the PID file.

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

Default is `/var/opt/universal`.

Note
If the default value is changed, the PID file directory location in the Universal Broker startup script requires the same value. See Starting Universal Broker for UNIX for details on the Broker startup script.

PRIVATE_KEY - UBROKER configuration option

Description

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

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

z/OS
PRIVATE_KEY is used only when the SSL_IMPLEMENTATION option is set to **openssl**.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	private_key *ddname* or *file*	✓		✓	✓	✓

Values

z/OS	*ddname* 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.
UNIX	*file* is the path of the PEM-formatted RSA private key file that corresponds to the X.509 certificates.
Windows	*file* is the path of the PEM-formatted RSA private key file that corresponds to the X.509 certificates.
IBM i	*file* 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. If not, the library list ***LIBL** is searched for the first occurrence of the file name.

PRIVATE_KEY_PWD - UBROKER 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 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	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	private_key_password *password*	✓		✓	✓	✓

Values

password is the password for the private key.

RUNNING_MAX - UBROKER configuration option

Description

The RUNNING_MAX option specifies the maximum number of components that can run simultaneously.

If this maximum is reached, any command received to start a component is rejected.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	running_max *maximum*	✓		✓	✓	✓

Values

maximum is the maximum number of components that can be run simultaneously.

Default is 100.

SAF_KEY_RING - UBROKER configuration option

Description

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

The key ring must be associated with the user profile with which the Universal Broker started task executes.

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

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	saf_key_ring *name*					

Values

name is the name of the SAF certificate key ring.

SAF_KEY_RING_LABEL - UBROKER configuration option

Description

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

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

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	saf_key_ring_label *label*					

Values

label is the label of the SAF certificate key ring.

Default is the default certificate in the key ring.

SERVICE_BACKLOG - UBROKER configuration option

Description

The SERVICE_BACKLOG option specifies the service interface backlog size for pending connection requests.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	service_backlog *size*	✓		✓	✓	✓

Values

size is the service interface backlog size.

size must be greater than 0.

Default is 100.

z/OS
The system-wide default maximum backlog size for TCP/IP is **10**. The **TCPIP.PROFILE** parameter **SOMAXCONN** sets the maximum backlog size.

If you require a SERVICE_BACKLOG *size* greater than **10**, the **SOMAXCONN** value must be increased.

SERVICE_IP_ADDRESS - UBROKER configuration option

Description

The SERVICE_IP_ADDRESS option specifies the IP interface on which to accept network connection requests.

SERVICE_IP_ADDRESS is useful only if the system has multiple IP interfaces.

- If the system has multiple interfaces and SERVICE_IP_ADDRESS is not used, connection requests are accepted on all interfaces defined on the system.
- If the system has only one interface, do not use SERVICE_IP_ADDRESS.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	service_ip_address *ipaddress*	✓	✓	✓	✓	✓

Values

ipaddress is the IP address on which to accept network connection requests.

Valid values for *ipaddress* are:

- Dotted numeric format (for example, *20.30.40.50*)
- Domain name format (for example, *myinterface*).

> **Note**
> An asterisk (*****) specifies all interfaces.

Default is *.

SERVICE_PORT - UBROKER configuration option

Description

The SERVICE_PORT option specifies IP port on which to accept network connection requests.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	service_port *port*	✓	✓	✓	✓	✓

Values

port is the IP port on which to accept network connection requests.

Valid values for *port* are:

- Numeric value (for example, 7000)
- Service name (for example, **ubroker**)

Default is 7887.

> **Note**
> It is recommended that the default value be used, if possible.

SMF_EXIT_LOAD_LIBRARY - UBROKER configuration option

Description

The SMF_EXIT_LOAD_LIBRARY option specifies a cataloged data set from which the SMF exit routine **UNVACTRT** is loaded and dynamically installed at exit point **`SYSSTC.IEFACTRT`**.

If SMF_EXIT_LOAD_LIBRARY is not specified, the exit routine is not dynamically installed. It then must be installed prior to the Universal Broker address space starting with an alternative method. (See the zOS Configuration - SMF Exits for alternative methods.)

