



Universal Broker

Reference Guide

Universal Products
Version 3.2.0

Universal Broker

Reference Guide

Universal Broker 3.2.0

Document Name	Universal Broker 3.2.0 Reference Guide				
Document ID	ub-ref-3205				
Products	z/OS	UNIX	Windows	OS/400	HP NonStop*
Universal Broker	√	√	√	√	√
* Universal Broker 2.1.1 is used on the HP NonStop operating system.					

Stonebranch Documentation Policy

This document contains proprietary information that is protected by copyright. All rights reserved. No part of this publication may be reproduced, transmitted or translated in any form or language or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission, in writing, from the publisher. Requests for permission to make copies of any part of this publication should be mailed to:

Stonebranch, Inc.
950 North Point Parkway, Suite 200
Alpharetta, GA 30005 USA
Tel: (678) 366-7887
Fax: (678) 366-7717

Stonebranch, Inc.® makes no warranty, express or implied, of any kind whatsoever, including any warranty of merchantability or fitness for a particular purpose or use.

The information in this documentation is subject to change without notice.

Stonebranch shall not be liable for any errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this document.

All products mentioned herein are or may be trademarks of their respective owners.

© 2003-2010 by Stonebranch, Inc.

All rights reserved.



Summary of Changes

Changes for Universal Broker 3.2.0 Reference Guide (ub-ref-3205) September 8, 2009

Universal Broker 3.2.0.6

- Added the following configuration options:
 - [LOG_FILE_LINES](#)
 - [UNIX_DB_DATA_SET](#)
 - [UNIX_SPOOL_DATA_SET](#)
- Added zFS file system / data set information in the following configuration options:
 - [MOUNT_POINT](#)
 - [MOUNT_POINT_MODE](#)
- Specified, in [Table 5.2 Universal Broker for z/OS - Batch JCL DD Statements](#), that **UNVDB** and **UNVSPool** ddnames are not used if zFS data sets are used.
- Added the following code pages in Section [5.5 Character Code Pages](#):
 - IBM875
 - IBM4971

Changes for Universal Broker 3.2.0 Reference Guide (ub-ref-3204) July 29, 2009

Universal Broker 3.2.0.1 for OS/400

- Modified document for upgrade from Universal Broker 3.1.1 for OS/400 to Universal Broker 3.2.0 for OS/400, including:
 - Changed the following OS/400 names throughout the document:
 - Universal Broker subsystem name from **UBROKER** to **UNVUBR320**.
 - Universal Broker user profile name from **UBROKER** to **UNVUBR320**.

- Universal Products installation library name from **UNIVERSAL** to **UNVPRD320**.
- Universal Products spool library name from **UNVSPPOOL** to **UNVSPL320**.
- Universal Products temporary directory from **UNVTMP** to **UNVTMP320**.
- Specified the following configuration options for OS/400 in [Chapter 2 Universal Broker Configuration Options](#):
 - **ACTIVITY_MONITORING**
 - **CERTIFICATE_REVOCACTION_LIST**
 - **EVENT_GENERATION**
 - **MONITOR_EVENT_EXPIRATION**
 - **PERSISTENT_EVENT_EXPIRATION**
 - **SERVICE_BACKLOG**
- Specified the following Universal Access Control List entries for OS/400 in [Chapter 4 Universal Broker UACL Entries](#):
 - **EVENT_ACCESS**
 - **REMOTE_CONFIG_ACCESS**

Changes for Universal Broker 3.2.0 Reference Guide (ub-ref-3203) April 1, 2009

- Added DD statement for SAP RFC file used by Universal Connector, as of Universal Connector 3.2.0.1, in [Table 5.2 Universal Broker for z/OS - Batch JCL DD Statements](#).

Changes for Universal Broker 3.2.0 Reference Guide (ub-ref-3202) December 17, 2008

- Modified the description of the **SYSTEM_ID** configuration option.

Changes for Universal Broker 3.2.0 Reference Guide (ub-ref-3201) September 5, 2008

- Added toll-free telephone number for North America in [Appendix A Customer Support](#).

Changes for Universal Broker 3.2.0 Reference Guide (ub-ref-320) May 16, 2008

Universal Broker 3.2.0.6

- Deleted the following specification methods for all configuration options:
 - Command Line, Short Form
 - Command Line, Long Form
 - Environment Variables

- Added the following configuration options in [Chapter 2 Universal Broker Configuration Options](#):
 - `ACTIVITY_MONITORING`
 - `BIF_DIRECTORY`
 - `CERTIFICATE_REVOCACTION_LIST`
 - `EVENT_GENERATION`
 - `MONITOR_EVENT_EXPIRATION`
 - `SAF_KEY_RING`
 - `SAF_KEY_RING_LABEL`
 - `SSL_IMPLEMENTATION`
 - `PERSISTENT_EVENT_EXPIRATION`
 - `SYSTEM_ID`
- Added [Chapter 3 Component Definition Options for Universal Broker](#).
- Added the following UACL entries in [Chapter 4 Universal Broker UACL Entries](#):
 - `EVENT_ACCESS`
 - `REMOTE_CONFIG_ACCESS`

