Stonebranch Solutions

Version 4.2.0

Stonebranch Solutions Utilities

Reference Guide

util-ref-4200

stonebranch

Stonebranch Solutions Utilities

Reference Guide

Stonebranch Solutions 4.2.0

Document Name	Stonebranch Solutions Utilities 4.2.0 Reference Guide				
Document ID	**util-ref-4200**				
Components	**z/OS**	**UNIX**	**Windows**	**IBM i**	**HP NonStop***
Universal Certificate	√	√	√		
Universal Control	√	√	√	√	√
Universal Copy		√	√	√	√
Universal Database Dump	√	√	√		
Universal Database Load	√	√	√		
Universal Display Log File				√	
Universal Encrypt	√	√	√	√	√
Universal Event Log Dump			√		
Universal Message Translator	√	√	√	√	√
Universal Query	√	√	√	√	√
Universal Spool List	√	√	√	√	
Universal Spool Remove	√	√	√	√	
Universal Submit Job				√	
Universal WTO	√				
* Utilities for Universal Products 2.1.1 is used on the HP NonStop operating system.					

Stonebranch Documentation Policy

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.

SAP Certified Integration

Summary of Changes

Changes for Stonebranch Solutions Utilities 4.2.0 Reference Guide (util-ref-4200) August 6, 2010

Stonebranch Solutions Utilities 4.2.0

- Moved information from the Indesca / Infitran Utilities 4.1.0 User Guide into this Stonebranch Solutions Utilities 4.2.0 Reference Guide.

  Information on component features and examples was moved to the Indesca and Infitran 4.2.0 User Guides.

Changes for Indesca / Infitran Utilities 4.1.0 Reference Guide (util-ref-4100) February 10, 2010

Universal Submit Job 4.1.0.0

- Added Section 33.2.3 Universal Command Server Options Affecting USBMJOB.

Universal Database Dump 4.1.0.0

- Replaced RECOVER option with DUMP_OPTIONS option in Chapter 12 Universal Database Dump Configuration Options.
- Modified OVERWRITE option in Chapter 14 Universal Database Load Configuration Options.

Changes for Universal Products Utilities 3.2.0 Reference Guide (util-ref-3204) September 8, 2009

Universal Products 3.2.0.8

- Added the following code pages in Section 36.3 Character Code Pages:
  - IBM875
  - IBM4971

Changes for Universal Products Utilities 3.2.0 Reference Guide (util-ref-3203) July 29, 2009

Universal Products Utilities 3.2.0.1 for OS/400

- Modified document for upgrade from Universal Products Utilities 3.1.1 for OS/400 to Universal Products Utilities 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 **UNVSPOOL** to **UNVSPL320**.
    - Universal Products temporary directory from **UNVTMP** to **UNVTMP320**.

Universal Control 3.2.0.1 for OS/400

- Specified the following configuration option for OS/400 in Chapter 5 Universal Control Manager Configuration Options:
  - ACTIVITY_MONITORING
  - CERTIFICATE_REVOCATION_LIST
  - EVENT_GENERATION
  - OUTBOUND_IP

Universal Query 3.2.0.1 for OS/400

- Specified the following configuration options for OS/400 in Chapter 26 Universal Query Configuration Options:
  - COMMAND_ID
  - COMPONENT_ID
  - MANAGERS
  - PLF_DIRECTORY

Universal Spool 3.2.0.1 for OS/400

- Specified the following configuration options for OS/400 in Chapter 29 Universal Spool List Configuration Options:
  - COMPONENT
  - ID
  - LIST
  - MESSAGE_LEVEL
  - VERSION
- Specified the following configuration options for OS/400 in Chapter 31 Universal Spool Remove Configuration Options:
  - COMPONENT
  - MESSAGE_LEVEL
  - VERSION

Universal Submit Job 3.2.0.1 for OS/400

- Added the following SBMJOB Encapsulated configuration option in Chapter 33 Universal Submit Job Configuration Options:
  - INLASPGRP

Changes for Universal Products Utilities 3.2.0 Reference Guide (util-ref-3202) December 17, 2008

- Changed the name of the environment variable for the Universal Control Manager SYSTEM_ID configuration option from **`UCTLSYSTEM`** to **`UCTLSYSTEMID`**.
- Changed the name of the environment variable for the Universal Query SYSTEM_ID configuration option from **`UQRYSYSTEM`** to **`UQRYSYSTEMID`**.

Changes for Universal Products Utilities 3.2.0 Reference Guide (util-ref-3201) September 5, 2008

- Added toll-free telephone number for North America in Appendix A Customer Support.

Changes for Universal Products Utilities 3.2.0 Reference Guide
(util-ref-320)
May 16, 2008

Universal Products 3.2.0.0

- Added the following chapters:
  - 12 Universal Database Dump Configuration Options
  - 14 Universal Database Load Configuration Options
  - 24 Universal Products Install Merge Configuration Options
- Added the following configuration options in 3 Universal Certificate Configuration Options:
  - CERT_DB
  - CRL_FILE
  - CRL_FORMAT
  - NEXT_UPDATE_DAYS
  - NEXT_UPDATE_HOURS
  - REVOKE
  - REVOKE_REASON
  - STATE
  - VERIFY
  - TRANSPORT_FILE
  - TRANSPORT_FILE_PWD
- Added the following configuration options in 20 Universal Event Log Dump Configuration Options:
  - INSTALLATION_DIRECTORY
  - LOG_DIRECTORY
  - NLS_DIRECTORY
- Added the following configuration options in 26 Universal Query Configuration Options:
  - BIF_DIRECTORY
  - COMMAND_ID
  - COMPONENT_ID
  - MANAGERS
  - NLS_ DIRECTORY
  - PLF_DIRECTORY
  - SYSTEM_ID

Contents

List of Figures

List of Tables

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.

Cross-Reference Links

This document contains cross-reference links to and from the Indesca User Guide.

In order for the links 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\help.txt`**).

Command Line Syntax Diagrams

Command line syntax diagrams use the following conventions:

Convention	Description
`bold monospace font`	Specifies values to be typed verbatim, such as file / data set names.
`italic monospace font`	Specifies values to be supplied by the user.
`[]`	Encloses configuration options or values that are optional.
`{}`	Encloses configuration options or values of which one must be chosen.
`|`	Separates a list of possible choices.
`. . .`	Specifies that the previous item may be repeated one or more times.
BOLD UPPER CASE	Specifies a group of options or values that are defined elsewhere.

Table P.1 Command Line Syntax

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.
- **IBM i** is synonymous with IBM i/5, IBM OS/400, and OS/400 operating systems.
- **IBM System i** is synonymous with IBM i Power Systems, IBM iSeries, IBM AS/400, and AS/400 systems.

Note: These names do not imply software support in any manner. For a detailed list of supported operating systems, see the Stonebranch Solutions 4.2.0 Installation Guide.

Document Organization

The document is organized into the following chapters:

- Overview of Stonebranch Solutions Utilities (Chapter 1)
  Introduction to the reference information in this document.
- Universal Certificate (Chapter 2)
  Detailed information on Universal Certificate for all operating systems.
- Universal Certificate Configuration Options (Chapter 3)
  Detailed information on all Universal Certificate configuration options for all operating systems.
- Universal Control (Chapter 4)
  Detailed information on Universal Control for all operating systems.
- Universal Control Manager Configuration Options (Chapter 5)
  Detailed information on all Universal Control Manager configuration options for all operating systems.
- Universal Control Server Configuration Options (Chapter 6)
  Detailed information on all Universal Control Server configuration options for all operating systems.
- Universal Control Component Definition Options (Chapter 7)
  Detailed information on all Universal Control component definition options.
- Universal Control UACL Entries (Chapter 8)
  Detailed information on all Universal Access Control List (UACL) entries.
- Universal Copy (Chapter 9)
  Detailed information on Universal Copy for all operating systems.
- Universal Copy Configuration Options (Chapter 10)
  Detailed information on all Universal Copy configuration options for all operating systems.
- Universal Database Dump (Chapter 11)
  Detailed information on Universal Database Dump for all operating systems.
- Universal Database Dump Configuration Options (Chapter 12)
  Detailed information on all Universal Database Dump configuration options for all operating systems.
- Universal Database Load (Chapter 13)
  Detailed information on Universal Database Load for all operating systems.
- Universal Database Load Configuration Options (Chapter 14)
  Detailed information on all Universal Database Load configuration options for all operating systems.
- Universal Display Log File (Chapter 15)
  Detailed information on Universal Display Log File for all operating systems.
- Universal Display Log File Configuration Options (Chapter 16)
  Detailed information on all Universal Display Log File configuration options for all operating systems.
- Universal Encrypt (Chapter 17)
  Detailed information on Universal Encrypt for all operating systems.

- Universal Encrypt Configuration Options (Chapter 18)
  Detailed information on all Universal Encrypt configuration options for all operating systems.
- Universal Event Log Dump (Chapter 19)
  Detailed information on Universal Event Log Dump for all operating systems.
- Universal Event Log Dump Configuration Options (Chapter 20)
  Detailed information on all Universal Event Log Dump configuration options for all operating systems.
- Universal Products Install Merge (Chapter 21)
  Detailed information on Universal Products Install Merge for all operating systems.
- Universal Products Install Merge Configuration Options (Chapter 22)
  Detailed information on all Universal Products Install Merge configuration options for all operating systems.
- Universal Message Translator (Chapter 23)
  Detailed information on Universal Message Translator for all operating systems.
- Universal Message Translator Configuration Options (Chapter 24)
  Detailed information on all Universal Message Translator configuration options for all operating systems.
- Universal Query (Chapter 25)
  Detailed information on Universal Query for all operating systems.
- Universal Query Configuration Options (Chapter 26)
  Detailed information on all Universal Query configuration options for all operating systems.
- Universal Return Code (Chapter 27)
  Detailed information on Universal Return Code for all operating systems.
- Universal Spool List (Chapter 28)
  Detailed information on Universal Spool for all operating systems.
- Universal Spool List Configuration Options (Chapter 29)
  Detailed information on all Universal Spool configuration options for all operating systems.
- Universal Spool Remove (Chapter 30)
  Detailed information on Universal Spool Remove for all operating systems.
- Universal Spool Remove Configuration Options (Chapter 31)
  Detailed information on all Universal Spool Remove configuration options for all operating systems.
- Universal Submit Job (Chapter 32)
  Detailed information on Universal Submit Job for all operating systems.
- Universal Submit Job Configuration Options (Chapter 33)
  Detailed information on all Universal Submit Job configuration options for all operating systems.
- Universal Write-to-Operator (Chapter 34)
  Detailed information on Universal Write-to-Operator for all operating systems.
- Universal Write-to-Operator Configuration Options (Chapter 35)
  Detailed information on all Universal Write-to-Operator configuration options for all operating systems.
- Additional Information (Chapter 36)
  Additional information related to Stonebranch Solutions Utilities.

- Customer Support (Appendix A)
  Customer support contact information for Stonebranch Solutions Utilities.

CHAPTER 1

Overview of Stonebranch Solutions Utilities

This Reference Guide provides detailed technical information for the Stonebranch Solutions utilities, which are installed as part of each Stonebranch Solutions package.

There is a separate chapter for each utility, and a corresponding chapter that provides detailed information on the configuration options of each utility.

Some utilities are operating-system specific; they cannot be used on any operating system. The individual chapters in this document identify the operating systems on which they can be used.

1.1 Types of Stonebranch Solutions Utilities

Table 1.1, below, provides a description of each Stonebranch Solutions utility.

Each Utility Name in the table is a link to its chapter in this document.

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

Table 1.1 Stonebranch Solutions Utilities

1.2 Utilities Configuration

Configuration consists of:

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

Configuration options are read from the following sources:

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

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

Detailed information on these methods of configuration can be found in the Configuration Management chapter of the Indesca and Infitran 4.2.0 User Guides.

Configuration File

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

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

1.3 Configuration Options Information

For each utility option chapter, a configuration option chapter provides the following information.

Description

Describes the configuration option and how it is used.

Usage

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

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

Method

Identifies the different methods used to specify Stonebranch Solutions Utilities configuration options:

- Command Line Option, Short Form
- Command Line Option, Long Form
- Environment Variable
- Configuration File Keyword
- <IBM i> Parameter

Note: Each option can be specified using one or more methods.

Syntax

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

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

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

(Operating System)

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

- IBM i
- 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.

CHAPTER 2

Universal Certificate

2.1 Overview

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

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

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

See the Indesca or Infitran User Guide for an introduction to X.509 certificates and how they are used by Stonebranch Solutions components.

2.2 Usage

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

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

The following sections describe each of these operations.

2.2.1 Certificate

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

A certificate is created using the following input:

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

2.2.2 Certificate Requests

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

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

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

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

2.2.3 Certificate Revocation List

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

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

2.2.4 Transport Files

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

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

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

2.2.5 Printing

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

2.2.6 Verification

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

The certificate process consists of

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

2.2.7 File Formats

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

The following file formats are supported:

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

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

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

z/OS

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

2.2.8 Universal Certificate Database

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

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

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

z/OS

The database is allocated to ddname **UNVDB**. The database allocation attributes are DSORG=PS, RECFM={FB | F}, and LRECL=1120. The block size must be a multiple of LRECL if RECFM is FB.