The exit routine is deleted when last the Universal Broker address space running is stopped. If multiple Universal Broker address spaces are running, the last Universal Broker to stop removes the exit routine.

SMF_EXIT_LOAD_LIBRARY is required if the UCMD_STC_SUPPORT option is set to **yes**.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	smf_exit_load_library *dsn*					

Values

dsn is the cataloged data set from which the SMF exit routine is loaded and installed.

SPOOL_DIRECTORY - UBROKER configuration option

Description

The SPOOL_DIRECTORY option specifies the directory name that Universal Broker uses for its spool database files.

The Universal Broker spool files should not require a large amount of disk space; two or three MB should be sufficient.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	spool_directory *directory*			✓	✓	

Values

directory is the name of the directory for spool database files.

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

Defaults

Windows	Default is **`C:\Program Files\Universal\spool`**.
UNIX	Default is **`/var/opt/universal/spool`**.

SSL_IMPLEMENTATION - UBROKER configuration option

Description

The SSL_IMPLEMENTATION option specifies the Secure Socket Layer (SSL) implementation to be used for network communications.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	ssl_implementation *option*					

Values

option is the SSL implementation to be used.

Valid values for option are:

- **openssl**
  OpenSSL SSL library is used for the SSL protocol.
- **system**
  z/OS System SSL library is used for the SSL protocol. The z/OS System SSL library has installation and configuration prerequisites. (See the Workload Automation 5.1.0 Installation and Administration Guide for a description of the prerequisites before using System SSL.)

Default is openssl.

SYSTEM_ID - UBROKER configuration option

Description

The SYSTEM_ID option uniquely identifies the Universal Broker.

- If SYSTEM_ID is not used to identify the Universal Broker, the default (a blank value) is used. If there are more than one Universal Brokers running on an O/S image, only one can use the default. SYSTEM_ID must be used to identify all of the other Universal Brokers.
- If SYSTEM_ID is used to identify the Universal Broker, all of its Manager jobs must include the SYSTEM_ID option to identify the Universal Broker.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	system_id *ID*					

Values

ID is the system identifier of the local Universal Broker (1 to 8 characters in length).

Valid values for *ID* are:.

- First character must be alphabetic.
- All subsequent characters must be alphabetic or numeric.

Default is a blank value.

TMP_DIRECTORY - UBROKER configuration option

Description

The TMP_DIRECTORY option specifies the directory that the Universal Broker uses for temporary files.

z/OS
TMP_DIRECTORY specifies the name of a z/OS UNIX directory.

The amount of space required for the temporary directory is small. Most of the files are IPC pipes used for Broker and Server IPC.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	tmp_directory *directory*			✓	✓	✓

Values

directory is the name of the directory.

A fully qualified path name is recommended.

Defaults

UNIX	Default is **`/var/opt/universal/tmp`**.
Windows	Default is **`..\tmp`**.
z/OS	Default is **`/tmp`**.

TRACE_DIRECTORY - UBROKER configuration option

Description

The TRACE_DIRECTORY option specifies the directory that the Universal Broker uses for trace files.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	trace_directory *directory*			✓	✓	

Values

directory is the name of the directory for trace files.

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

Defaults

Windows	Default is **`C:\Program Files\Universal\UBroker`**.
UNIX	Default is **`/var/opt/universal/trace`**.

TRACE_FILE_LINES - UBROKER configuration option

Description

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

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

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

IBM i
Trace file records are 366 bytes long.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	trace_file_lines *lines*	✓		✓	✓	✓

Values

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

Default = 500,000.

Note
If space is limited in the trace file directory, set *lines* to a smaller value.

IBM i
If space is limited in the trace file ASP (ASP in which the **UNVTMP510** library is located), set the default to a smaller value. If a larger value is required, either create or change the maximum number of records allowed in the physical file **UNVTMP510/UNVTRCUBR** and increase this setting. The largest value allowed without increasing the number of records allowed is **509000**.