Contents

Summary of Changes	5
Contents	8
List of Tables	11
Preface	12
Document Structure	12
Format	12
Conventions	13
Vendor References	14
Document Organization	15
Chapter 1 Overview	16
1.1 Introduction	16
Chapter 2 Universal BrokerConfiguration Options	17
2.1 Overview	17
2.2 Configuration Options Information	18
2.3 Configuration Options List	20
2.4 ACTIVITY_MONITORING	22
2.5 BIF_DIRECTORY	23
2.6 CA_CERTIFICATES	24
2.7 CERTIFICATE	25
2.8 CERTIFICATE_REVOCATION_LIST	26

2.9	CODE_PAGE	27
2.10	COMPONENT_DIRECTORY	28
2.11	COMPONENT_PORT	29
2.12	CTL_SSL_CIPHER_LIST	30
2.13	DNS_CACHE_TIMEOUT	31
2.14	EVENT_GENERATION	32
2.15	INSTALLATION_DIRECTORY	34
2.16	LOG_DIRECTORY	35
2.17	LOG_FILE_LINES	36
2.18	MESSAGE_DESTINATION	37
2.19	MESSAGE_LANGUAGE	39
2.20	MESSAGE_LEVEL	40
2.21	MONITOR_EVENT_EXPIRATION	42
2.22	MOUNT_POINT	43
2.23	MOUNT_POINT_MODE	44
2.24	NLS_DIRECTORY	46
2.25	PERSISTENT_EVENT_EXPIRATION	47
2.26	PID_FILE_DIRECTORY	48
2.27	PRIVATE_KEY	49
2.28	PRIVATE_KEY_PWD	50
2.29	RUNNING_MAX	51
2.30	SAF_KEY_RING	52
2.31	SAF_KEY_RING_LABEL	53
2.32	SERVICE_BACKLOG	54
2.33	SERVICE_IP_ADDRESS	55
2.34	SERVICE_PORT	56
2.35	SMF_EXIT_LOAD_LIBRARY	57
2.36	SPOOL_DIRECTORY	58
2.37	SSL_IMPLEMENTATION	59
2.38	SYSTEM_ID	60
2.39	TMP_DIRECTORY	61
2.40	TRACE_DIRECTORY	62
2.41	TRACE_FILE_LINES	63
2.42	TRACE_TABLE	64

2.43 UCMD_STC_SUPPORT	65
2.44 UNIX_DB_DATA_SET	66
2.45 UNIX_SPOOL_DATA_SET	67
2.46 WORKING_DIRECTORY	68
Chapter 3 Component Definition Options for Universal Broker	69
3.1 Overview	69
3.2 Component Definition Options Information	70
3.3 Component Definition Options	72
3.4 AUTOMATICALLY_START	73
3.5 COMPONENT_NAME	74
3.6 COMPONENT_TYPE	75
3.7 CONFIGURATION_FILE	76
3.8 RUNNING_MAXIMUM	77
3.9 START_COMMAND	78
3.10 WORKING_DIRECTORY	79
Chapter 4 Universal Broker UACL Entries	80
4.1 Overview	80
4.2 UACL Entries Information	81
4.3 UACL Entries List	83
4.4 UBROKER_ACCESS	84
4.5 CERT_MAP	85
4.6 EVENT_ACCESS	86
4.7 REMOTE_CONFIG_ACCESS	88
Chapter 5 Additional Information for Universal Broker	90
5.1 Overview	90
5.2 DD Statements for JCL	91
5.3 z/OS UNIX Access Permission Modes	92
5.4 SSL Cipher Suites	93
5.5 Character Code Pages	94
5.6 UTT Files	96

List of Tables

Chapter 2 Universal Broker Configuration Options	17
Table 2.1 Universal Broker Configuration Options	21
Table 2.2 SSL Cipher Suites (for CTL_SSL_CIPHER_LIST)	30
Table 2.3 z/OS UNIX Access Permission Modes	44
Chapter 3 Component Definition Options for Universal Broker	69
Table 3.1 Component Definition Options	72
Chapter 4 Universal Broker UACL Entries	80
Table 4.1 Universal Broker UACL Entries	83
Chapter 5 Additional Information for Universal Broker	90
Table 5.1 Universal Broker - Additional Information	90
Table 5.2 Universal Broker for z/OS - Batch JCL DD Statements	91
Table 5.3 z/OS UNIX Access Permission Modes	92
Table 5.4 SSL Cipher Suites	93
Table 5.5 Character Code Pages	95
Table 5.6 UTT File Locations	96

Preface

Document Structure

This document is written using specific conventions for text formatting and according to a specific document structure in order to make it as useful as possible for the largest audience. The following sections describe the document formatting conventions and organization.