UNIX and Windows

The database file name is specified with the CERT_DB option. If CERT_DB is not used, the database is created in the current working directory with name **`ucert.db`**.

2.3 Configuration Options

Table 2.1, below, identifies the Universal Certificate for configuration options for the UNIX, Windows, and z/OS operating systems.

Each **Option Name** is a link to detailed information about that configuration option.

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

Option Name	Description
PRINT	Specification that UCERT is to print a certificate request or a certificate.
PRIVATE_KEY_FILE	File name from which the RSA private key is read or to which the RSA private key is written.
PRIVATE_KEY_FORMAT	Format of the private key file specified by PRIVATE_KEY_FILE.
PRIVATE_KEY_PWD	Password used to read and write the private key file specified by PRIVATE_KEY_FILE.
REQUEST_FILE	File name from which the certificate request is read or to which the certificate request is written.
REQUEST_FORMAT	Format of the certificate request file specified by REQUEST_FILE.
REVOKE	Specification that UCERT is to revoke a certificate.
REVOKE_REASON	Reason a certificate is being revoked.
SERIAL_NUMBER	Unique serial number to be assigned to the created certificate.
STATE	State name of the **`subject`** field of a certificate.
TRANSPORT_FILE	File containing certificate / private key information.
TRANSPORT_FILE_PWD	Password used to protect the file specified by TRANSPORT_FILE.
VERIFY	Specification that UCERT is to verify a certificate.
VERSION	Prints the program version and copyright information.

Table 2.1 Universal Certificate Configuration Options - z/OS, UNIX, and Windows

2.4 Universal Certificate for z/OS

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

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

2.4.1 JCL Procedure

Figure 2.1, below, illustrates the Universal Certificate for z/OS JCL procedure (**UCRPRC**, located in the **SUNVSAMP** library), that is provide to simplify the execution JCL and future maintenance.

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

Figure 2.1 Universal Certificate for z/OS – JCL Procedure

2.4.2 DD Statements used in JCL Procedure

Table 2.2, below, describes the DD statements used in the Universal Certificate for z/OS JCL illustrated in Figure 2.1.

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

Table 2.2 Universal Certificate for z/OS – DD Statements in JCL

2.4.3 JCL

Figure 2.2, below, illustrates the Universal Certificate for z/OS JCL.

```
//UCERT    EXEC PGM=UCERT
//STEPLIB  DD  DISP=SHR,DSN=UNV.SUNVLOAD
//UNVNLS   DD  DISP=SHR,DSN=UNV.SUNVNLS
//UNVDB    DD  DISP=SHR,DSN=UNV.UCRDB

//UNVTRACE DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
//SYSIN    DD  DUMMY
```

Figure 2.2 Universal Certificate for z/OS – JCL

2.4.4 Command Line Syntax

Figure 2.3, Figure 2.4, and Figure 2.5, below, illustrate the syntax – using the long form of command line options – of Universal Certificate for z/OS.

```
ucert
[-codepage codepage]
[-level {trace|audit|info|warn|error}]
[ -file ddname | -encryptedfile ddname [-key key] ]

Creating a certificate request.
{-create request
-request_file ddname [-request_format {pem|der}]
-private_key_file ddname [-private_key_format {pem|der}]
[-private_key_pwd password]
[-key_size {512|1024|2048}]
[-country name]
[-state name]
[-locality name]
[-organization name]
[-organizational_unit name]
[-common_name name]
{ [-dns_name name] | [-ip_address name] }
[-email_address name]

Creating a certificate from a certificate request.
| -create cert
-request_file ddname [-request_format {pem|der}]
-cert_file ddname [-cert_format {pem|der}]
-private_key_file ddname [-private_key_format {pem|der}]
[-private_key_pwd password]
-ca_cert_file ddname [-ca_cert_format {pem|der}]
[-serial_number number]
[-not_before_date date] [-not_after_date date]
[-ca {yes|no}]
[-cert_db ddname]
```

Figure 2.3 Universal Certificate for z/OS - Command Line Syntax (1 of 3)

```
Creating a certificate from a transport file.
| -create cert
-transport_file ddname [-transport_file_pwd password]
-cert_file ddname [-cert_format {pem|der}]
-private_key_file ddname [-private_key_format {pem|der}]
-ca_cert_file ddname [-ca_cert_format {pem|der}]

Creating a certificate revocation list.
| -create crl
-crl_file ddname [-crl_format {pem|der}]
-ca_cert_file ddname [-ca_cert_format {pem|der}]
-private_key_file ddname [-private_key_format {pem|der}]
[-private_key_pwd password]
-next_update_days days
-next_update_hours hours
[-cert_db ddname]

Creating a transport file.
| -create transport
-transport_file ddname [-transport_file_pwd password]
-cert_file ddname [-cert_format {pem|der}]
-private_key_file ddname [-private_key_format {pem|der}]
[-private_key_pwd password]
-ca_cert_file ddname [-ca_cert_format {pem|der}]

Revoking a certificate.
| -revoke cert
[-revoke_reason {unspecified|keyCompromise|caCompromised|
affiliationChange|superseded|cessationofOperation|privelegeWithdrawn}]
-cert_file ddname [-cert_format {pem|der}]
[-cert_db ddname]

Printing a certificate request.
| -print request
-request_file ddname [-request_format {pem|der}]

Printing a certificate.
| -print cert
-cert_file ddname [-cert_format {pem|der}]
```

Figure 2.4 Universal Certificate for z/OS - Command Line Syntax (2 of 3)

```
Printing a certificate revocation list.
| -print crl
-crl_file ddname [-crl_format {pem|der}]

Printing a transport file.
| -print transport
-transport_file ddname [-transport_file_pwd password]

Verifying a certificate.
| -verify cert
-cert_file ddname [-cert_format {pem|der}]
-ca_cert_file ddname [-ca_cert_format {pem|der}]
-crl_file ddname [-crl_format {pem|der}]

}

ucert
{ -help | -version }
```

Figure 2.5 Universal Certificate for z/OS - Command Line Syntax (3 of 3)

2.5 Universal Certificate for UNIX and Windows

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

This section describes the command line syntax of Universal Certificate for UNIX and Windows.

2.5.1 Command Line Syntax

Figure 2.6, Figure 2.7, and Figure 2.8, below, illustrate the syntax – using the long form of command line options – of Universal Certificate for UNIX and Windows.

```
ucert
[-codepage codepage]
[-nls_directory directory]
[-level {trace|audit|info|warn|error}]
[ -file filename | -encryptedfile filename [-key key] ]

Creating a certificate request.
{-create request
-request_file file [-request_format {pem|der}]
-private_key_file file [-private_key_format {pem|der}]
[-private_key_pwd password]
[-key_size {512|1024|2048}]
[-country name]
[-state name]
[-locality name]
[-organization name]
[-organizational_unit name]
[-common_name name]
{ [-dns_name name] | [-ip_address name] }
[-email_address name]
```

Figure 2.6 Universal Certificate for UNIX and Windows - Command Line Syntax (1 of 3)

```
Creating a certificate from a certificate request.
| -create cert
-request_file file [-request_format {pem|der}]
-cert_file file [-cert_format {pem|der}]
-private_key_file file [-private_key_format {pem|der}]
[-private_key_pwd password]
-ca_cert_file file [-ca_cert_format {pem|der}]
[-serial_number number]
[-not_before_date date] [-not_after_date date]
[-ca {yes|no}]
[-cert_db file]

Creating a certificate from a transport file.
| -create cert
-transport_file file [-transport_file_pwd password]
-cert_file file [-cert_format {pem|der}]
-private_key_file file [-private_key_format {pem|der}]
-ca_cert_file file [-ca_cert_format {pem|der}]

Creating a certificate revocation list.
| -create crl
-crl_file file [-crl_format {pem|der}]
-ca_cert_file file [-ca_cert_format {pem|der}]
-private_key_file file [-private_key_format {pem|der}]
[-private_key_pwd password]
-next_update_days days
-next_update_hours hours
[-cert_db file]

Creating a transport file.
| -create transport
-transport_file file [-transport_file_pwd password]
-cert_file file [-cert_format {pem|der}]
-private_key_file file [-private_key_format {pem|der}]
[-private_key_pwd password]
-ca_cert_file file [-ca_cert_format {pem|der}]
```

Figure 2.7 Universal Certificate for UNIX and Windows - Command Line Syntax (2 of 3)

```
Revoking a certificate.
| -revoke cert
[-revoke_reason {unspecified|keyCompromise|caCompromised|
affiliationChange|superseded|cessationofOperation|privelegeWithdrawn}]
-cert_file file [-cert_format {pem|der}]
[-cert_db file]

Printing a certificate request.
| -print request
-request_file file [-request_format {pem|der}]

Printing a certificate.
| -print cert
-cert_file file [-cert_format {pem|der}]

Printing a certificate revocation list.
| -print crl
-crl_file file [-crl_format {pem|der}]

Printing a transport file.
| -print transport
-transport_file file [-transport_file_pwd password]

Verifying a certificate.
| -verify cert
-cert_file file [-cert_format {pem|der}]
-ca_cert_file file [-ca_cert_format {pem|der}]
-crl_file file [-crl_format {pem|der}]

}

ucert
{ -help | -version }
```

Figure 2.8 Universal Certificate for UNIX and Windows - Command Line Syntax (3 of 3)

CHAPTER 3

Universal Certificate Configuration Options

3.1 Overview

This chapter provides detailed information on the configuration options available for use with Universal Certificate.

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

Information on how these options are used is documented Chapter 2 Universal Certificate.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

3.2 Configuration Options List

Table 3.1, below, identifies the Universal Certificate configuration options.

Option Name	Description	Page
CA	Specification for whether or not the certificate should be marked as a Certificate Authority certificate	64
CA_CERT_FILE	File from which the CA certificate is read or to which the CA certificate is written	65
CA_CERT_FORMAT	Format of the CA certificate file specified by CA_CERT_FILE	66
CERT_DB	Certificate database name	67
CERT_FILE	File from which the certificate is read or to which the certificate is written	68
CERT_FORMAT	Format of the certificate file specified by CERT_FILE	69
CODE_PAGE	Character code page used to translate text data	70
COMMAND_FILE_ENCRYPTED	Name of an encrypted command file	71
COMMAND_FILE_PLAIN	Name of a plain text command file	72
COMMON_NAME	Common name of the **subject** field of a certificate	73
COUNTRY	Country name of the **subject** field of a certificate	74
CREATE	Specification for Universal Certificate to create a certificate, certificate request, certificate revocation list (CRL), or transport file	76
CRL_FILE	File name to which the Certificate Revocation List (CRL) is written	77
CRL_FORMAT	Format of the CRL file specified by CRL_FILE	78
DNS_NAME	Domain Name System (DNS) name of the computer system for which the certificate identifies	79
EMAIL_ADDRESS	Email address of the entity identified by the certificate	80
ENCRYPTION_KEY	Key used to encrypt the command file	81
HELP	Writes a description of the command options and their format	82
IP_ADDRESS	Internet Protocol (IP) address of the computer system for which the certificate identifies	83
KEY_SIZE	Key size of the RSA public / private keys	84
LOCALITY	Locality name of the subject field of a certificate	85
MESSAGE_LEVEL	Level of messages to write	86
NEXT_UPDATE_DAYS	Number of days to the next CRL update	88
NEXT_UPDATE_HOURS	Number of hours to the next CRL update	89
NLS_DIRECTORY	Directory name where the code page UTT files are located	90
NOT_AFTER_DATE	Last day for which the certificate is considered valid	91
NOT_BEFORE_DATE	First day for which the certificate is considered valid	92
ORGANIZATION	Organization name of the **subject** field of a certificate	93

Option Name	Description	Page
ORGANIZATIONAL_UNIT	Organizational unit name of the **subject** field of a certificate	94
PRINT	Specification that Universal Certificate is to print a certificate or certificate request	95
PRIVATE_KEY_FILE	File name from which the RSA private key is read or to which the RSA private key is written	96
PRIVATE_KEY_FORMAT	Format of the private key file specified by PRIVATE_KEY_FILE	97
PRIVATE_KEY_PWD	Password used to read and write the private key file specified by PRIVATE_KEY_FILE	98
REQUEST_FILE	File from which the certificate request is read or to which the certificate request is written	99
REQUEST_FORMAT	Format of the certificate request file specified by REQUEST_FILE	100
REVOKE	Specification that Universal Certificate is to revoke a certificate	101
REVOKE_REASON	Reason that a certificate is being revoked	102
SERIAL_NUMBER	Unique serial number to be assigned to the created certificate	103
STATE	State name of the **subject** field of a certificate	104
TRANSPORT_FILE	File containing certificate / private key information	105
TRANSPORT_FILE_PWD	Password used to protect the file specified by TRANSPORT_FILE	106
VERIFY	Specification that Universal Certificate is to verify a certificate	107
VERSION	Writes the program version and copyright information	108

Table 3.1 Universal Certificate Configuration Options

3.3 CA

Description

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

Usage

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

Values

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

Valid values for *option* are:

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

[Default is no.]