TRACE_TABLE - UBROKER configuration option

Description

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

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

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	trace_table *size, condition*	✓		✓	✓	✓

Values

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

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

- **M** indicates that the size is specified in megabytes
- **K** indicates that the size is specified in kilobytes

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

> **Note**
> If *size* is **0**, the trace table is not used.

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

Possible values for *condition* are:

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

UCMD_STC_SUPPORT - UBROKER configuration option

Description

The UCMD_STC_SUPPORT option specifies whether or not the Universal Broker establishes the environment to support Universal Command start task requests.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	ucmd_stc_support *option*					

Values

option is the specification for whether or not the Universal Broker establishes the environment.

Valid values for *option* are:

- **yes**
  Universal Broker establishes the environment.
- **no**
  Universal Broker does not establish the environment.

Note
If the value for *option* = **no**, Universal Command will not support the execution of started tasks.

The environment support for Universal Command started tasks consists of installing SMF exit routine **UNVACTRT** at exit point **SYSSTC.IEFACTRT** and a small amount of CSA storage for address space communication.

Default is yes.

UNIX_DB_DATA_SET - UBROKER configuration option

Description

The UNIX_DB_DATA_SET option specifies the HFS or zFS data set used for the Universal Broker's databases. The data set can be mounted prior to starting the Broker. If not, the Broker will mount the data set at a specified mount point derived from the MOUNT_POINT option.

UNIX_DB_DATA_SET is the only way to specify a zFS data set. HFS data sets can be allocated in the Broker's started task procedure as ddname **UNVDB**. zFS data sets cannot be allocated on a ddname.

> **Note**
> When using a zFS data set, the **UNVDB** ddname statement in the Broker's started task procedure should be removed.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	unix_db_data_set *DSN*					✓

Values

DSN is the HFS or zFS data set used for the databases.

UNIX_SPOOL_DATA_SET - UBROKER configuration option

Description

The UNIX_SPOOL_DATA_SET option specifies the HFS or zFS data set used for the Universal Broker's spool. The data set can be mounted prior to starting the Broker. If not, the Broker will mount the data set at a specified mount point derived from the MOUNT_POINT option.

UNIX_SPOOL_DATA_SET is the only way to specify a zFS data set. HFS data sets can be allocated in the Broker's started task procedure as ddname **UNVSPOOL**. zFS data sets cannot be allocated on a ddname.

> **Note**
> When using a zFS data set, the **UNVSPOOL** ddname statement in the Broker's started task procedure should be removed.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	unix_spool_data_set *DSN*					

Values

DSN is the HFS or zFS data set used for the spool.

WORKING_DIRECTORY - UBROKER configuration option

Description

The WORKING_DIRECTORY option specifies the directory name that the Universal Broker uses as its working directory.

WORKING_DIRECTORY may be of value if you want the Universal Broker daemon to use a working directory other than the default. Ideally, daemons should use the root directory as their working directory. This prevents the need to stop the daemon should a file system require unmounting.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	working_directory *directory*			✓	✓	

Values

directory is the name of the working directory.

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

Defaults

Windows	Default is Universal Broker installation directory.
UNIX	Default is startup directory.

Universal Broker Component Definition Options

Universal Broker Component Definition Options - Overview

Universal Broker Component Definition Options

This page provides links, in the Component Definition Options table, below to detailed information about the options that comprise the Workload Automation component definitions provided to the Universal Broker for all server components:

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

Information on how component definitions are used is documented in the operating system-specific pages of this Universal Broker Reference Guide.

To see which options comprise the component definition for each server component, see:

- Universal Automation Center Agent Server
- Universal Command Server
- Universal Data Mover Server
- Universal Event Monitor Server
- Universal Control Server
- Universal Application Container Server

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 Workload Automation component definition option:

- Component Definition Keyword

Syntax

Identifies the syntax of the method used to specify the option:

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

(Operating System)

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

- IBM i
- HP NonStop
- UNIX
- Windows
- z/OS

Values

Identifies all possible values for the specified value type.

Defaults are identified in **bold type**.