Format

Starting with the Universal Products 3.2.0 release, this Universal Broker 3.2.0 Reference Guide serves as a companion document to the newly created Universal Broker 3.2.0 User Guide.

Links to detailed information in this reference guide have been created in the user guide.

In order for the links between these documents to work correctly:

- Place the documents in the same folder.
- In Adobe Reader / Adobe Acrobat, de-select **Open cross-document link in same window** in the **General** category of your **Preferences** dialog (selected from the **Edit** menu).

Conventions

Specific text formatting conventions are used within this document to represent different information. The following conventions are used.

Typeface and Fonts

This document provides tables that identify how information is used. These tables identify values and/or rules that are either pre-defined or user-defined:

- *Italics* denotes user-supplied information.
- **Boldface** indicates pre-defined information.

Elsewhere in this document, **This Font** identifies specific names of different types of information, such as file names or directories (for example, `\abc\123\he1p.txt`).

Operating System-Specific Text

Most of this document describes the product in the context of all supported operating systems. At times, it is necessary to refer to operating system-specific information. This information is introduced with a special header, which is followed by the operating system-specific text in a different font size from the normal text.

z/OS

This text pertains specifically to the z/OS line of operating systems.

This text resumes the information pertaining to all operating systems.

Tips from the Stoneman



Look to the Stoneman for suggestions
or for any other information
that requires special attention.

Stoneman's Tip

Vendor References

References are made throughout this document to a variety of vendor operating systems. We attempt to use the most current product names when referencing vendor software.

The following names are used within this document:

- **z/OS** is synonymous with IBM z/OS and IBM OS/390 line of operating systems.
- **Windows** is synonymous with Microsoft's Windows 2000 / 2003 / 2008, Windows XP, Windows Vista, and Windows 7 lines of operating systems. Any differences between the different systems will be noted.
- **UNIX** is synonymous with operating systems based on AT&T and BSD origins and the Linux operating system.
- **OS/400** is synonymous with IBM OS/400, IBM i/5, and IBM i operating systems.
- **AS/400** is synonymous for IBM AS/400, IBM iSeries, and IBM System i systems.

Note: These names do not imply software support in any manner. For a detailed list of supported operating systems, see the Universal Products 3.2.0 Installation Guide.

Document Organization

This document is organized into the following chapters:

- [Overview](#) (Chapter 1)
Introduction to the reference information in this document.
- [Universal Broker Configuration Options](#) (Chapter 2)
Detailed information about the configuration options used with Universal Broker.
- [Component Definition Options for Universal Broker](#) (Chapter 3)
Detailed information about the component definitions used with Universal Broker.
- [Universal Broker UACL Entries](#) (Chapter 4)
Detailed information about the Universal Access Control List (UACL) entries available for use with Universal Broker.
- [Additional Information for Universal Broker](#) (Chapter 5)
Additional technical information used by or specific to Universal Broker (and all Universal Products).
- [Customer Support](#) (Appendix A)
Customer support contact information for Universal Broker.

Chapter 1 Overview

1.1 Introduction

The Universal Broker 3.2.0 Reference Guide is a companion document to the Universal Broker 3.2.0 User Guide.

It provides technical detail for the information and procedures presented in that document:

- Universal Broker Manager options
- Universal Broker Server options
- Universal Broker component definition options
- Universal Broker UACL entries
- Additional information

Chapter 2

Universal Broker

Configuration Options

2.1 Overview

This chapter provides detailed information on the configuration options available for use with the Universal Broker. Information on how these options are used is documented in the Universal Broker 3.2.0 User Guide.

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

Section [2.2 Configuration Options Information](#) provides a guideline for understanding the information presented on each option.

2.2 Configuration Options Information

For each configuration option, this chapter provides the following information.

Description

Describes the configuration option and how it is used.

Usage

Provides a table of the following information:

Method	Syntax	OS/400	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 (with a ✓) the operating systems for which each method of specifying the option is valid:

- OS/400
- NonStop (HP NonStop)
- UNIX
- Windows
- z/OS

Values

Identifies all possible values for the specified value type.

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

<Additional Information>

Identifies any additional information specific to the option.

2.3 Configuration Options List

Table 2.1 identifies all Universal Broker configuration options.