3.4 CA_CERT_FILE

Description

The CA_CERT_FILE option specifies either:

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

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

The format of the file is specified by the CA_CERT_FORMAT option.

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-a *ddname* or *file*			√	√	√
Command Line, Long Form	-ca_cert_file *ddname* or *file*			√	√	√
Environment Variable	UCRCACERTFILE=*ddname* or *file*			√	√	√

Values

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

3.5 CA_CERT_FORMAT

Description

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

Usage

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

Values

format is the format of the CA certificate file.

Valid values for *format* are:

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

[Default is pem.]

See Section 2.2.7 File Formats in 2 Universal Certificate for details on file formats.

3.6 CERT_DB

Description

The CERT_DB option specifies the name of the certificate database.

UNIX and Windows

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

Usage

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

Values

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

3.7 CERT_FILE

Description

The CERT_FILE option specifies either:

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

The format of the file is specified by the CERT_FORMAT option.

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

Usage

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

Values

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

3.8 CERT_FORMAT

Description

The CERT_FORMAT option specifies the format of the certificate file specified by the CERT_FILE option.

Usage

Method	Method	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-D *format*			√	√	√
Command Line, Long Form	-cert_format *format*			√	√	√
Environment Variable	UCRCERTFORMAT=*format*			√	√	√

Values

format is the format of the certificate file.

Valid values for *format* are:

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

[Default is pem.]

See Section 2.2.7 File Formats in 2 Universal Certificate for details on file formats.

3.9 CODE_PAGE

Description

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-t *codepage*			√	√	√
Command Line, Long Form	-codepage *codepage*			√	√	√
Environment Variable	UCRCODEPAGE=*codepage*			√	√	√

Values

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

codepage references a Universal Translate Table (UTT) file provided with the product (see Section 36.4 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 36.3 Character Code Pages for a complete list of character code pages provided by Stonebranch Inc. for use with Stonebranch Solutions.

3.10 COMMAND_FILE_ENCRYPTED

Description

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

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

Universal Certificate is able to process command files that are either encrypted or plain text (see COMMAND_FILE_PLAIN). Use the Universal Encrypt utility to encrypt a plain text command file (see 17 Universal Encrypt).

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

Note: If an encrypted file is specified in this option, a plain text file should not be specified additionally in the COMMAND_FILE_PLAIN option. If it is, the file specified in COMMAND_FILE_PLAIN will be used.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-x *ddname* or *filename*			√	√	√
Command Line, Long Form	-encryptedfile *ddname* or *filename*			√	√	√
Environment Variable	n/a					

Values

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

3.11 COMMAND_FILE_PLAIN

Description

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

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

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

Note: If an encrypted file is specified in this option, a plain text file should not be specified additionally in the COMMAND_FILE_ENCRYPTED option. If it is, the file specified in this COMMAND_FILE_PLAIN option will be used.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-f *ddname* or *filename*			√	√	√
Command Line, Long Form	-file *ddname* or *filename*			√	√	√
Environment Variable	n/a					

Values

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

3.12 COMMON_NAME

Description

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

Usage

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

Values

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

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

3.13 COUNTRY

Description

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

Usage

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

Values

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

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

Country Codes

Table 3.2, below, identifies a subset of ISO 3166 country codes.

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

Table 3.2 Country Codes (Subset of ISO 3166)

3.14 CREATE

Description

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

Usage

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

Values

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

Valid values for *object* are:

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

3.15 CRL_FILE

Description

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

The format of the file is specified by the CRL_FORMAT option.

Usage

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

Values

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

3.16 CRL_FORMAT

Description

The CRL_FORMAT option specifies the format of the CRL file specified by the CRL_FILE option.

Usage

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

Values

format is the format of the CRL file.

Valid values for *format* are:

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

[Default is pem.]

See Section 2.2.7 File Formats in Chapter 2 Universal Certificate for details on file formats.

3.17 DNS_NAME

Description

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-dns_name *name*			√	√	√
Environment Variable	UCRDNSNAME=*name*			√	√	√

Values

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

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

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

3.18 EMAIL_ADDRESS

Description

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

Usage

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

Values

name is the e-mail address of the entity.

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

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

3.19 ENCRYPTION_KEY

Description

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

This key acts much like a password for the encrypted command file in that can be used to protect the file from decryption by unauthorized users. If a key was used to encrypt a command file (when Universal Encrypt was run), that same key must be specified to decrypt the file; otherwise, the decryption will fail.

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-K *key*			√	√	√
Command Line, Long Form	-key *key*			√	√	√
Environment Variable	UCRKEY=*key*			√	√	√

Values

key is the key used to encrypt the command file.

3.20 HELP

Description

The HELP option writes a description of the command options and their format.

Usage

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

Values

(There are no values used with this option.)

3.21 IP_ADDRESS

Description

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

Usage

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

Values

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

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

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

3.22 KEY_SIZE

Description

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

Usage

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

Values

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

Valid values for *size* are:

- **512**
- **1024**
- **2048**

[Default is *1024*.]

3.23 LOCALITY

Description

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

Usage

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

Values

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

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

3.24 MESSAGE_LEVEL

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l *level*			√	√	√
Command Line, Long Form	-level *level*			√	√	√
Environment Variable	UCRLEVEL=*level*			√	√	√

Values

level indicates either of the following level of messages:

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

z/OS

[Default is info.]

UNIX and Windows

[Default is warn.]

Trace Files

UNIX and Windows

Trace file name is **`ucert.trc`**. It is created in the current working directory.

z/OS

Trace file is written to ddname **UNVTRACE**.

3.25 NEXT_UPDATE_DAYS

Description

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

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

CRL creation requires the use of either this option or NEXT_UPDATE_HOURS.

Usage

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

Values

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

[Default is *0*.]

3.26 NEXT_UPDATE_HOURS

Description

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

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

CRL creation requires the use of either this option or NEXT_UPDATE_DAYS.

Usage

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

Values

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

[Default is *0*.]

3.27 NLS_DIRECTORY

Description

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

Usage

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

Values

directory is the name of the directory.

UNIX and Windows

Relative path names are relative to the current working directory.

3.28 NOT_AFTER_DATE

Description

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

Usage

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

Values

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

The format of date is either:

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

[Default is *365*.]

3.29 NOT_BEFORE_DATE

Description

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

Usage

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

Values

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

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

[Default is the current date.]

3.30 ORGANIZATION

Description

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

Usage

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

Values

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

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

3.31 ORGANIZATIONAL_UNIT

Description

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

Usage

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

Values

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

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

3.32 PRINT

Description

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

Usage

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

Values

object is the specification for what to print.

Valid values for *object* are:

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

3.33 PRIVATE_KEY_FILE

Description

The PRIVATE_KEY_FILE option specifies either:

- File from which the RSA private key is read
- File to which the RSA private key is written

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-e *ddname* or *file*			√	√	√
Command Line, Long Form	-private_key_file *ddname* or *file*			√	√	√
Environment Variable	UCRPRIVATEKEYFILE=*ddname* or *file*			√	√	√

Values

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

The format of the file is specified by the PRIVATE_KEY_FORMAT option.

3.34 PRIVATE_KEY_FORMAT

Description

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-E *format*			√	√	√
Command Line, Long Form	-private_key_format *format*			√	√	√
Environment Variable	UCRPRIVATEKEYFORMAT=*format*			√	√	√

Values

format is the format of the private key file.

Valid values for *format* are:

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

[Default is pem.]

See Section 2.2.7 File Formats in Chapter 2 Universal Certificate for details on file formats.

3.35 PRIVATE_KEY_PWD

Description

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-w *password*			√	√	√
Command Line, Long Form	-private_key_pwd *password*			√	√	√
Environment Variable	UCRPRIVATEKEYPWD=*password*			√	√	√

Values

password is the private key password.

3.36 REQUEST_FILE

Description

The REQUEST_FILE option specifies either:

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

The request is encoded in PKCS #10 syntax.

Usage

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

Values

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

The format of the file is specified by the REQUEST_FORMAT option.

3.37 REQUEST_FORMAT

Description

The REQUEST_FORMAT option specifies the format of the certificate request file specified by the REQUEST_FILE option.

Usage

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

Values

format is the format of the certificate request file.

Valid values for *format* are:

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

[Default is pem.]

See Section 2.2.7 File Formats in Chapter 2 Universal Certificate for details on file formats.

3.38 REVOKE

Description

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

Usage

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

Values

object is the specification to revoke a certificate.

Valid values for *object* are:

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

3.39 REVOKE_REASON

Description

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

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

Usage

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

Values

reason is the reason a certificate is being revoked.

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

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

[Default is ***unspecified***.]

3.40 SERIAL_NUMBER

Description

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

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

Usage

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

Values

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

3.41 STATE

Description

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

Usage

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

Values

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

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

3.42 TRANSPORT_FILE

Description

The TRANSPORT_FILE option specifies either:

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

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

Usage

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

Values

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

3.43 TRANSPORT_FILE_PWD

Description

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-transport_file_pwd *password*			√	√	√
Environment Variable	UCRTRANFILEPWD=*password*			√	√	√

Values

password is the password used to protect the transport file.

3.44 VERIFY

Description

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

Usage

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

Values

object is the specification to verify a certificate.

Valid values for *object* are:

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

3.45 VERSION

Description

The VERSION option writes the program version and copyright information.

Usage

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

Values

(There are no values used with this option.)

CHAPTER 4

Universal Control

4.1 Overview

This chapter provides information on the Universal Control (UCTL) utility.

Universal Control consists of two components:

- UCTL Manager
- UCTL Server

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

1. Start (START_CMD option)
   Starts a component on the specified system.
2. Stop (STOP_CMD option)
   Stops a component on the specified system.
3. Refresh (REFRESH_CMD option)
   Directs Universal Broker on the remote system to refresh the configuration data of all components, including itself, or a single component (currently, only Universal Event Monitor Server).

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

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

4.2 Universal Control Manager for z/OS

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

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

4.2.1 Usage

This section describes the control requests, JCL procedure and JCL, configuration and configuration options, and command line syntax of UCTL Manager for z/OS.

Control Requests

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

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

JCL Procedure

Figure 4.1, below, identifies the UCTL Manager for z/OS JCL procedure (**UCTLPRC**, located in the **SUNVSAMP** library) that is provided to simplify the execution JCL and future maintenance.

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

Figure 4.1 Universal Control Manager for z/OS – JCL Procedure

DD Statements used in JCL Procedure

Table 4.1, below, describes the DD statements used in the UCTL Manager for z/OS JCL procedure illustrated in Figure 4.1.

ddname	DCB Attributes *	Mode	Description
STEPLIB	DSORG=PO, RECFM=U	Input	Load library containing the program being executed.
UNVNLS	DSORG=PO, RECFM=(F, FB, V, VB)	Input	UCTL national language support library. Contains message catalogs and code page translation tables.
UNVTRACE	DSORG=PS, RECFM=(F, FB, V, VB)	Output	UCTL trace output.
SYSPRINT	DSORG=PS, RECFM=(F, FB, V, VB)	Output	stdout file for the UCTL program. UCTL does not write any messages to SYSPRINT.
SYSOUT *	DSORG=PS, RECFM=(F, FB, V, VB)	Output	stderr file for the UCTL program. UCTL writes its messages to SYSOUT.
* The C runtime library determines the default DCB attributes. Refer to the IBM manual *OS/390 C/C++ Programming Guide* for details on default DCB attributes for stream I/O			

Table 4.1 Universal Control Manager for z/OS – DD Statements in JCL Procedure

JCL

Figure 4.2, below, illustrates the UCTL Manager for z/OS JCL using the **UCTLPRC** procedure illustrated in Figure 4.1.

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

Figure 4.2 Universal Control Manager for z/OS – JCL

Job step STEP1 executes **UCTLPRC**.

The configuration options are specified on the SYSIN DD.

Configuration

Configuration consists of:

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

Configuration options are read from the following sources:

1. PARM keyword
2. SYSIN ddname
3. Command file ddname
4. Configuration file

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

Detailed information on these methods of configuration can be found in the Configuration Management chapters of the Infitran and Indesca 4.2.0 User Guides.

Configuration File

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

The configuration file, provided by the local Universal Broker, provides the simplest method of specifying configuration options whose values will not change with each command invocation. These default values are used if the options are not read from one or more other sources.

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

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

Configuration Options

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

Configuration Options Categories

Table 4.2, below, categorizes the configuration options into logical areas of application.

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

Table 4.2 Universal Control Manager for z/OS - Configuration Options Categories

The UCTL Manager configuration options for each category are summarized in the following tables. Each **Option Name** is a link to detailed information about that option.