{anchor:Component Definition Options)Component Definition Options

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

Option	Description
AUTOMATICALLY_START	Specification for whether the component automatically starts by the Universal Broker at start-up time or only on demand.
COMPONENT_NAME	Name by which clients know the component.
COMPONENT_TYPE	Type of component.
CONFIGURATION_FILE *	Component's configuration file name.
RESTART	Specification for whether or not the component should be restarted if it ends.
RESTART_CONDITIONS	Exit conditions criteria for which the server is considered eligible for restart.
RESTART_DELAY	Number of seconds to wait before restarting.
RESTART_MAX_FREQUENCY	Maximum frequency a server can be restarted.
RUNNING_MAXIMUM	Maximum number of this component that can run simultaneously.
START_COMMAND *	Component program name.
WORKING_DIRECTORY *	Path used as the working directory of the component.

* These options are required in all component definitions.

AUTOMATICALLY_START - UBROKER Component Definition option

Description

The AUTOMATICALLY_START option specifies whether the component automatically starts by the Universal Broker at startup time or only on demand.

> **Note**
> AUTOMATICALLY_START is optional in a component definition.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Component Definition Keyword	auto_start *option*	✓	✓	✓	✓	✓

Values

option is the specification for how the component is started.

Valid values for *option* are:

- **yes**
  Component is started automatically by Universal Broker.
- **no**
  Component is started only on demand.

Default is no.

COMPONENT_NAME - UBROKER Component Definition option

Description

The COMPONENT_NAME option specifies the name by which the clients know the component.

> **Note**
> COMPONENT_NAME is optional in a component definition. If it is not specified, the file name is used as the component name.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Component Definition Keyword	component_name *name*	✓	✓	✓	✓	✓

Values

name is the name by which the clients know the component.

COMPONENT_TYPE - UBROKER Component Definition option

Description

The COMPONENT_TYPE option specifies the type of component.

Some components can execute multiple instances simultaneously with different component names. The COMPONENT_TYPE specifies the common type of component that applies to this component definition.

> **Note**
> COMPONENT_TYPE is optional in a component definition. If it is not specified, the component name is used.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Component Definition Keyword	component_type *type*	✓		✓	✓	✓

Values

type is the type of component.

CONFIGURATION_FILE - UBROKER Component Definition option

Description

The CONFIGURATION_FILE option specifies the component's configuration file name (member name in z/OS).

> **Note**
> CONFIGURATION_FILE is required in a component definition.

IBM i	Non-qualified file names are located in the library list ***LIBL**.
HP NonStop	Relative paths are relative to the component's working subvolume.
UNIX	Relative paths are relative to the component's working directory.
z/OS	Member names are located in the **UNVCONF** library allocated to the **UNVCONF** ddname.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Component Definition Keyword	configuration_file *member* or configuration_file *filename*	✓	✓	✓	✓	✓

Values

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

RESTART - UBROKER Component Definition option

Description

The RESTART option specifies whether or not the component should be restarted if it ends.

The component restart facility is only applicable to auto-started components.

A component is restarted when the following conditions are met:

1. Universal Broker is not in shutdown mode.
2. Component has not been stopped by the Broker, Universal Control, or Universal Enterprise Controller. This is considered a controlled shutdown.
3. RESTART option value is **yes**.
4. Component's exit conditions must meet one of the values specified by the RESTART_CONDITIONS option.
5. Component must not have been restarted more than specified by the RESTART_MAX_FREQUENCY option.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	restart *options*	✓		✓	✓	✓

Values

options is the specification for whether or not the component should be restarted.

Valid values for *options* are:

- **yes**
  Component should be restarted if it meets the restart criteria. (This is applicable only for auto-started components.)
- **no**
  Component should not be restarted.

Default is no.

RESTART_CONDITIONS - UBROKER Component Definition option

Description

The RESTART_CONDITIONS option specifies the exit conditions of the component for which the component should be considered eligible for restart.

If the exit conditions of the component do not meet the criteria, it will not be restarted.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	restart_conditions *conditions*	✓		✓	✓	✓

Values

conditions is a comma-separated list of exit conditions.

The exit conditions names are based on the Workload Automation 5 return codes. Category names are used instead of numeric values, as the exit code numeric value may not be consistent across all platforms.