Option	Description	Page
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.	22
BIF_DIRECTORY	Broker Interface File directory that specifies where Universal Broker will create its interface file.	23
CA_CERTIFICATES	Path to PEM-formatted trusted CA X.509 certificates.	24
CERTIFICATE	Path to Broker's PEM-formatted X.509 certificate.	25
CERTIFICATE_REVOCATION_LIST	Path to PEM-formatted CRL.	26
CODE_PAGE	Text translation code page.	27
COMPONENT_DIRECTORY	Component definition file directory.	28
COMPONENT_PORT	TCP/IP port used for Broker-Component communications.	29
CTL_SSL_CIPHER_LIST	SSL cipher list for the control sessions.	30
DNS_CACHE_TIMEOUT	Time-out for DNS cache.	31
EVENT_GENERATION	Events to be generated as persistent events.	32
INSTALLATION_DIRECTORY	Base directory where product is installed.	34
LOG_DIRECTORY	Log file directory.	35
LOG_FILE_LINES	Total number of lines to be written to the log file before the log file is wrapped.	36
MESSAGE_DESTINATION	Location where messages are written.	37
MESSAGE_LANGUAGE	Language of messages written.	39
MESSAGE_LEVEL	Level of messages written.	40
MONITOR_EVENT_EXPIRATION	Duration of a monitoring event record in the Universal Broker local UES database.	42
MOUNT_POINT	HFS or zFS database mount directory.	43
MOUNT_POINT_MODE	HFS or zFS permission mode for MOUNT_POINT.	44
NLS_DIRECTORY	UMC and UTT file directory.	46
PERSISTENT_EVENT_EXPIRATION	Duration of a persistent event record in the Universal Broker local UES database.	47
PID_FILE_DIRECTORY	PID file location.	48
PRIVATE_KEY	Path to Broker's PEM formatted RSA private key.	49
PRIVATE_KEY_PWD	Password for the Broker's PRIVATE_KEY.	50
RUNNING_MAX	Maximum number of simultaneous components.	51
SAF_KEY_RING	SAF certificate key ring name.	52
SAF_KEY_RING_LABEL	SAF certificate key ring label.	53
SERVICE_BACKLOG	Service interface backlog size for pending connection requests.	54
SERVICE_IP_ADDRESS	TCP/IP address on which the Broker listens.	55

Option	Description	Page
SERVICE_PORT	TCP/IP port number on which the Broker listens.	56
SMF_EXIT_LOAD_LIBRARY	UNVACTRT SMF exit load library.	57
SPOOL_DIRECTORY	Spool file directory.	58
SSL_IMPLEMENTATION	SSL implementation to be used for network configuration.	59
SYSTEM_ID	Universal Broker running on a system (O/S image).	60
TMP_DIRECTORY	Directory for temporary files.	61
TRACE_DIRECTORY	Directory for trace files.	62
TRACE_FILE_LINES	Maximum number of lines written to the trace file.	63
TRACE_TABLE	Memory trace table specification.	64
UCMD_STC_SUPPORT	Support for Universal Command started tasks.	65
UNIX_DB_DATA_SET	HFS or zFS data set used for the Universal Broker's databases.	66
UNIX_SPOOL_DATA_SET	HFS or zFS data set used for the Universal Broker's spool.	67
WORKING_DIRECTORY	Universal Broker's working directory.	68

Table 2.1 Universal Broker Configuration Options

2.4 ACTIVITY_MONITORING

Description

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

Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	activity_monitoring <i>option</i>	√		√	√	√

Values

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

Valid values for *option* are:

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

[Default is yes).]

2.5 BIF_DIRECTORY

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<code>bif_directory <i>directory</i></code>			✓		

Values

directory is the name of the BIF directory.

[Default is `/var/opt/universa1.`]

2.6 CA_CERTIFICATES

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	ca_certificates <i>ddname</i> or <i>file</i>	✓		✓	✓	✓

Values

z/OS

ddname is the ddname of the X.509 certificates. The value is used only when the [SSL_IMPLEMENTATION](#) option is set to *OPENSSL*.

Allocated to the ddname must be either a sequential data set or a member of a PDS that has a variable record format.

UNIX and Windows

file is the path name of the X.509 certificates file. Relative paths are relative the current working directory.

OS/400

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.

2.7 CERTIFICATE

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	certificate <i>ddname</i> or <i>file</i>	✓		✓	✓	✓

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 and Windows

file is the path name of the X.509 certificate file. Relative paths are relative to the current working directory.

OS/400

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.

2.8 CERTIFICATE_REVOCATION_LIST

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<i>crl file or ddname</i>	√		√	√	√

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 and Windows

file is the path name of the file containing the CRL. Relative paths are relative to the current working directory.

OS/400

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.

2.9 CODE_PAGE

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	codepage <i>codepage</i>	✓	✓	✓	✓	✓

Value

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

codepage references a Universal Translate Table (UTT) file provided with the product (see Section [5.6 UTT Files](#) for information on UTT files). UTT files are used to translate between Unicode and the local single-byte code page. (All UTT files end with an extension of `.utt`.)

Default

The default code page is different for different operating systems:

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

See Section [5.5 Character Code Pages](#) for a complete list of character code pages provided by Stonebranch Inc. for use with Universal Products.

2.10 COMPONENT_DIRECTORY

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	component_directory <i>directory</i>			✓		

Value

directory is the name of the directory.