Certificate Category Options

Option Name	Description
CA_CERTIFICATES	ddname of the PEM-formatted trusted CA X.509 certificates
CERTIFICATE	ddname of Manager's PEM-formatted X.509 certificate.
CERTIFICATE_REVOCATION_LIST	Location of Manager's PEM-formatted CRL.
PRIVATE_KEY	ddname of Manager's PEM-formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Manager's PRIVATE_KEY.
SSL_IMPLEMENTATION	Secure Socket Layer (SSL) implementation to be used for network communications
VERIFY_HOST_NAME	Specification that the Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specification that the Broker's X.509 certificate serial number field must be verified.

Command Category Options

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

Events Category Options

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

Local Category Options

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

Messages Category Options

Option Name	Description
MESSAGE_LANGUAGE	Language of messages written.
MESSAGE_LEVEL	Level of messages written.

Miscellaneous Category Options

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

Network Category Options

Option Name	Description
CODE_PAGE	Code page used to translate text data to and from the network.
CTL_SSL_CIPHER_LIST	SSL cipher list for the control session.
NETWORK_DELAY	Maximum number of seconds considered acceptable to wait for data communications.

Options Category Options

Option Name	Description
COMMAND_FILE_ENCRYPTED	Encrypted command file.
COMMAND_FILE_PLAIN	Plain text command file.
ENCRYPTION_KEY	Encryption key used to decrypt an encrypted command file specified by option COMMAND_FILE_ENCRYPTED.

Remote Category Options

Option Name	Description
HOSTNAME_RETRY_COUNT	Number of host connection attempts before ending with an error.
OUTBOUND_IP	Host or IP address to use for all outgoing IP connections.
REMOTE_HOST	TCP/IP host name of the remote Broker.
REMOTE_PORT	TCP/IP port number of the remote Broker.

User Category Options

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

Command Line Syntax

Figure 4.3, below, illustrates the command line syntax – using the command line, long form of the configuration options – of Universal Control Manager for z/OS.

```
uctl
{ -start compname [-cmdid id] | -stop compID [-userid user [-pwd password] ] |
      -refresh [compname] }
-host hostaddress
[-file ddname | -encryptedfile ddname [-key key] ]  *
[-port port]
[-cmdid id]
[-hostname_retry_count count]
[-outboundip host]
[-ssl_implementation {openssl | system}
[-system_id ID]
[-lang language]
[-level {trace|audit|info|warn|error}]
[-ca_certs ddname [-verify_host_name {yes|no|hostname}]
   [-verify_serial_number number] ]
[-cert ddname -private_key ddname [-private_key_pwd password] ]
[-crl ddname]
[-codepage codepage]
[-ctl_ssl_cipher_list cipherlist]
[-delay seconds]

uctl
{-help | -version}
```

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

Figure 4.3 Universal Control Manager for z/OS - Command Line Syntax

4.3 Universal Control Manager for Windows

This chapter provides information on Universal Control (UCTL) Manager specific to the Windows operating system.

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

- Command prompt
- Universal Configuration Manager

Command Prompt

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

Universal Configuration Manager

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

4.3.1 Usage

This section describes the control requests, configuration and configuration options, and command line syntax of UCTL Manager for Windows.

Control Requests

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

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

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

Configuration

Configuration consists of:

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

Configuration options are read from the following sources:

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

The order of precedence is the same as the list above; command line being the highest, and configuration file being the lowest. That is, options specified via a command line override options specified via a command file, and so on. The UCTL Manager configuration file is provided to the manager by the local Universal Broker with which it registers.

The configuration file, **`uctl.conf`**, provides the simplest method of specifying configuration options whose values will not change with each command invocation. These default values are used if the options are not read from one or more other sources.

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

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

Note: For any changes made directly to the UCTL Manager configuration file to become active, a Universal Broker refresh is required, or the Universal Broker service must be restarted. Changes made by the Universal Configuration Manager do not require any additional action for the options to become active.

Configuration Options

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

Configuration Options Categories

Table 4.3, below, categorizes configuration options into logical areas of application.

Category	Description
Certificates	X.509 certificate related options.
Command	Control command to execute.
Events	Options used to define event generation.
Installation	Options that specify installation requirements, such as directory locations.
Messages	Universal Control message options.
Miscellaneous	Options use to display command help and program versions.
Network	Options used to control the process of network data.
Options	Alternative methods to specify command options.
Remote	Network address of the remote system.
User	User account the Control command executes with on the remote system.

Table 4.3 Universal Control Manager for Windows - Command Options Categories

The Universal Control Manager command options for each of the categories listed in Table 4.3 are summarized in the following tables. Each Option Name is a link to detailed information about that option.

Certificate Category Options

Option Name	Description
CA_CERTIFICATES	Location of the PEM-formatted trusted CA X.509 certificates.
CERTIFICATE	Location of Manager's PEM-formatted X.509 certificate.
CERTIFICATE_REVOCATION_LIST	Location of Manager's PEM-formatted CRL.
PRIVATE_KEY	Location of Manager's PEM-formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Manager's PRIVATE_KEY.
VERIFY_HOST_NAME	Specification that the Universal Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specification that the Universal Broker's X.509 certificate serial number field must be verified.

Command Category Options

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

Events Category Options

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

Installation Category Options

Option Name	Description
INSTALLATION_DIRECTORY	Directory in which Universal Control Server is installed.

Messages Category Options

Option Name	Description
MESSAGE_LANGUAGE	Language of messages written.
MESSAGE_LEVEL	Level of messages written.
NLS_DIRECTORY	Location of UMC and UTT files

Miscellaneous Category Options

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

Network Category Options

Option Name	Description
CODE_PAGE	Code page used to translate text data to and from the network.
CTL_SSL_CIPHER_LIST	SSL cipher list for the control session.
NETWORK_DELAY	Maximum number of seconds considered acceptable to wait for data communications.

Options Category Options Summary

Option Name	Description
COMMAND_FILE_ENCRYPTED	Encrypted command file.
COMMAND_FILE_PLAIN	Plain text command file.
ENCRYPTION_KEY	Encryption key used to decrypt an encrypted command file specified by option COMMAND_FILE_ENCRYPTED.

Remote Category Options

Option Name	Description
HOSTNAME_RETRY_COUNT	Number of host connection attempts before ending with an error.
OUTBOUND_IP	Host or IP address to use for all outgoing IP connections.
REMOTE_HOST	TCP/IP host name of the remote Broker.
REMOTE_PORT	TCP/IP port number of the remote Broker.

User Category Options

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

Command Line Syntax

Figure 4.3, below, illustrates the command options syntax — using the command line, long form of the configuration options — of Universal Control Manager for Windows.

```
uctl
{ -start compname [-cmdid id] | -stop compID [-userid user [-pwd password] ] |
      -refresh [compname] }
-host hostaddress
[-file filename | -encryptedfile filename [-key key] ]  *
[-port port]
[-cmdid id]
[-hostname_retry_count count]
[-outboundip host]
[-lang language]
[-level {trace|audit|info|warn|error}]
[-ca_certs file [-verify_host_name {yes|no|hostname}]
   [-verify_serial_number number] ]
[-cert file -private_key file [-private_key_pwd password] ]
[-crl file]
[-codepage codepage]
[-ctl_ssl_cipher_list cipherlist]
[-delay seconds]

uctl
{-help | -version}
```

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

Figure 4.4 Universal Control Manager for Windows - Command Syntax

4.4 Universal Control Manager for UNIX

This chapter provides information on Universal Control (UCTL) Manager specific to the UNIX operating system.

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

4.4.1 Usage

This section describes the control requests, configuration and configuration options, and command line syntax of UCTL Manager for UNIX.

Control Requests

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

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

Configuration

Configuration consists of:

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

Configuration options are read from the following sources:

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

The order of precedence is the same as the list above; command line being the highest, and configuration file being the lowest. That is, options specified via a command line override options specified via a command file, and so on. The UCTL Manager configuration file is provided to the manager by the local Universal Broker with which it registers.

The configuration file, **`uctl.conf`**, provides the simplest method of specifying configuration options whose values will not change with each command invocation. These default values are used if the options are not read from one or more other sources.

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

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

Configuration Options

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

Configuration Options Categories

Table 4.4, below, categorizes configuration options into logical areas of application.

Category	Description
Certificates	X.509 certificate related options.
Command	Control command to execute.
Events	Options used to define event generation.
Installation	Options that specify installation requirements, such as directory locations.
Local	Options required for local broker registration.
Messages	Universal Control message options.
Miscellaneous	Options use to display command help and program versions.
Network	Options used to control the process of network data.
Options	Alternative methods to specify command options.
Remote	Network address of the remote system.
User	User account the Control command executes with on the remote system.

Table 4.4 Universal Control Manager for UNIX - Command Options Categories

The UCTL Manager configuration options for each category are summarized in the following tables. Each **Option Name** is a link to detailed information about that option.

Certificate Category Options

Option Name	Description
CA_CERTIFICATES	Location of the PEM-formatted trusted CA X.509 certificates.
CERTIFICATE	Location of Manager's PEM-formatted X.509 certificate.
CERTIFICATE_REVOCATION_LIST	Location of Manager's PEM-formatted CRL.
PRIVATE_KEY	Location of Manager's PEM-formatted RSA private key.
PRIVATE_KEY_PWD	Password for the Manager's PRIVATE_KEY.
VERIFY_HOST_NAME	Specification that the Universal Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specification that the Universal Broker's X.509 certificate serial number field must be verified.

Command Category Options

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

Events Category Options

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

Installation Category Options

Option Name	Description
INSTALLATION_DIRECTORY	Directory in which Universal Control Server is installed.

Local Category Options

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

Messages Category Options

Option Name	Description
MESSAGE_LANGUAGE	Language of messages written.
MESSAGE_LEVEL	Level of messages written.
NLS_DIRECTORY	Location of UMC and UTT files.

Miscellaneous Category Options

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

Network Category Options

Option Name	Description
CODE_PAGE	Code page used to translate text data to and from the network.
CTL_SSL_CIPHER_LIST	SSL cipher list for the control session.
NETWORK_DELAY	Maximum number of seconds considered acceptable to wait for data communications.

Options Category Options

Option Name	Description
COMMAND_FILE_ENCRYPTED	Encrypted command file.
COMMAND_FILE_PLAIN	Plain text command file.
ENCRYPTION_KEY	Encryption key used to decrypt an encrypted command file specified by option COMMAND_FILE_ENCRYPTED.

Remote Category Options

Option Name	Description
HOSTNAME_RETRY_COUNT	Number of host connection attempts before ending with an error.
OUTBOUND_IP	Host or IP address to use for all outgoing IP connections.
REMOTE_HOST	TCP/IP host name of the remote Broker.
REMOTE_PORT	TCP/IP port number of the remote Broker.

User Category Options

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

Command Line Syntax

Figure 4.3, below, illustrates the command line syntax – using the command line, long form of the configuration options – of Universal Control Manager for UNIX.

```
uctl
{ -start compname [-cmdid id] | -stop compID [-userid user [-pwd password] ] |
      -refresh [compname] }
-host hostaddress
[-file filename | -encryptedfile filename [-key key] ]  *
[-port port]
[-cmdid id]
[-hostname_retry_count count]
[-outboundip host]
[-bif_directory directory]
[-plf_directory directory]
[-lang language]
[-level {trace|audit|info|warn|error}]
[-ca_certs file [-verify_host_name {yes|no|hostname}]
   [-verify_serial_number number] ]
[-cert file -private_key file [-private_key_pwd password] ]
[-crl file]
[-codepage codepage]
[-ctl_ssl_cipher_list cipherlist]
[-delay seconds]

uctl
{-help | -version}
```

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

Figure 4.5 Universal Control Manager for UNIX - Command Line Syntax

4.5 Universal Control Manager for IBM i

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

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

4.5.1 Usage

This section describes the command execution environments, control requests, configuration and configuration options, and command line syntax of UCTL Manager for IBM i.

Stonebranch Solutions for IBM i Commands

The names of the Stonebranch Solutions for IBM i commands that are installed in the IBM i **QSYS** library are tagged with the Stonebranch Solutions for IBM i **v**ersion / **r**elease / **m**odification number, **420**. The names of the commands installed in the Stonebranch Solutions for IBM i product library, **UNVPRD420**, are untagged.

To maintain consistency across releases, you may prefer to use the untagged names in your production environment. The **UCHGRLS** (Change Release Tag) program lets you change the tagged command names in **QSYS** to the untagged command names in **UNVPRD420**.

(See the Stonebranch Solutions 4.2.0 Installation Guide for detailed information on **UCHGRLS**.)

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

Command Execution Environment

The command is valid in all environments:

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

Control Requests

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

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

Configuration

Configuration consists of:

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

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

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

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

The configuration file, **`UNVPRD420/UNVCONF(UCTL)`**, provides the simplest method of specifying configuration options whose values will not change with each command invocation. These default values are used if the options are not read from one or more other sources.

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

Configuration Options

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

Configuration Options Categories

Table 4.5, below, categorizes configuration options into logical areas of application.