The exit conditions are:

- ABNORMAL - Component ended abnormally due to a UNIX signal, Windows Exception, z/OS ABEND, etc.
- SUCCESS - Component ended normally with exit code 0.
- WARN - Component ended normally with a warning exit code.
- ERROR - Component ended normally with an error exit code.
- FATAL - Component ended normally with a fatal exit code.
- CONFIG - Component ended normally with a configuration error exit code.
- SECURITY - Component ended normally with a security related exit code.
- NETWORK - Component ended normally with a network related exit code.
- SHUTDOWN - Component ended normally with a shutdown related exit code.
- LICENSE - Component ended normally with a license violation related exit code.
- ALL - All of the above.

Default is ABNORMAL.

RESTART_DELAY - UBROKER Component Definition option

Description

The RESTART_DELAY option specifies the number of seconds to wait from the time the Universal Broker detects the component has ended until the Broker restarts the component.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	restart_delay *seconds*	✓		✓	✓	✓

Values

seconds is the number of seconds to wait.

Default is 5.

RESTART_MAX_FREQUENCY - UBROKER Component Definition option

Description

The RESTART_MAX_FREQUENCY option specifies the maximum frequency in which a component can be restarted in a specific time interval.

If a component becomes eligible for restart but exceeds the maximum restart frequency, it will not be restarted.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Configuration File Keyword	restart_max_frequency *number/interval*	✅		✅	✅	✅

Values

number is the maximum number of restarts.

interval is the time interval in which the specified maximum number of restarts (*number*) is allowed.

Valid values for interval are **week**, **day**, **hour**, and **minute**.

Default is 2 / day.

RUNNING_MAXIMUM - UBROKER Component Definition option

Description

The RUNNING_MAXIMUM option specifies the maximum number of this component that can run simultaneously.

If this maximum number is reached, any command received to start the component is rejected.

> **Note**
> RUNNING_MAXIMUM is optional in a component definition.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Component Definition Keyword	running_max *maximum*	✓	✓	✓	✓	✓

Values

maximum is the maximum number of this component that can run simultaneously.

Default is 100.

START_COMMAND - UBROKER Component Definition option

Description

The START_COMMAND option specifies the full path name (member name for z/OS) of the program.

Optionally, START_COMMAND also can specify command line options.

> **Note**
> START_COMMAND is required in a component definition.

z/OS	Member names are located in the **SUNVLOAD** library.
IBM i	Non-qualified program names are located in the library list ***LIBL**.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Component Definition Keyword	start_command *member* **or** start_command *name[options]*	✓	✓	✓	✓	✓

Values

member / *name* is the program name of the component.

options is the optional list of command line options.

> **z/OS**
> *options* is not a valid value for START_COMMAND.

WORKING_DIRECTORY - UBROKER Component Definition option

Description

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

> **Note**
> WORKING_DIRECTORY is required in a component definition.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
Component Definition Keyword	working_directory *directory*	✓	✓	✓	✓	✓

Values

directory is the full path name of the working directory.

Default is (.).

IBM i	working_directory serves as a required placeholder only. Do not change its value.
HP NonStop	The path is the path used as the working subvolume of the component. Relative path names are relative to the Universal Broker working subvolume.
UNIX	Relative path names are relative to the Universal Broker working directory. Full path names are recommended.
Windows	Relative path names are relative to the Universal Broker working directory. Full path names are recommended.
z/OS	The path is the z/OS UNIX path used as the working directory of the component.

Universal Broker UACL Entries

Universal Broker UACL Entries - Overview

Universal Broker UACL Entries

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

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

Information on how these UACL entries are used is documented in the operating system-specific pages of this document.

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: Workload Automation 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 Broker UACL Entries.