Relative path names are relative to the installation directory.

[Default is /etc/universal/comp.]

2.11 COMPONENT_PORT

Description

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

Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	component_port <i>port</i>				√	

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.

2.12 CTL_SSL_CIPHER_LIST

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	ctl_ssl_cipher_list <i>cipherlist</i>	√		√	√	√

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.

Table 2.2 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.

Table 2.2 SSL Cipher Suites (for CTL_SSL_CIPHER_LIST)

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

2.13 DNS_CACHE_TIMEOUT

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	dns_cache_timeout <i>seconds</i>	✓		✓	✓	✓

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.]

2.14 EVENT_GENERATION

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 Universal Products components, see the Universal Event Subsystem 3.2.0 Event Definitions Guide.)

Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	event_generation <i>types</i>	√		√	√	√

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).]

2.15 INSTALLATION_DIRECTORY

Description

The `INSTALLATION_DIRECTORY` option specifies the Universal Broker base installation directory.

Note: This is a required option.

Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<code>installation_directory <i>directory</i></code>		✓	✓	✓	

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.

2.16 LOG_DIRECTORY

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	log_directory <i>directory</i>			✓	✓	

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.

Defaults

UNIX

[Default is /var/opt/universa1/log.]

Windows

[Default is log.]

2.17 LOG_FILE_LINES

Description

The LOG_FILE_LINES option specifies the total number of lines to be written to the log file before the log file is wrapped.

Log file creation is specified by the [MESSAGE_DESTINATION](#) option.

Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	log_file_lines <i>lines</i>			✓		

Value

lines is the total number of lines to be written to the log file before the log file is wrapped.

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

[Default is 2000.]

2.18 MESSAGE_DESTINATION

Description

The MESSAGE_DESTINATION option specifies the location where messages are written.

Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	message_dest <i>destination</i>	✓	✓	✓	✓	✓

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. Currently, 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.]**

OS/400

- **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, 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.]**

2.19 MESSAGE_LANGUAGE

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	language <i>language</i>	✓	✓	✓	✓	✓

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.

OS/400

UMC file members are located in the physical source file UNVPRD320/UNVNLS.

HP NonStop

UMC files are located in subvolume \$SYSTEM.UNVNLS.

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

2.20 MESSAGE_LEVEL

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax *	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	message_level <i>level</i>	√	√	√	√	√

Values

level is the level of messages to write.

Valid values for *level* are:

- **trace**
Writes trace messages used for diagnostic purposes (see Section [Trace Files](#)).
Note: Use **trace** only as directed by Stonebranch, Inc. [Customer Support](#).
- **audit**
Writes audit, informational, warning, and error messages.
- **info**
Writes informational, warning, and error messages.
- **warn**
Writes warning and error messages.
- **error**
Writes error messages only.

[Default is info.]

Trace Files

OS/400

The trace file name is **UNVTMP320/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 $\text{cylCount} = (\text{totalSize} / 16) / 750000$.

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.

2.21 MONITOR_EVENT_EXPIRATION

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	monitor_event_expiration <i>seconds</i>	√		√	√	√

Values

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

[Default is 600 (10 minutes).]

2.22 MOUNT_POINT

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	mount_point <i>directory</i>					√

Values

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

[Default is /tmp.]

2.23 MOUNT_POINT_MODE

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<code>mount_point_mode mode</code>					✓

Values

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

Table 2.3, below, 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.

Table 2.3 z/OS UNIX Access Permission Modes

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]**

2.24 NLS_DIRECTORY

Description

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

Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	nls_directory <i>directory</i>			✓	✓	

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.`]

2.25 PERSISTENT_EVENT_EXPIRATION

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 UEC within this time period, Universal Broker will delete the record from the local UES database. (A persistent event record is saved in a Universal Enterprise Controller (UEC) database for long-term storage.)

Note: Events are identified as persistent events via the [EVENT_GENERATION](#) option.

Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<code>persistent_event_expiration seconds</code>	√		√	√	√

Values

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

[Default is 172800 (2 days).]

2.26 PID_FILE_DIRECTORY

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	pid_file_directory <i>directory</i>			✓		

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/universa1.]

Note: If the default value is changed, the PID file directory location in the Universal Broker startup script requires the same value. See Section [5.1.2 Daemon](#) in the Universal Broker 3.2.0 User Guide for details on the Broker startup script.

2.27 PRIVATE_KEY

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	private_key <i>ddname</i> or <i>file</i>	√		√	√	√

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 and Windows

file is the path of the PEM-formatted RSA private key file that corresponds to the X.509 certificates.

OS/400

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.

2.28 PRIVATE_KEY_PWD

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<code>private_key_password password</code>	✓		✓	✓	✓

Values

password is the password for the private key.

2.29 RUNNING_MAX

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	running_max <i>maximum</i>	✓		✓	✓	✓

Values

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

[Default is 100.]