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

Table 4.5 Universal Control Manager for IBM i - Command Options Categories

The UCTL Manager configuration options for each category are summarized in the following tables. Each **Option Name** is a link to detailed information about that option.

Certificate Category Options

Option Name	Description
CA_CERTIFICATES	Location of the PEM-formatted trusted CA X.509 certificates.
CERTIFICATE	Location of Manager's PEM-formatted X.509 certificate.
CERTIFICATE_REVOCATION_LIST	Location of Manager's PEM-formatted CRL.
PRIVATE_KEY	Location of Manager's PEM-formatted RSA private key.
PRIVATE_KEY_PWD	The password for the Manager's PRIVATE_KEY.
VERIFY_HOST_NAME	Specification that the Universal Broker's X.509 certificate host name field must be verified.
VERIFY_SERIAL_NUMBER	Specification that the Universal Broker's X.509 certificate serial number field must be verified.

Command Category Options

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

Events Category Options

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

Local Category Options

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

Messages Category Options

Option Name	Description
MESSAGE_LANGUAGE	Language of messages written.
MESSAGE_LEVEL	Level of messages written.

Miscellaneous Category Options

Option Name	Description
VERSION	Write program version.

Network Category Options

Option Name	Description
CODE_PAGE	Code page used to translate text data to and from the network.
CTL_SSL_CIPHER_LIST	SSL cipher list for the control session.
NETWORK_DELAY	Maximum number of seconds considered acceptable to wait for data communications.

Options Category Options

Option Name	Description
COMMAND_FILE_ENCRYPTED	Encrypted command file.
COMMAND_FILE_PLAIN	Plain text command file.
ENCRYPTION_KEY	Encryption key used to decrypt an encrypted command file specified by option COMMAND_FILE_ENCRYPTED.

Remote Category Options

Option Name	Description
HOSTNAME_RETRY_COUNT	Number of host connection attempts before ending with an error.
OUTBOUND_IP	Host or IP address to use for all outgoing IP connections.
REMOTE_HOST	TCP/IP host name of the remote Broker.
REMOTE_PORT	TCP/IP port number of the remote Broker.

User Category Options

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

Command Line Syntax

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

Figure 4.6, below, illustrates the command line syntax — using the STRUCT parameter form of the configuration options — of Universal Control Manager for Windows.

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

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

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

Figure 4.6 Universal Control Manager for IBM i - Command Options Syntax

4.6 Universal Control Manager for HP NonStop

This chapter provides information on Universal Control (UCTL) Manager specific to the UNIX operating system.

**Currently, HP NonStop runs Universal Control 2.1.1.
This chapter provides information for that version.**

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

4.6.1 Usage

This section describes the control requests, configuration and configuration options, and command line syntax of UCTL Manager for HP NonStop.

Control Requests

UCTL Manager for HP NonStop supports only the following two Universal Control control requests:

4. Stop (STOP_CMD option)
5. Refresh (REFRESH_CMD option)

Configuration

Configuration consists of:

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

Configuration options are read from the following sources:

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

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

The configuration file, **UCTLCFG**, provides the simplest method of specifying configuration options whose values will not change with each command invocation. These default values are used if the options are not read from one or more other sources.

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

Configuration Options

This section describes the configuration options used to execute UCTL Manager for HP NonStop.

Configuration Options Categories

Table 4.7, below, categorizes configuration options into logical areas of application.

Category	Description
Command	Control command to execute.
Remote	Network address of the remote system.
User	User account the Control command executes with on the remote system.
Messages	Universal Control message options.
Network	Options used to control the process of network data.
Options	Alternative methods to specify command options.
Miscellaneous	Options use to display command help and program versions.
Installation	Options that specify installation requirements, such as, directory locations.

Figure 4.7 Universal Control Manager for HP NonStop - Command Options Categories

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

Each **Option Name** is a link to detailed information about that option.

Command Category Options

Option Name	Description
STOP_CMD	Instruction to stop a component being executed by a Broker.
REFRESH_CMD	Instruction to a Broker to refresh configuration data.

Installation Category Options

Option Name	Description
INSTALLATION_DIRECTORY	Directory in which the product is installed.

Messages Category Options

Option Name	Description
MESSAGE_LANGUAGE	Language of messages written.
MESSAGE_LEVEL	Level of messages written.

Miscellaneous Category Options

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

Network Category Options

Option Name	Description
CODE_PAGE	Code page used to translate text data to and from the network.
NETWORK_DELAY	Maximum number of seconds considered acceptable to wait for data communications.

Options Category Options

Option Name	Description
COMMAND_FILE_ENCRYPTED	Encrypted command file.
COMMAND_FILE_PLAIN	Plain text command file.
ENCRYPTION_KEY	Encryption key used to decrypt an encrypted command file specified by option COMMAND_FILE_ENCRYPTED.

Remote Category Options

Option Name	Description
REMOTE_HOST	TCP/IP host name of the remote Broker.
REMOTE_PORT	TCP/IP port number of the remote Broker.

User Category Options

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

Command Line Syntax

Figure 4.3, below, illustrates the command options syntax of Universal Control Manager for HP NonStop.

Section Configuration Options provides a description of the options.

```
uctl
{ -stop compID [-userid user [-pwd password] ] | -refresh [compname] }
-host hostaddress
[-file filename | -encryptedfile filename [-key key] ]  *
[-port port]
[-lang language]
[-level {trace|audit|info|warn|error}]
[-codepage codepage]
[-delay seconds]

uctl
{-help | -version}
```

Figure 4.8 Universal Control Manager for HP NonStop - Command Options Syntax

4.7 Universal Control Server for z/OS

This chapter documents the Universal Control (UCTL) Server at a detailed level, specific to the z/OS operating system.

Environment

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

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

User Identification

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

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

4.7.1 Component Definition

All Stonebranch Solutions components managed by Universal Broker have a component definition. The component definition is a text file of options containing component-specific information required by Universal Broker. (For details on how Universal Broker manages components, see the Universal Broker 4.2.0 Reference Guide.)

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

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

Table 4.6, below, identifies all of the options that comprise the UCTL Server for z/OS component definition.

Each **Option Name** is a link to detailed information about that component definition option.

Option Name	Description
AUTOMATICALLY_START	Specification for whether or not UCTL Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCTL Server,
CONFIGURATION_FILE	Name of the UCTL Server's configuration file,
RUNNING_MAXIMUM	Maximum number of UCTL Servers that can run simultaneously,
START_COMMAND	Member name of the UCTL Server program,
WORKING_DIRECTORY	Directory used as the working directory of the UCTL Server,

Table 4.6 UCTL Server for z/OS - Component Definition Options

4.7.2 Configuration

UCTL Server configuration consists of defining runtime and default values. This section describes the Server configuration options.

Configuration File

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

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

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

Configuration Options Summary

Table 4.7, below, identifies all of the UCTL Server for z/OS configuration options. Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
CODE_PAGE	Code page used for text translation.
EVENT_GENERATION	Events to be generated as persistent events.
MESSAGE_LEVEL	Level of messages printed.
TMP_DIRECTORY	HFS directory name used for temporary files.
USER_SECURITY	Specification for whether or not user authentication is active.

Table 4.7 UCTL Server for z/OS - Configuration Options

Universal Access Control List

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

UACL Entries

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

Table 4.8 identifies all UCTL Server for z/OS UACL entries.

Each **UACL Entry Name** is a link to detailed information about that UACL entry.

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

Table 4.8 UCTL Server for z/OS - UACL Entries

4.8 Universal Control Server for Windows

This chapter documents the Universal Control (UCTL) Server at a detailed level, specific to the Windows family of operating systems.

Environment

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

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

User Identification

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

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

4.8.1 Component Definition

All Stonebranch Solutions components managed by Universal Broker have a component definition. The component definition is a text file of options containing component-specific information required by Universal Broker. (For details on how Universal Broker manages components, see the Universal Broker 4.2.0 Reference Guide.)

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

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

Note: The component definitions for all Stonebranch Solutions components are identified in the Component Definitions property page of the Universal Broker (see Figure 4.9, below).

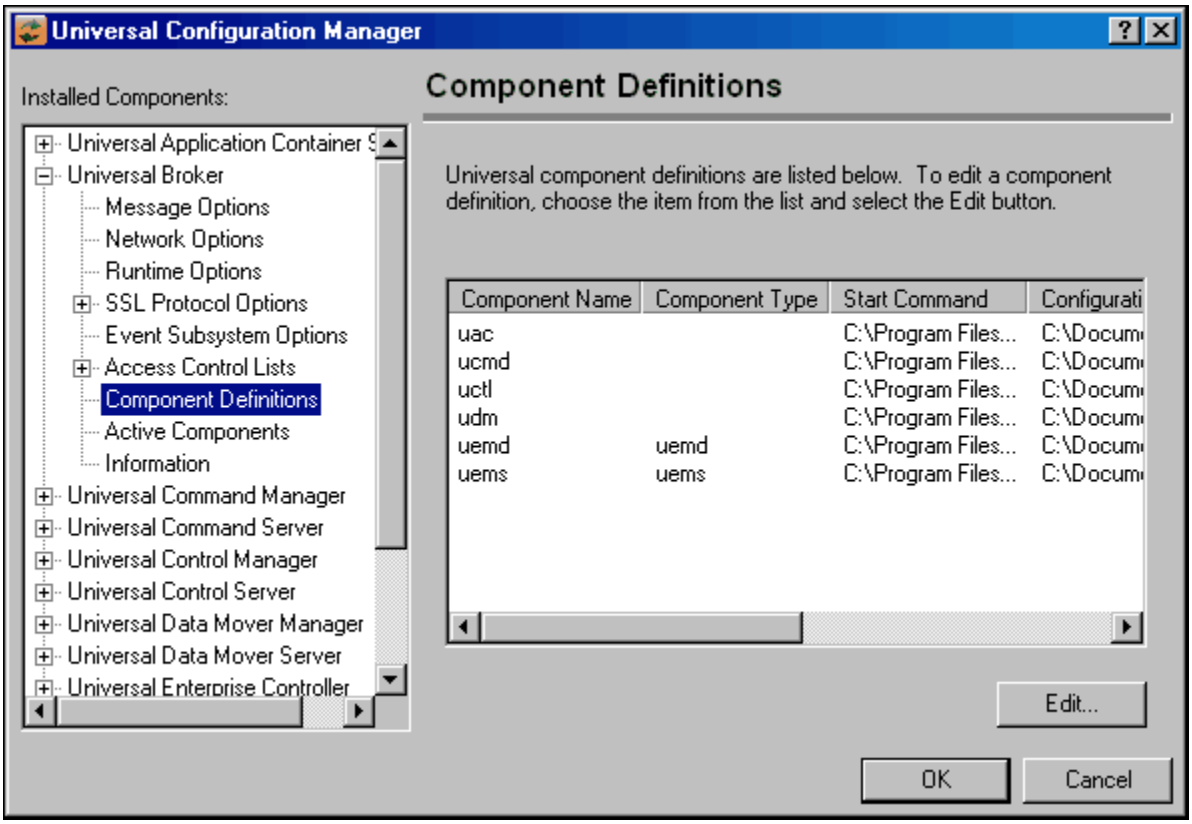

Figure 4.9 Universal Configuration Manager - Component Definitions

Table 4.9, below, identifies all of the options that comprise the UCTL Server for WIndows component definition.

Each **Option Name** is a link to detailed information about that component definition option.

Option Name	Description
AUTOMATICALLY_START	Specification for whether or not UCTL Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCTL Server.
CONFIGURATION_FILE	Name of the UCTL Server configuration file.
RUNNING_MAXIMUM	Maximum number of UCTL Servers that can run simultaneously.
START_COMMAND	Full path name of the UCTL Server program.
WORKING_DIRECTORY	Directory used as the working directory of the UCTL Server.

Table 4.9 UCTL Server for Windows - Component Definition Options

4.8.2 Configuration

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

Configuration File

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

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

However, the Universal Configuration Manager application, accessible via the Control Panel, is the recommended way to set Windows configuration options. The Universal Configuration Manager provides a graphical interface and context-sensitive help, and helps protect the integrity of the configuration file by validating all changes to configuration option values.

Note: For any changes made directly to the UCTL Server configuration file to become active, a Universal Broker refresh is required, or the Universal Broker service must be restarted. Changes made by the Universal Configuration Manager do not require any additional action for the options to become active.

Configuration Options Summary

Table 4.7, below, identifies all of the UCTL Server for Windows configuration options. Each **Option Name** is a link to detailed information about that configuration option.

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

Table 4.10 UCTL Server for Windows - Configuration Options

Universal Access Control List

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

UACL Entries

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

Table 4.8 identifies all UCTL Server for Windows UACL entries.

Each **UACL Entry Name** is a link to detailed information about that UACL entry.

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

Table 4.11 Universal Control for Windows - UACL Entries

4.9 Universal Control Server for UNIX

This chapter documents the Universal Control (UCTL) Server at a detailed level, specific to the UNIX operating system.