UACL Entry	Description
UBROKER_ACCESS	Allows or denies a Workload Automation component access to Universal Broker services.
CERT_MAP	Maps a client X.509 certificate to certificate identifier.
EVENT_ACCESS	Controls which Universal Enterprise Controller has read and delete access to the Universal Event Subsystem event data maintained by the Universal Broker.
REMOTE_CONFIG_ACCESS	Authorizes update access to the product configuration files and setting of the configuration managed mode of the Broker. There are two forms to this entry: • **remote_config_access** • **remote_config_cert_access**

UBROKER_ACCESS - UBROKER UACL entry

Description

A UBROKER_ACCESS UACL entry specifies whether to allow or deny a Workload Automation component access to Universal Broker services.

If a request from a component comes from an IP address identified in this UBROKER_ACCESS entry, the rule is considered a match. The first matching rule is used to control access.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
UACL File Keyword	ubroker_access *host,access*	✓	✓	✓	✓	✓

Values

host specifies an IP address of a Workload Automation component.

(See UACL Entries for details on *host* specification syntax.)

access specifies whether the connection is allowed or denied.

Valid values for *access* are:

- **deny**
  IP connection is denied. No message is returned to the remote end. The connection is immediately closed.
- **allow**
  IP connection is accepted and processed.

Default is allow.

CERT_MAP - UBROKER UACL entry

Description

A CERT_MAP UACL entry maps a client X.509 certificate to certificate identifier.

CERT_MAP defines one or more certificate fields and values that are used to match against the client's certificate. All of the fields defined by CERT_MAP must match the client certificate in order for the rule to be considered a match.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
UACL File Keyword	cert_map *id=certid,cert-field(s)*	✓		✓	✓	✓

Values

id is the certificate identifier.

cert-fields is a comma-separated list of one or more certificate fields.

(See X.509 Certificates for a detail discussion on the *cert-fields* values.)

EVENT_ACCESS - UBROKER UACL entry

Description

A EVENT_ACCESS entry controls which Universal Enterprise Controller has read and delete access to the Universal Event Subsystem event data maintained by the Universal Broker.

There are two forms of the EVENT_ACCESS entry:

- **event_access** is based on the host name and user ID of the client.
- **event_cert_access** is based on a certificate map of the client.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
UACL File Keyword	event_access *host,remote_user,read_access,delete_access* event_cert_access *certid,read_access,delete_access*	✓		✓	✓	✓

Values

host specifies an IP address of a Workload Automation 5 component.

remote_user is the user identifier with which Universal Enterprise Controller is executing on the remote system.

(See UACL Entries - Client Identification for details on *host* and *remote_user* specification syntax.)

read_access specifies whether or not reading event data is allowed.

Valid values for *read_access* are:

- **deny**
  Access is denied to the read request.
- **allow**
  Access is allowed to the read request.

delete_access specifies whether or not deleting event data is allowed.

Valid values for *delete_access* are:

- **deny**
  Access is denied to the delete request.
- **allow**
  Access is allowed to the delete request.

Defaults

event_access ALL,*,allow,deny

event_cert_access *,allow,deny

Examples

event_access 10.20.30.40,uecprod,allow,allow

```
event_access ALL,*,deny,deny
event_cert_access uecprod,allow,allow
event_cert_access *,deny,deny
```

REMOTE_CONFIG_ACCESS - UBROKER UACL entry

Description

A REMOTE_CONFIG_ACCESS entry authorizes update access to the product configuration files and setting of the configuration managed mode of the Broker.

Universal Enterprise Controller requests this access when it needs to configure a product using its remote configuration capabilities.

There are two forms of the REMOTE_CONFIG_ACCESS entry:

- **remote_config_access** is based on the host name and user ID of the client.
- **remote_config_cert_access** is based on a certificate map of the client.

Usage

Method	Syntax	IBM i	HP NonStop	UNIX	Windows	z/OS
UACL File Keyword	remote_config_access *host,remote_user,update_access,control_access* remote_config_cert_access *certid,update_access,control_access*	✓		✓	✓	✓

Values

host specifies an IP address of a Workload Automation 5 component.

remote_user is the user identifier with which Universal Enterprise Controller is executing on the remote system.

(See UACL Entries - Client Identification for details on *host* and *remote_user* specification syntax.)

update_access specifies whether or not configuration file updates are allowed.