2.30 SAF_KEY_RING

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 use 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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	saf_key_ring <i>name</i>					√

Values

name is the name of the SAF certificate key ring.

2.31 SAF_KEY_RING_LABEL

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	saf_key_ring_label <i>label</i>					√

Values

label is the label of the SAF certificate key ring.

[Default is the default certificate in the key ring.]

2.32 SERVICE_BACKLOG

Description

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

Usage

Method	Syntax	OS/400	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.

2.33 SERVICE_IP_ADDRESS

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	service_ip_address <i>ipaddress</i>	√	√	√	√	√

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 *.]

2.34 SERVICE_PORT

Description

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

Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	service_port <i>port</i>	✓	✓	✓	✓	✓

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.

2.35 SMF_EXIT_LOAD_LIBRARY

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 Universal Products 3.2.0 Installation Guide 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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	smf_exit_load_library <i>dsn</i>					√

Values

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

2.36 SPOOL_DIRECTORY

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	spool_directory <i>directory</i>			✓	✓	

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.

Windows

[Default is C:\Program Files\Universal\spool.]

UNIX

[Default is /var/opt/universal/spool.]

2.37 SSL_IMPLEMENTATION

Description

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

Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	ssl_implementation <i>option</i>					√

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 Universal Products 3.2.0 Installation Guide for a description of the prerequisites before using System SSL.)

[Default is *OPENSSL*.]

2.38 SYSTEM_ID

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	system_id <i>ID</i>					√

Values

ID is the system identifier of the local Universal Broker.

ID is a value 1-8 characters in length.

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

[Default is a blank value.]

2.39 TMP_DIRECTORY

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	tmp_directory <i>directory</i>			√	√	√

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.`]

2.40 TRACE_DIRECTORY

Description

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

Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	trace_directory <i>directory</i>			✓	✓	

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.

Windows

[Default is C:\Program Files\Universal\UBroker.]

UNIX

[Default is /var/opt/universal/trace.]

2.41 TRACE_FILE_LINES

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.)

OS/400

Trace file records are 366 bytes long.

Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	trace_file_lines <i>lines</i>	✓		✓	✓	✓

Values

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

[Default is 500,000.]

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

OS/400

If space is limited in the trace file ASP (ASP in which the UNVTMP320 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 UNVTMP320/UNVTRCUBR and increase this setting. The largest value allowed without increasing the number of records allowed is 509000.

2.42 TRACE_TABLE

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	OS/400	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.

2.43 UCMD_STC_SUPPORT

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	ucmd_stc_support <i>option</i>					√

Values

opt 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 *opt* is *NO* is specified, 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.]

2.44 UNIX_DB_DATA_SET

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	unix_db_data_set <i>DSN</i>					√

Values

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

2.45 UNIX_SPOOL_DATA_SET

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	unix_spool_data_set <i>DSN</i>					√

Values

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

2.46 WORKING_DIRECTORY

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	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<code>working_directory <i>directory</i></code>			✓	✓	

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.

Windows

[Default is the Universal Broker installation directory.]

UNIX

[Default is the startup directory.]

Chapter 3

Component Definition Options for Universal Broker

3.1 Overview

This chapter provides detailed information about the options that comprise the Universal Products component definitions provided to the Universal Broker.

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

Information on how component definitions are used is documented in the Universal Broker 3.2.0 User Guide.

Section [3.2 Component Definition Options Information](#) provides a guideline for understanding the information presented, in this chapter, for each component definition option.

3.2 Component Definition Options Information

For each component definition option, this chapter provides the following information.

Description

Describes the option and how it is used.

Usage

Provides a table of the following information:

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Component Definition Keyword	<Format / Value>					

Method

Identifies the method used for specifying a Universal Products component definition option:

- Component Definition Keyword

Syntax

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

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

(Operating System)

Identifies (with a ✓) the operating systems for which the method of specifying the option is valid:

- OS/400
- HP NonStop
- UNIX
- Windows
- z/OS

Values

Identifies all possible values for the specified value type.

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

3.3 Component Definition Options

Table 3.1 identifies all of the options that can comprise a component definition provided to Universal Broker.

Option	Description	Page
AUTOMATICALLY_START	Specification for whether the component automatically starts by the Universal Broker at start-up time or only on demand.	73
COMPONENT_NAME	Name by which clients know the component.	74
COMPONENT_TYPE	Type of component.	75
CONFIGURATION_FILE *	Component's configuration file name.	76
RUNNING_MAXIMUM	Maximum number of this component that can run simultaneously.	77
START_COMMAND *	Component program name.	78
WORKING_DIRECTORY *	Path used as the working directory of the component.	79
* These options are required in all component definitions.		

Table 3.1 Component Definition Options

3.4 AUTOMATICALLY_START

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	OS/400	NonStop	UNIX	Windows	z/OS
Component Definition Keyword	<code>auto_start option</code>	√	√	√	√	√

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.]