Environment

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

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

User Identification

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

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

4.9.1 Component Definition

All Stonebranch Solutions components managed by Universal Broker have a component definition. The component definition is a text file of options containing component-specific information required by Universal Broker. (For details on how Universal Broker manages components, see the Universal Broker 4.2.0 Reference Guide.)

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

Table 4.12, below, identifies all of the options that comprise the UCTL Server for UNIX component definition.

Each **Option Name** is a link to detailed information about that component definition option.

Option Name	Description
AUTOMATICALLY_START	Specification for whether or not UCTL Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCTL Server.
CONFIGURATION_FILE	Name of the UCTL Server configuration file.
RUNNING_MAXIMUM	Maximum number of UCTL Servers that can run simultaneously.
START_COMMAND	Full path name of the UCTL Server program.
WORKING_DIRECTORY	Directory used as the working directory of the UCTL Server.

Table 4.12 UCTL Server for UNIX - Component Definition Options

4.9.2 Configuration

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

Configuration File

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

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

Note: For any changes made directly to the UCTL Server configuration file to become active, a Universal Broker refresh is required, or the Universal Broker service must be restarted.

Configuration Options Summary

Table 4.7, below, identifies all of the Universal Control Server for UNIX configuration options. Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.
CODE_PAGE	Code page used for text translation.
EVENT_GENERATION	Events to be generated as persistent events.
INSTALLATION_DIRECTORY	Base directory in which Universal Control Server is installed.
MESSAGE_LEVEL	Level of messages written.
NLS_ DIRECTORY	Location of UMC and UTT files.
TMP_DIRECTORY	Directory name used for temporary files.
TRACE_DIRECTORY	Location of trace files.
USER_SECURITY	Specification for whether or not user authentication is active.

Table 4.13 UCTL Server for UNIX - Configuration Options

Universal Access Control List

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

UACL Entries

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

Table 4.8 identifies all UCTL Server for UNIX UACL entries.

Each **UACL Entry Name** is a link to detailed information about that UACL entry.

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

Table 4.14 UCTL Server for UNIX - UACL Entries

4.10 Universal Control Server for IBM i

This chapter documents the Universal Control Server at a detailed level, specific to the IBM i operating system.

Environment

The Universal Control Server runs under the **UNVUBR420** subsystem's pre-start job **UNVSRV**. When the Broker receives a request for a Universal Command component, it passes the request to the **UCTSRV** program running under the **UNVSRV** pre-start job.

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

User Identification

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

- With user security active, the Server requires the Manager to supply a valid user ID and password for the local system. The user command executes with the user profile of the received user ID.
- With user security inactive, the Server does not require the Manager to supply a valid user ID. The user command executes with the user profile of the Server. The user profile of the Server is inherited from the Broker. The inherited profile is **UNVUBR420**; as installed, this profile provides a very high level of authority including *ALLOBJ.

Current Library and Working Directory

The current library and working directory of a user command depends on whether user security is active or inactive:

- With user security active, the user's current library is designated by the user profile and the working directory is the home directory of the user profile.
- With user security inactive, the current library is that for the user profile (the installation default, **UNVUBR420**) associated with the service program. Note that the default value used for the current library is **UNVTMP420**. Care should be taken to avoid name clashes and other consequences of multiple processes sharing a common current library and working directory.

4.10.1 Component Definition

All Stonebranch Solutions components managed by Universal Broker have a component definition. The component definition is a text file of options containing component-specific information required by Universal Broker. (For details on how Universal Broker manages components, see the Universal Broker 4.2.0 Reference Guide.)

The default location for Universal Broker component definition files is **`UNVPRD420/UNVCOMP`**. The UCTL Server component member is **`UCTL`**.

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

Table 4.12, below, identifies all of the options that comprise the UCTL Server for IBM i component definition.

Each **Option Name** is a link to detailed information about that component definition option.

Option Name	Description
AUTOMATICALLY_START	Specification for whether or not UCTL Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCTL Server.
CONFIGURATION_FILE	Name of the UCTL Server configuration file.
RUNNING_MAXIMUM	Maximum number of UCTL Servers that can run simultaneously.
START_COMMAND	Full path name of the UCTL Server program.
WORKING_DIRECTORY	Directory used as the working directory of the UCTL Server.

Table 4.15 UCTL Server for IBM i - Component Definition Options

4.10.2 Configuration

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

Configuration File

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

The Universal Control Server configuration file name is specified in the Universal Control Server component definition. The default file name is **`UNVPRD420/UNVCONF(UCTS)`**.

Configuration Options Summary

Table 4.7, below, identifies all of the Universal Control Server for IBM i configuration options. Each **Option Name** is a link to detailed information about that configuration option.

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

Table 4.16 Universal Control Server for IBM i - Configuration Options

Universal Access Control List

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

UACL Entries

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

Table 4.8 identifies all UCTL Server for IBM i UACL entries.

Each **UACL Entry Name** is a link to detailed information about that UACL entry.

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

Table 4.17 Universal Control for IBM i - UACL Entries

4.11 Universal Control Server for HP NonStop

This chapter documents the Universal Control (UCTL) Server at a detailed level, specific to the HP NonStop operating system.

**Currently, HP NonStop runs Universal Control 2.1.1.
This chapter provides information for that version.**

Environment

UCTL Server runs as a background OSS process. It does not interact with a console.

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

User Identification

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

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

4.11.1 Component Definition

All Stonebranch Solutions components managed by Universal Broker have a component definition. The component definition is a text file of options containing component-specific information required by Universal Broker. (For details on how Universal Broker manages components, see the Universal Broker 4.2.0 Reference Guide.)

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

Table 4.12, below, identifies all of the options that comprise the UCTL Server for HP NonStop component definition.

Each **Option Name** is a link to detailed information about that component definition option.

Option Name	Description
AUTOMATICALLY_START	Specification for whether or not UCTL Server starts automatically when Universal Broker is started.
COMPONENT_NAME	Name by which the clients know the UCTL Server.
CONFIGURATION_FILE	Name of the UCTL Server configuration file.
RUNNING_MAXIMUM	Maximum number of UCTL Servers that can run simultaneously.
START_COMMAND	Full path name of the UCTL Server program.
WORKING_DIRECTORY	Directory used as the working directory of the UCTL Server.

Table 4.18 UCTL Server for HP NonStop - Component Definition Options

4.11.2 Configuration

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

Configuration File

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

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

Configuration Options

Table 4.7, below, identifies all of the Universal Control Server for IBM i configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

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

Table 4.19 Universal Control Server for IBM i - Configuration Options

Universal Access Control List

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

UACL Entries

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

Table 4.8 identifies all UCTL Server for HP NonStop UACL entries.

Each **UACL Entry Name** is a link to detailed information about that UACL entry.

UACL Entry Name	Description
UCTL_ACCESS	Allows or denies access to Universal Control Server services.

Table 4.20 UCTL Server for HP NonStop - UACL Entries

CHAPTER 5

Universal Control Manager Configuration Options

5.1 Overview

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

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

Information on how these options are used is documented in Chapter 4 Universal Control.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented or each option.

5.2 Configuration Options List

Table 5.1 identifies all Universal Control configuration options.

Option	Description	Page
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.	165
BIF_DIRECTORY	Broker Interface Directory that specifies the location of the Universal Broker interface file.	166
CA_CERTIFICATES	ddname of the PEM formatted trusted CA X.509 certificates.	167
CERTIFICATE	ddname of Manager's PEM-formatted X.509 certificate.	168
CERTIFICATE_REVOCATION_LIST	Location of Manager's PEM-formatted CRL.	169
CODE_PAGE	Code page used to translate text data to and from the network.	170
COMMAND_FILE_ENCRYPTED	Encrypted command file.	171
COMMAND_FILE_PLAIN	Plain text command file.	172
COMMAND_ID	Identity of a started component.	173
CTL_SSL_CIPHER_LIST	SSL cipher list for the control session.	174
ENCRYPTION_KEY	Encryption key used to decrypt an encrypted command file specified by option COMMAND_FILE_ENCRYPTED.	175
EVENT_GENERATION	Events to be generated as persistent events.	176
HELP	Write configuration option help.	178
HOSTNAME_RETRY_COUNT	Number of host connection attempts before ending with an error.	179
INSTALLATION_DIRECTORY	Directory in which the product is installed.	180
MESSAGE_LANGUAGE	Language of messages formatted.	181
MESSAGE_LEVEL	Level of messages written.	182
NETWORK_DELAY	Maximum number of seconds considered acceptable to wait for data communications.	184
NLS_DIRECTORY	NLS directory.	185
OUTBOUND_IP	Host or IP address to use for all outgoing IP connections.	186
PLF_DIRECTORY	Program Lock File directory that specifies the location of the Universal Control Manager program lock file.	187
PRIVATE_KEY	ddname of Manager's PEM-formatted RSA private key.	188
PRIVATE_KEY_PWD	Password for the Manager's PRIVATE_KEY.	189
REFRESH_CMD	Instruct a Broker or component to refresh its configuration.	190
REMOTE_HOST	TCP/IP host name of the remote computer on which Universal Broker is running and accepting connections.	192
REMOTE_PORT	TCP/IP port number of the remote computer on which Universal Broker is running and accepting connections.	193

Option	Description	Page
SSL_IMPLEMENTATION	Secure Socket Layer (SSL) implementation to be used for network communications.	194
START_CMD	Instruction to a Universal Broker to start a component.	195
STOP_CMD	Instruction to stop a component being executed by a Broker.	196
SYSTEM_ID	Local Universal Broker with which the Universal Control Manager must register.	198
USER_ID	User ID or account with which to execute the Control command.	198
USER_PASSWORD	Password associated with USER_ID.	199
VERIFY_HOST	Specification that the Broker's X.509 certificate host name field must be verified.	200
VERIFY_SERIAL_NUMBER	Specification that the Broker's X.509 certificate serial number field must be verified.	202
VERSION	Write program version.	203

Table 5.1 Universal Control Manager Configuration Options

5.3 ACTIVITY_MONITORING

Description

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	n/a					
Environment Variable	n/a					
Configuration File Keyword	activity_monitoring *option*	√		√	√	√
STRUCT Parameter	n/a					

Values

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

Valid values for *option* are:

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

[Default is yes.]

5.4 BIF_DIRECTORY

Description

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

Usage

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

Values

directory is the name of the BIF directory.

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

5.5 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 Universal Broker certificate authentication and verification is desired.

Usage

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

Values

z/OS

ddname is the ddname of the X.509 certificates. The value is used only when the SSL_IMPLEMENTATION option is set to *OPENSSL*.

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

UNIX and Windows

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

IBM i

file is the qualified file name of the X.509 certificates file.

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

5.6 CERTIFICATE

Description

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

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

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

Usage

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

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.

IBM i

file is the qualified file name of the X.509 certificate file. The file name can be qualified by a library name.

5.7 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	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-crl *ddname* or *file*			√	√	√
Environment Variable	UCTLCRL=*file*	√		√	√	
Configuration File Keyword	crl *ddname* or *file*	√		√	√	√
STRUCT Parameter	CRLFILE(*file*[*lib*]) [CRLMBR(*member*)]	√				

Values

z/OS

ddname is the ddname of the file containing the CRL.

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

UNIX and 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.

5.8 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.

Usage

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

Value

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

codepage references a Universal Translate Table (UTT) file provided with the product (see Section 36.4 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 36.3 Character Code Pages for a complete list of character code pages provided by Stonebranch Inc. for use with Stonebranch Solutions.

5.9 COMMAND_FILE_ENCRYPTED

Description

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

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

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

Use the Universal Encrypt utility provided with Universal Command to encrypt a plain text command file. (For information on Universal Encrypt, see Chapter 17 Universal Encrypt). If a key was used to encrypt the file, the same key must be supplied using the ENCRYPTION_KEY option.

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

Usage

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

Value

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

5.10 COMMAND_FILE_PLAIN

Description

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

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

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

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

Usage

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

Value

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

5.11 COMMAND_ID

Description

The COMMAND_ID option identifies a started Stonebranch Solutions component.

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

Usage

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

Value

id is any value that identifies the component.

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

IBM i

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

5.12 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	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-ctl_ssl_cipher_list *cipherlist*			√	√	√
Environment Variable	UCTLCTLSSLCIPHERLIST= *cipherlist*	√		√	√	
Configuration File Keyword	ctl_ssl_cipher_list *cipherlist*	√		√	√	√
STRUCT Parameter	CTLCPHRLST(*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.

Table 5.2 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 5.2 SSL Cipher Suites (for CTL_SSL_CIPHER_LIST)

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

5.13 ENCRYPTION_KEY

Description

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

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

If a key was used to encrypt a command file (when Universal Encrypt was run), that same key must be specified to decrypt the file, or the decryption will fail. If no key is specified, the default key is used.

Usage

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

Value

key is the key used to encrypt the command file.

5.14 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 Stonebranch Solutions components, see the Universal Event Subsystem 4.2.0 Event Definitions Guide.)