Valid values for *update_access* are:

- **deny**
  Access is denied to the update request.
- **allow**
  Access is allowed to the update request.

control_access specifies whether or not the Broker can be placed into managed mode or taken out of managed mode.

Valid values for *control_access* are:

- **deny**
  Access is denied to the managed mode request.
- **allow**
  Access is allowed to the managed mode request.

Defaults

remote_config_access ALL,*,deny,deny

remote_config_cert_access *,deny,deny

Examples

remote_config_access 10.20.30.40,uecprod,allow,allow

```
remote_config_access ALL,*,deny,deny
remote_config_cert_access uecprod,allow,allow
remote_config_cert_access *,deny,deny
```

Universal Broker Configuration Options Refresh

Universal Broker Configuration Options Refresh - Overview

Universal Broker Configuration Options Refresh

As with all Workload Automation components, all Universal Broker configuration options can be modified by editing the configuration file directly.

However, unlike other components, not all Universal Broker options can be modified via I-Management Console. (In I-Management Console, these Universal Broker options are read-only.) These options can be modified only by editing the Universal Broker configuration file, **`unbroker.conf`**. For these modifications to be updated in Universal Broker memory and take immediate effect, Universal Broker must be recycled; they do not take effect when Universal Broker is simply refreshed. (See Options - Configuration File Editable Only, Recycle Required.)

All other Universal Broker options can be modified either:

- By editing **`ubroker.conf`**.
- Via I-Management Console.
- Via the Universal Configuration Manager.

Depending on the option, for a modification to be updated in Universal Broker memory and take immediate effect:

- Universal Broker must be recycled. (See Options - I-Management Console and Configuration File Editable, Recycle Required.)
- Universal Broker must be refreshed:
  - By issuing a Universal Control configuration refresh request (via the REFRESH_CMD configuration option), if the modifications are made in the configuration file.
  - Automatically, if the modifications are made via I-Management Console or the Universal Configuration Manager.

  (See Options - I-Management Console and Configuration File Editable, Refresh Required.)

Options - Configuration File Editable Only, Recycle Required

Configuration File Editable Only, Recycle Required

The following table identifies Universal Broker options that you can modify only by editing the Universal Broker configuration file. (In I-Management Console, these options are Read-Only.)

Additionally, Universal Broker must be recycled in order for the modified values to be used. These options are not updated when Universal Broker simply is refreshed.

Option	Description
BIF_DIRECTORY	Broker Interface File directory that specifies where Universal Broker will create its interface file.
COMPONENT_DIRECTORY	Component definition file directory.
INSTALLATION_DIRECTORY	Base directory where product is installed.
MOUNT_POINT	HFS or zFS database mount directory.
MOUNT_POINT_MODE	HFS or zFS permission mode for MOUNT_POINT.
NLS_DIRECTORY	UMC and UTT file directory.
PID_FILE_DIRECTORY	PID file location.
SMF_EXIT_LOAD_LIBRARY	UNVACTRT SMF exit load library.
SPOOL_DIRECTORY	Spool file directory.
SYSTEM_ID	Universal Broker running on a system (O/S image).
UCMD_STC_SUPPORT	Support for Universal Command started tasks.
UNIX_DB_DATA_SET	HFS or zFS data set used for the Universal Broker's databases.
UNIX_SPOOL_DATA_SET	HFS or zFS data set used for the Universal Broker's spool.

A Stonebranch tip:
If the PID_FILE_DIRECTORY value is modified, the UNIX script that starts/stops/restarts the Universal Broker, **ubrokerd**, also must be modified to indicate the location of the Broker's PID file.

If **ubrokerd** is not modified, it will not know the Process ID of the executing Universal Broker. Thus, it will not be able to return status information of the executing Universal Broker successfully.

Options - I-Management Console and Configuration File Editable, Recycle Required

I-Management Console and Configuration File Editable, Recycle Required

The following table identifies Universal Broker options that you can modify either by editing the Universal Broker configuration file or via I-Management Console, and for which Universal Broker must be recycled in order for the modifications to take effect. These options are not updated when Universal Broker simply is refreshed.