3.5 COMPONENT_NAME

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	OS/400	NonStop	UNIX	Windows	z/OS
Component Definition Keyword	component_name <i>name</i>	√	√	√	√	√

Values

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

3.6 COMPONENT_TYPE

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	OS/400	NonStop	UNIX	Windows	z/OS
Component Definition Keyword	component_type <i>type</i>	✓		✓	✓	✓

Values

type is the type of component.

3.7 CONFIGURATION_FILE

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.

OS/400

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	OS/400	NonStop	UNIX	Windows	z/OS
Component Definition Keyword	configuration_file <i>member</i> or configuration_file <i>filename</i>	√	√	√	√	√

Values

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

3.8 RUNNING_MAXIMUM

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	OS/400	NonStop	UNIX	Windows	z/OS
Component Definition Keyword	running_max <i>maximum</i>	√	√	√	√	√

Values

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

[Default is 100.]

3.9 START_COMMAND

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.

OS/400

Non-qualified program names are located in the library list *LIBL.

Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Component Definition Keyword	start_command <i>member</i> or start_command <i>name[options]</i>	√	√	√	√	√

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.

3.10 WORKING_DIRECTORY

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	OS/400	NonStop	UNIX	Windows	z/OS
Component Definition Keyword	working_directory <i>directory</i>	✓	✓	✓	✓	✓

Values

directory is the full path name of the working directory.

[Default is (.).

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.

OS/400 and 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.

OS/400

working_directory serves as a required placeholder only. Do not change its value.

Chapter 4

Universal Broker

UACL Entries

4.1 Overview

This chapter provides 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 Universal Broker 3.2.0 User Guide.

Section [4.2 UACL Entries Information](#) provides a guideline for understanding the information presented for each UACL entry.

4.2 UACL Entries Information

For each UACL entry, this chapter provides the following information.

Description

Describes the UACL entry and how it is used.

Usage

Provides a table of the following information:

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
UACL File Keyword	<Type / Rule>					

Method

Identifies the method used for specifying a UACL entry:

- UACL File Keyword

Syntax

Identifies the syntax of the method used for a UACL entry:

- Type Universal Products 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 (with a ✓) the operating systems for which the method of specifying the UACL entry is valid:

- OS/400
- HP NonStop
- UNIX
- Windows
- z/OS

Values

Identifies all possible values for the fields in a UACL entry rule.

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

4.3 UACL Entries List

Table 4.1 identifies all Universal Broker UACL Entries.

UACL Entry	Description	Page
UBROKER_ACCESS	Allows or denies a Universal Products component access to Universal Broker services.	84
CERT_MAP	Maps a client X.509 certificate to certificate identifier.	85
EVENT_ACCESS	Controls which Universal Enterprise Controller has read and delete access to the Universal Event Subsystem event data maintained by the Universal Broker.	86
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: <ul style="list-style-type: none">• remote_config_access• remote_config_cert_access	86

Table 4.1 Universal Broker UACL Entries

4.4 UBROKER_ACCESS

Description

A UBROKER_ACCESS UACL entry specifies whether to allow or deny a Universal Products 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	OS/400	NonStop	UNIX	Windows	z/OS
UACL File Keyword	ubroker_access <i>host,access</i>	√	√	√	√	√

Values

host specifies an IP address of a Universal Products component.

(See Section [2.8.2 UACL Entries](#) in the Universal Broker 3.2.0 User Guide 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 to accept all connections.]

4.5 CERT_MAP

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	OS/400	NonStop	UNIX	Windows	z/OS
UACL File Keyword	cert_map <i>id=certid,cert-field(s)</i>	✓		✓	✓	✓

Values

id is the certificate identifier.

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

(See Section [2.10 X.509 Certificates](#) in the Universal Broker 3.2.0 User Guide for a detail discussion on the *cert-fields* values.)

4.6 EVENT_ACCESS

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	OS/400	NonStop	UNIX	Windows	z/OS
UACL File Keyword	<code>event_access</code> <code>host,remote_user,read_access,</code> <code>delete_access</code> <code>event_cert_access</code> <code>certid,read_access,delete_access</code>	✓		✓	✓	✓

Values

host specifies an IP address of a Universal Products component.

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

(See [Table 2.8 Client IP Address - Matching Criteria](#) in the Universal Broker 3.2.0 User Guide 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.

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
```

Defaults

```
event_access ALL,*,allow,deny
```

```
event_cert_access *,allow,deny
```

4.7 REMOTE_CONFIG_ACCESS

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	OS/400	NonStop	UNIX	Windows	z/OS
UACL File Keyword	<code>remote_config_access</code> <code>host,remote_user,update_access,</code> <code>control_access</code> <code>remote_config_cert_access</code> <code>certid,update_access,control_access</code>	✓		✓	✓	✓

Values

host specifies an IP address of a Universal Products component.

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

(See [Table 2.8 Client IP Address - Matching Criteria](#) in the Universal Broker 3.2.0 User Guide 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.

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
```

Defaults

```
remote_config_access ALL,*,deny,deny
```

```
remote_config_cert_access *,deny,deny
```

Chapter 5

Additional Information for Universal Broker

5.1 Overview

This chapter provides additional information used by or specific to Universal Broker.