Usage

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

Values

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

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

Event types can be selected for inclusion or exclusion:

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

Examples

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

[Default is *X (no event types).]**

5.15 HELP

Description

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

Usage

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

Value

(There are no values for the HELP option.)

5.16 HOSTNAME_RETRY_COUNT

Description

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

The Universal Control Manager will sleep for one second between connection attempts.

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

Usage

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

Value

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

[Default is *1*.]

5.17 INSTALLATION_DIRECTORY

Description

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

Usage

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

Value

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

HP NonStop

[Default is `$SYSTEM.UNVBIN`.]

UNIX

[Default is `/opt/universal/uctlmgr`.]

Windows

[Default is `c:\Program Files\Universal\uctlmgr`.]

5.18 MESSAGE_LANGUAGE

Description

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

Usage

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

Values

language is any UMC file provided by Stonebranch Inc.

There are different UMC files for different languages.

z/OS

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

HP NonStop, UNIX, and Windows

The first three characters of the language name are used as a three-character suffix in the UMC file base name. All UMC files have a .UMC extension.

IBM i

The first three characters of the language name are used as a three-character suffix in the UMC member base name **UCMMC**. UMC files are located in the source physical file **UNVPRD420/UNVNLS**.

[Default is *ENGLISH*.]

5.19 MESSAGE_LEVEL

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

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

Values

level indicates either of the following level of messages:

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

IBM i and z/OS

[Default is info.]

HP NonStop, UNIX, and Windows

[Default is warn.]

Trace Files

IBM i

Trace file name is ***CURLIB/UNVTRCUCT(UCTn)**, where **n** is the job number of the job invoking Universal Control.

HP NonStop

Trace file name is **UCTLTRC**. It is created in the working subvolume of Universal Control Manager.

UNIX and Windows

Trace file name is **uctl.trc**. It is created in the working directory of Universal Control Manager.

z/OS

Trace file is written to the data set referenced by the **UNVTRACE** ddname.

5.20 NETWORK_DELAY

Description

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

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

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

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

Usage

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

Values

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

[Default is *120*.]

5.21 NLS_DIRECTORY

Description

The NLS_DIRECTORY option specifies the directory in which Stonebranch Solutions NLS files are installed.

Usage

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

Values

directory is the directory in which NLS files are installed.

Defaults

UNIX

[Default is `/opt/universal/nls`.]

Windows

[Default is `..\nls`.]

5.22 OUTBOUND_IP

Description

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

Note: By default, the OUTBOUND_IP option is not set.

Usage

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

Values

host is the host or IP address.

5.23 PLF_DIRECTORY

Description

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

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

IBM i

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

Usage

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

Values

directory is the name of the PLF directory.

A full path name must be specified.

Defaults

UNIX

[Default is `/var/opt/universal/tmp`.]

IBM i

[Default is `/tmp`.]

5.24 PRIVATE_KEY

Description

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

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

z/OS

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

Usage

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

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.

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 (*lib*). If not, the library list ***LIBL** is searched for the first occurrence of the file name.

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

z/OS

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-private_key_pwd *password*			√	√	√
Environment Variable	UCTLPRIVATEKEYPWD=*password*	√		√	√	
Configuration File Keyword	privatekeypwd=*password*	√		√	√	√
STRUCT Parameter	PVTKEYPWD(*password*)	√				

Values

password is the password for the private key.

5.26 REFRESH_CMD

Description

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

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

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

Currently, only Universal Event Monitor Server can be specified, as it is the only component that accepts a REFRESH_CMD option.

Usage

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

Values

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

Currently, only *uems* (Universal Event Monitor Server) is a valid value for *compname*.

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

IBM i

Valid values for *option* are:

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

5.27 REMOTE_HOST

Description

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

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

Usage

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

Values

hostaddress is the IP address of the host computer.

The format of *hostaddress* can be either:

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

5.28 REMOTE_PORT

Description

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

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

Usage

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

Values

port is the TCP port on the remote computer.

The format of *port* can be either:

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

[Default is *7887*.]

5.29 SSL_IMPLEMENTATION

Description

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

Usage

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

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 Stonebranch Solutions 4.2.0 Installation Guide for a description of the prerequisites before using System SSL.)

[Default is openssl.]

5.30 START_CMD

Description

The START_CMD option specifies the name of the Stonebranch Solutions component that is to be started on the machine specified by the REMOTE_HOST option.

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

Note: In Stonebranch Solutions 4.2.0, an event-driven UEM Server is the only component that can be started with Universal Control. An event-driven UEM Server component has a component type of **`uems`**.

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

Usage

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

Values

compname is the name of the component to be started.

5.31 STOP_CMD

Description

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

A component ID is assigned to a component when Universal Broker starts it. The Universal Query utility can be used to obtain a list of active components (along with their component IDs) that are managed by an instance of Universal Broker. (See Chapter 25 Universal Query for information about Universal Query.)

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

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

Usage

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

Values

compID is the ID of the component to be terminated.

5.32 SYSTEM_ID

Description

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

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-system_id *ID*					√
Environment Variable	UCTLSYSTEMID=*ID*					√
Configuration File Keyword	n/a					
STRUCT Parameter	n/a					

Values

ID is the system identifier of the local Universal Broker.

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

5.33 USER_ID

Description

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

The Universal Control Server determines if this option is required.

Usage

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

Values

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

Note: *user* must be a valid user identifier on the remote computer.

5.34 USER_PASSWORD

Description

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

The password is always encrypted, regardless of how encryption is configured on the Universal Control Manager and Universal Control Server.

Usage

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

Values

password is the password for the user identifier.

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

5.35 VERIFY_HOST_NAME

Description

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

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

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

The following certificate fields are matched in the order listed:

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

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

Usage

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

Values

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

Valid values for *option* are:

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

[Default is no.]

5.36 VERIFY_SERIAL_NUMBER

Description

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

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

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

Usage

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

Values

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

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

5.37 VERSION

Description

The VERSION option writes the program version information and copyright.

Usage

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

Values

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

IBM i

Valid values for *option* are:

- YES
  Write program version information and copyright.
- NO
  Do not write program version information and copyright.

[Default is NO.]

CHAPTER 6

Universal Control Server Configuration Options

6.1 Overview

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

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

Information on how these options are used is documented in Chapter 4 Universal Control.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented or each option.

6.2 Configuration Options List

Table 6.1 identifies all Universal Control Server configuration options.

Option	Description	Page
ACTIVITY_MONITORING	Specification for whether or not product activity monitoring events are generated.	206
CODE_PAGE	Code page used for text translation.	207
EVENT_GENERATION	Events to be generated as persistent events.	208
INSTALLATION_DIRECTORY	Base directory in which Universal Control Server is installed.	210
LOGON_METHOD	Method of how users are logged onto the system.	211
MESSAGE_LEVEL	Level of messages printed.	212
NLS_DIRECTORY	Location of UMC and UTT files.	215
TMP_DIRECTORY	Directory name used for temporary files.	216
TRACE_DIRECTORY	Location of trace files.	217
USER_SECURITY	Specification for whether or not user authentication is active.	217

Table 6.1 Universal Control Server Configuration Options

6.3 ACTIVITY_MONITORING

Description

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

Usage

Method	Syntax	IBM i	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 monitoring events.
- **no**
  Deactivate monitoring events.

[Default is no.]

6.4 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.

Usage

Method	Syntax	IBM i	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 (see Section 36.4 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 36.3 Character Code Pages for a complete list of character code pages provided by Stonebranch Inc. for use with Stonebranch Solutions.

6.5 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 Stonebranch Solutions components, see the Universal Event Subsystem 4.2.0 Event Definitions Guide.)

Usage

Method	Syntax	IBM i	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,X300
  Generate all event types except for 200 through 250 and 300.

[Default is *X (no event types).]**

6.6 INSTALLATION_DIRECTORY

Description

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

Usage

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

Values

directory is the location in which UCMD Server is installed.

Defaults

UNIX

[Default is `/opt/universal/uctlsrv`.]

Windows

[Default is `c:\Program Files\Universal\uctlsrv`.]

6.7 LOGON_METHOD

Description

The LOGON_METHOD option specifies the user's log on method.

If the UCMD Server is configured for user security (see the USER_SECURITY option), the log on method determines how the user is logged onto the Windows system.

If security is inactive, LOGON_METHOD is ignored.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	logon *option*				√	

Values

option is the user's log on method.

Valid values for *option* are:

- **batch**
  Windows log on type is **`batch`**. A batch log on prevents the command from interacting with the desktop. The user ID logging on as a batch user requires the Windows User Right "Log on as a batch job." If the user does not have this right, the log on action will fail.
- **interactive**
  Windows log on type is **`interactive`**. An interactive log on permits the command to interact with the desktop. No additional rights are required for a user to log on as an interactive user.

[Default is interactive.]

6.8 MESSAGE_LEVEL

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	message_level *level*[	√	√	√	√	√

Values

level indicates either of the following level of messages:

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

  Note: Use **trace** only as directed by Stonebranch, Inc. Customer Support.
- **audit**
  Writes audit, informational, warning, and error messages.
- **info**
  Writes informational, warning, and error messages.
- **warn**
  Writes warning and error messages.
- **error**
  Writes error messages only.

[Default is info.]

Trace Files

IBM i

Trace file name is ***CURLIB/UNVTRCUCT(UCTn)**, where **n** is the Universal Server job number under which the Universal Control Server program is running. ***CURLIB** is the temporary library designated during the Stonebranch Solutions installation process; the default temporary library is **UNVTMP420**.

HP NonStop

Trace file name is **uctXXXXX**, where:

- **XXXXX** is the last five decimal values of the component ID of the Universal Control Server.

The trace file is created in the **$SYSTEM.UNVTRACE** subvolume.

UNIX and Windows

Trace file name is **uctsrv-N.trc**, where:

- **N** is the process ID of the Universal Control Server.

It is created in the working directory of the Universal Control Server.

z/OS

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.UCT.Dyymmdd.Thhmmss.Cnnnnnnn**, 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.
   - **nnnnnnn** is the last seven digits of the Server's component ID in hexadecimal format.

   Each time that a server is restarted, its sequence number is incremented. If a server is restarted more than 15 times, tracing is disabled.
2. If ddname **UNVTRMDL** is not defined in the UBROKER started task procedure, member name **Tnnnnnn** is created in the PDS or PDS/E allocated to the **UNVTRACE** ddname, where:
   - **nnnnnnn** is the last seven digits of the Server's component ID in hexadecimal format.
   - **s** is the component ID's sequence number from `0` - `F`.

   Each time that a server is restarted, its sequence number is incremented. If a server is restarted more than 15 times, tracing is disabled.

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

6.9 NLS_ DIRECTORY

Description

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

Usage

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

Values

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

Full path names are recommended.

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

Defaults

UNIX

[Default is `/opt/universal/nls`.]

Windows

[Default is `..\nls`.]

6.10 TMP_DIRECTORY

Description

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

Usage

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

Values

directory is the name of the directory.

It should specify a fully qualified path name.

Defaults

UNIX

Default is */var/opt/universal/tmp*.

Windows

[Default is *..\tmp*.

z/OS

[Default is `/tmp`.]

6.11 TRACE_DIRECTORY

Description

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

Usage

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

Values

directory is the name of the trace file directory.

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

Defaults

Windows

[Default is *C:\Program Files\Universal\UCtSrv*.]

UNIX

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

6.12 USER_SECURITY

Description

The USER_SECURITY option specifies whether or not to user security and, if so, the security method.

If user security is activated, the remote Universal Control Manager requesting command execution is required to supply a local user ID and password. The user's command is started as that user.

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	security *option*	√	√	√	√	√

Values

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

z/OS

- **default**
  Use z/OS SAF user authentication method. The user ID must have a OMVS segment.
- **none**
  No user security. Not recommended.

Windows

- **default**
  User-supplied user ID and password is authenticated against the user profile.
- **none**
  No user security. Not recommended.

UNIX

- **default**
  Use UNIX default user authentication method, **/etc/passwd**.
- **trusted**
  Use HP Trust Security authentication.
- **pam**
  Use the Pluggable Authentication Modules (PAM) interface.
- **none**
  No user security.

IBM i

- **default**
  Security is activated and uses IBM i authentication.
- **none**
  Security is not activated.

HP NonStop

- **default**
  Use HP NonStop default user authentication method, **SAFEGUARD**.
- **trusted**
  Use HP Trust Security authentication.
- **pam**
  Use the Pluggable Authentication Modules (PAM) interface.
- **none**
  No user security.

CHAPTER 7

Universal Control Component Definition Options

7.1 Overview

This chapter provides detailed information about the options that comprise Universal Control (UCTL) component definitions.

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

Information on how component definitions are used is documented in Chapter 4 Universal Control.

Section 7.2 Component Definition Options Information provides a guideline for understanding the information presented for each component definition option.

7.2 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	IBM i	Tandem	UNIX	Windows	z/OS
Component Definition Keyword	<Format / Value>					

Method

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

- Component Definition Keyword

Syntax

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

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

(Operating System)

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

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

Values

Identifies all possible values for the specified value type.