Windows
If the options are modified via the Universal Configuration Manager, Universal Broker must be recycled.

Option	Description
CA_CERTIFICATES	Path to PEM-formatted trusted CA X.509 certificates.
CERTIFICATE	Path to Broker's PEM-formatted X.509 certificate.
CERTIFICATE_REVOCATION_LIST	Path to PEM-formatted CRL.
COMPONENT_PORT	TCP/IP port used for Broker-Component communications.
PRIVATE_KEY	Path to Broker's PEM formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Broker's PRIVATE_KEY.
SAF_KEY_RING	SAF certificate key ring name.
SAF_KEY_RING_LABEL	SAF certificate key ring label.
SERVICE_BACKLOG	Service interface backlog size for pending connection requests.
SERVICE_IP_ADDRESS	TCP/IP address on which the Broker listens.
SERVICE_PORT	TCP/IP port number on which the Broker listens.
SSL_IMPLEMENTATION	SSL implementation to be used for network configuration.

Options - I-Management Console and Configuration File Editable, Refresh Required

I-Management Console and Configuration File Editable, Refresh Required

The following table identifies Universal Broker options that you can modify by editing the Universal Broker configuration file or via I-Management Console, and for which Universal Broker only needs to be refreshed in order for the modifications to take effect.

- If the options are modified by editing the Universal Broker configuration file, a Universal Control REFRESH command must be issued.
- If the options are modified via I-Management Console, Universal Broker is refreshed automatically.

Windows
If the options are modified via the Universal Configuration Manager, Universal Broker is refreshed automatically.

Option	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
CODE_PAGE	Text translation code page.
CTL_SSL_CIPHER_LIST	SSL cipher list for the control sessions.
DNS_CACHE_TIMEOUT	Time-out for DNS cache.
EVENT_GENERATION	Events to be generated as persistent events.
LOG_DIRECTORY	Log file directory.
LOG_FILE_GENERATIONS	Total number of log files that will be saved within the log directory.
LOG_FILE_LINES	Total number of lines to be written to the log file before the log file is wrapped.
MESSAGE_DESTINATION	Location where messages are written.
MESSAGE_LANGUAGE	Language of messages written.
MESSAGE_LEVEL	Level of messages written.
MONITOR_EVENT_EXPIRATION	Duration of a monitoring event record in the Universal Broker local UES database.
PERSISTENT_EVENT_EXPIRATION	Duration of a persistent event record in the Universal Broker local UES database.

RUNNING_MAX	Maximum number of simultaneous components.
TMP_DIRECTORY	Directory for temporary files.
TRACE_DIRECTORY	Directory for trace files.
TRACE_FILE_LINES	Maximum number of lines written to the trace file.
TRACE_TABLE	Memory trace table specification.
WORKING_DIRECTORY	Broker’s working directory.

Universal Broker Additional Information

Universal Broker Additional Information

The following table identifies and provides links to additional information used by or specific to Universal Broker.

Information	Description
Character Code Pages	Character code pages provided by Stonebranch Inc. for use with Workload Automation components 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

Universal Broker Additional Information - Overview

Universal Broker Additional Information

The following table identifies and provides links to additional information used by or specific to Universal Broker.

Information	Description
Character Code Pages	Character code pages provided by Stonebranch Inc. for use with Workload Automation components 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 - UBROKER

The following table identifies the character code pages provided by Stonebranch Inc. for use with Workload Automation 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						

UTT Files - UBROKER

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 **UNVPRD510/UNVNLS** file. *codepage* is the member name of the UTT file.
z/OS	UTT files are members of the PDS allocated to the Broker ddname **UNVNLS**. *codepage* specifies the member name.
UNIX	UTT files are located in the directory specified by the NLS_DIRECTORY option, which defaults to **/opt/universal/nls**. *codepage* is the base file name of the UTT file.
Windows	UTT files are located in the **NLS** subdirectory of the installation directory. *codepage* is the base file name of the UTT file.
HP NonStop	UTT files are located in the **$SYSTEM.UNVNLS** subvolume. *codepage* is the base file name of the UTT file.