[Table 5.1](#) identifies the types of additional information in this chapter and provides a link to each section.

Information	Description	Page
DD statements	DD statements used in the Universal Broker for z/OS batch JCL.	91
z/OS UNIX Access Permission Modes	z/OS UNIX access permission modes specified via the MOUNT_POINT_MODE configuration option.	92
SSL cipher suites	SSL cipher suites supported by Stonebranch, Inc. for use with Universal Broker.	93
Character Code Pages	Character code pages provided by Stonebranch Inc. for use with Universal Products on each supported operating system.	95
UTT Files	Universal Translate Table (UTT) files are used to translate between Unicode and the local single-byte code page	96

Table 5.1 Universal Broker - Additional Information

5.2 DD Statements for JCL

Table 5.2 describes the DD statements used in the Universal Broker for z/OS batch JCL.

DD name	DCB Attributes	Mode	Description
STEPLIB	DSORG=PO, RECFM=U	input	Universal Products load library containing the program being executed.
UNVCONF	DSORG=PS, RECFM=(F, FB, V, VB)	input	Universal Broker configuration member.
UNVRFC	DSORG=PS, RECFM=(F, FB, V, VB)	input	SAP RFC file used by Universal Connector.
UNVACL	DSORG=PS, RECFM=(F, FB, V, VB)	input	Universal Access Control List configuration member.
UNVCOMP	DSORG=PO, RECFM=(F, FB, V, VB)	input	Universal Broker component definition PDS.
UNVNLS	DSORG=PO, RECFM=(F, FB, V, VB)	input	Universal Products national language support library. Contains message catalogs and code page translation tables.
UNVCREP	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.
UNVPOOL	DSNTYPE=HFS	input, output	Universal Products 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	Universal Products 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	Universal Products 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 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.

Table 5.2 Universal Broker for z/OS - Batch JCL DD Statements

5.3 z/OS UNIX Access Permission Modes

Table 5.3 describes the z/OS UNIX access permission modes, which are specified via the `MOUNT_POINT_MODE` configuration option.

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.

Table 5.3 z/OS UNIX Access Permission Modes

5.4 SSL Cipher Suites

Table 5.4 identifies all of SSL cipher suites supported by Stonebranch Inc. for use with Universal Broker.

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.

Table 5.4 SSL Cipher Suites

5.5 Character Code Pages

Table 5.5 identifies the character code pages provided by Stonebranch Inc. for use with Universal Products on each supported operating system.

Code Page	CCSID	z/OS	UNIX	Windows	OS/400		HP NonStop
					HFS	LIB	
IBM037	037	✓			✓	✓	
IBM273	273	✓			✓	✓	
IBM277	277	✓			✓	✓	
IBM278	278	✓			✓	✓	
IBM280	280	✓			✓	✓	
IBM284	284	✓			✓	✓	
IBM500	500	✓			✓	✓	
IBM875	875	✓					
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			✓	✓		

Code Page	CCSID	z/OS	UNIX	Windows	OS/400		HP NonStop
					HFS	LIB	
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			√	√		

Table 5.5 Character Code Pages

5.6 UTT Files

[Table 5.6](#) 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
OS/400	UTT files are located in the UNVPRD320/UNVNLS file. <i>codepage</i> is the member name of the UTT file.
z/OS	UTT files are members of the PDS allocated to the Broker ddname UNVNLS . <i>codepage</i> specifies the member name.
UNIX	UTT files are located in the directory specified by the NLS_DIRECTORY option, which defaults to /opt/universal/nls . <i>codepage</i> is the base file name of the UTT file.
Windows	UTT files are located in the NLS subdirectory of the installation directory. <i>codepage</i> is the base file name of the UTT file.
HP NonStop	UTT files are located in the \$SYSTEM.UNVNLS subvolume. <i>codepage</i> is the base file name of the UTT file.

Table 5.6 UTT File Locations

Appendix A

Customer Support

Stonebranch, Inc. provides customer support, via telephone and e-mail, for Universal Broker and all Universal Products.

TELEPHONE

Customer support via telephone is available 24 hours per day, 7 days per week.

North America

(+1) 678 366-7887, extension 6

(+1) 877 366-7887, extension 6 [toll-free]

Europe

+49 (0) 700 5566 7887

E-MAIL

All Locations

support@stonebranch.com

Customer support contact via e-mail also can be made via the Stonebranch website:

www.stonebranch.com



**950 North Point Parkway, Suite 200
Alpharetta, Georgia 30005
U.S.A.**