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

7.3 Component Definition Options

Table 7.1 identifies all of the options that can comprise a Universal Control component definition.

Component	Description	Page
AUTOMATICALLY_START	Specification for whether or not the UCTL Server starts automatically when Universal Broker is started.	224
COMPONENT_NAME	Name by which the clients know the UCTL Server.	225
CONFIGURATION_FILE *	Name of the UCTL Server's configuration file.	226
RUNNING_MAXIMUM	Maximum number of UCTL Servers that can run simultaneously.	227
START_COMMAND *	Program name of the UCTL Server.	228
WORKING_DIRECTORY *	Directory used as the working directory of the UCTL Server.	229
* These options are required in all component definitions.		

Table 7.1 Universal Control Component Definition Options

7.4 AUTOMATICALLY_START

Description

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

Note: AUTOMATICALLY_START is optional in a component definition.

Usage

Method	Syntax	IBM i	Tandem	UNIX	Windows	z/OS
Component Definition Keyword	auto_start *option*	√	√	√	√	√

Values

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

The only valid value for *option* is:

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

7.5 COMPONENT_NAME

Description

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

Component start requests refer to Universal Control Server by this name.

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

Usage

Specification Method	Syntax	IBM i	Tandem	UNIX	Windows	z/OS
Component Definition Keyword	component_name *name*	√	√	√	√	√

Values

name is the name of the Universal Control Server.

There is only one valid value for *name*: **`uctl`**.

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

Note: This name should not be changed.

7.6 CONFIGURATION_FILE

Description

The CONFIGURATION_FILE option specifies the name of the Universal Control Server configuration file.

Note: CONFIGURATION_FILE is required in a component definition.

Usage

Method	Syntax	IBM i	Tandem	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.

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 and Windows

Relative paths are relative to the component's working directory.

z/OS

Member names are located in the **UNVCONF** library allocated to the Universal Broker ddname **UNVCONF**. The installation default is **UTSCFG00**.

7.7 RUNNING_MAXIMUM

Description

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

If this maximum number is reached, any command received to start a UCTL Server is rejected.

Note: RUNNING_MAXIMUM is optional in a component definition.

Usage

Method	Syntax	IBM i	Tandem	UNIX	Windows	z/OS
Component Definition Keyword	running_max *maximum*	√	√	√	√	√

Values

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

[Default is *100*.]

7.8 START_COMMAND

Description

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

Optionally, START_COMMAND also can specify command line options.

Note: START_COMMAND is required in a component definition.

Usage

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

Values

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

options is the optional list of command line options.

z/OS

The program object must be in the Universal Broker's search order for loading program objects. The default location is the **SUNVLOAD** library allocated to the Broker's **STEPLIB** ddname.

HP NonStop, UNIX, Windows

name is the full path name of the Universal Control Server program.

IBM i

name is the Universal Command Server program. If the program name is non-qualified, the library list ***LIBL** is searched.

7.9 WORKING_DIRECTORY

Description

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

Note: WORKING_DIRECTORY is required in a component definition.

Usage

Method	Syntax	IBM i	Tandem	UNIX	Windows	z/OS
Component Definition Keyword	working_directory *directory*	√	√	√	√	√

Values

directory is the full path name of the working directory.

[Default is (.).

HP NonStop, UNIX, Windows

directory is the full path name of the directory Universal Control Server uses as its working directory.

z/OS

directory is the HFS directory name that the Universal Control Server uses as its working directory.

IBM i

working_directory serves as a required placeholder only.

Note: Do not change this directory.

CHAPTER 8

Universal Control UACL Entries

8.1 Overview

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

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

Information on how these UACL entries are used is documented in Chapter 4 Universal Control.

Section 8.2 UACL Entries Information provides a guideline for understanding the information presented for each UACL entry.

8.2 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	IBM i	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 Stonebranch Solutions 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:

- IBM i
- NonStop (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]**.

8.3 UACL Entries List

Table 8.1 identifies all Universal Control UACL Entries.

UACL Entry	Description	Page
UCTL_ACCESS	Allows or denies access to Universal Control Server services. There are two forms to this entry: • **`uctl_access`** • **`uctl_cert_access`**	234
UCTL_REQUEST	Allows or denies access to Universal Control Server services based on client identification and request type.	236

Table 8.1 Universal Control UACL Entries

8.4 UCTL_ACCESS

Description

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

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

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

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

A **`uctl_access`** UACL entry is matched if all of the following occur:

- Request comes from an IP address identified by *host*.
- Remote end is executing as user *remote_user*.
- Remote user is requesting to execute a command as local user *local_user*.

A **`uctl_cert_access`** UACL entry is matched if both of the following occur:

- Request comes from a client with a certificate identifier of *certid*.
- Remote user is requesting to execute a command as local user *local_user*.

The first matching rule is used to control access.

See the Indesca User Guide, Section 6.5.2 UACL Entries for details on *host*, *remote_user*, *local_user*, and *certid* specification syntax.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
UACL File Keyword	uctl_access *host,remote_user,local_user,access, auth* uctl_cert_access * *certid,local_user,access,auth*	√	√	√	√	√
* uctl_cert_access is not a valid form of UCTL_ACCESS for HP NonStop.						

Values

Valid values for *access* are:

- **deny**
  Service is denied. A message is returned to the remote end. The connection is closed.
- **allow**
  Service is accepted and processed.

Valid values for *auth* are:

- **auth**
  Local user account must be authenticated. The Manager must provide a proper password for the account.
- **noauth**
  User ID provided by the Manager does not have to match the user process being stopped.

Windows

To set **noauth** via the Universal Configuration Manager, de-select **`Require matching local user account`** when you are adding or editing an Access ACL (uctl_access) entry.

IBM i, HP NonStop, UNIX, z/OS

Additionally, **noauth** specifies that the local user account does not require user authentication. The Manager still must supply a password to satisfy command syntax rules, but it will not be verified. Any password value will suffice.

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

8.5 UCTL_REQUEST

Description

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

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

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

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

A **`uctl_request`** UACL entry is matched if all of the following occur:

- Request comes from an IP address identified by *host*.
- Remote end is executing as user *remote_user*.
- Remote user is requesting to execute a command as local user *local_user*.

A **`uctl_cert_request`** UACL entry is matched if both of the following occur:

- Request comes from a client with a certificate identifier of **`certid`**.
- Remote user is requesting to execute a command as local user **`local_user`**.

The first matching rule is used to control access.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
UACL File Keyword	uctl_request *host,remote_user,local_user, req_type,req_name,access,auth* uctl_cert_request *certid,local_user,req_type,req_name, access,auth*	√		√	√	√

Values

req_type specifies what type of request the Manager is requesting.

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

Valid values for *req_type* are:

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

  Note: **refresh** does not include a Universal Broker REFRESH command, which is not processed by any UACL entry.
- **start**
  Manager is requesting the start of the component.

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

  req_name value is blank.

See the Indesca User Guide, Section 6.5.2 UACL Entries for details on *host*, *remote_user*, *local_user*, and *certid* specification syntax.

Valid values for *access* are:

- **deny**
  Service is denied. A message is returned to the remote end. The connection is closed.
- **allow**
  Service is accepted and processed.

Valid values for *auth* are:

- **auth**
  Local user account must be authenticated. The Manager must provide a proper password for the account.
- **noauth**
  User ID provided by the Manager does not have to match the user process being stopped.

IBM i, HP NonStop, UNIX, z/OS

Additionally, **noauth** specifies that the local user account does not require user authentication. The Manager still must supply a password to satisfy command syntax rules, but it will not be verified. Any password value will suffice.

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

CHAPTER 9

Universal Copy

9.1 Overview

Universal Copy provides a means to copy files from either:

- Manager to a Server
- Server to Manager

9.1.1 Usage

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

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

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

9.2 Universal Copy for Windows and UNIX

This section describes the configuration options and command line syntax of Universal Copy for the Windows and UNIX operating systems.

9.2.1 Configuration Options

Table 9.1 identifies all Universal Copy for Windows and UNIX configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

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

Table 9.1 Universal Copy for Windows and UNIX - Configuration Options

9.2.2 Command Line Syntax

Figure 9.1, below, illustrates the syntax — using the long form of command line options — of Universal Copy for Windows and UNIX.

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

ucopy
{ -version | -help }
```

Figure 9.1 Universal Copy for Windows and UNIX - Command Line Syntax

9.2.3 Command Operands

FILE

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

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

z/OS USS

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

HFS files are specified simply as UNIX file names.

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

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

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

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

9.3 Universal Copy for IBM i

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

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

9.3.1 Stonebranch Solutions for IBM i Commands

The names of the Stonebranch Solutions for IBM i commands that are installed in the IBM i **QSYS** library are tagged with the Stonebranch Solutions for IBM i **v**ersion / **r**elease / **m**odification number, **420**. The names of the commands installed in the Stonebranch Solutions for IBM i product library, **UNVPRD420**, are untagged.

To maintain consistency across releases, you may prefer to use the untagged names in your production environment. The **UCHGRLS** (Change Release Tag) program lets you change the tagged command names in **QSYS** to the untagged command names in **UNVPRD420**.

(See the Stonebranch Solutions 4.2.0 Installation Guide for detailed information on **UCHGRLS**.)

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

9.3.2 Description

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

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

9.3.3 Configuration Options

Table 9.2, below, identifies all Universal Copy for IBM i configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
CPY_MODE	Copy mode for reading and writing files.
FRMFILE	Name of a file to copy.
FRMFILES	Names of additional files to copy.
FRMMBR	Name of a member in the file specified by FRMFILE.
INPUTMODE	Mode for opening the file for input.
MESSAGE_LEVEL	Level of messages displayed by Universal Copy.
OUTPUTMODE	Mode for opening the file for output.
REPLACE	Specification for whether or not existing output file should be replaced.
TOFILE	Name of the output file that receives the specified concatenated input files.
TOMBR	Name of a member within the file specified by TOFILE.
TRANSACTIONAL	Specification for whether or not the copy operation is performed in transactional mode.
VERSION	Writes the program version and copyright information.

Table 9.2 Universal Copy Configuration Options - IBM i

9.3.4 Command Line Syntax

Figure 9.2, below, illustrates the syntax — using the STRUCP parameter form of command line options — of the Universal Copy for IBM i.

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

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

STRUCP
VERSION(yes|no)
```

Figure 9.2 Universal Copy for IBM i - Command Line Syntax

9.4 Universal Copy for HP NonStop

This section describes Universal Copy command syntax and options for the HP NonStop operating system.

Currently, HP NonStop runs Universal Copy 2.1.1.
This section provides information for that version.

The Universal Copy program on the HP NonStop is strictly an OSS process. When used with UCMD Manager, the UCMD Manager's SERVER_OPTIONS option must specify a SCRIPT_TYPE option value of *OSS* (**`-server " -script_type OSS"`**) in order to indicate that this is an OSS process.

9.4.1 Configuration Options

Table 9.3, below, identifies all Universal Copy configuration options for HP NonStop.

Each **Option Name** is a link to detailed information about that configuration option.

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

Table 9.3 Universal Copy Configuration Options - HP NonStop

9.4.2 Command Line Syntax

Figure 9.3, below, illustrates the syntax — using the long form of command line options — of Universal Copy for HP NonStop.

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

ucopy
{ -version | -help }
```

Figure 9.3 Universal Copy for HP NonStop - Command Line Syntax

9.4.3 Command Operands

FILE

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

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

CHAPTER 10

Universal Copy Configuration Options

10.1 Overview

This chapter provides detailed information on the configuration options available for use with Universal Copy.

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

Information on how these options are used is documented in Chapter 9 Universal Copy.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

10.2 Configuration Options List

Table 10.1, below, identifies the Universal Copy configuration options.

Option Name	Description	Page
CPYMODE	Copy mode for reading and writing files	248
FRMFILE	Name of a file to copy	249
FRMFILES	Names of additional files to copy	250
FRMMBR	Name of a member in the file specified by FRMFILE	251
HELP	Writes a description of the command options and their format	252
INPUTMODE	Mode for opening the file for input	253
MESSAGE_LEVEL	Level of messages displayed by Universal Copy	254
MODE	Mode in which input files are read and output files are written	255
OUTPUT	File name that data is written to instead of standard output	256
OUTPUTMODE	Mode for opening the file for output	257
REPLACE	Specification for whether or not the file name specified with the OUTPUT option is replaced if it already exists	258
TOFILE	Output file that receives specified concatenated input files	259
TOMBR	Name of a member in the file specified by TOFILE	260
TRANSACTIONAL	Specification for whether or not the copy operation is performed in transactional mode	261
VERSION	Writes the program version and copyright information	262

Table 10.1 Universal Copy Configuration Options

10.3 CPY_MODE

Description

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

Usage

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

Values

mode is the copy mode for reading and writing files.

Valid values for *mode* are:

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

[Default is text.]

10.4 FRMFILE

Description

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

Usage

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

Values

file is the name of a file to copy.

Valid values for *file* are:

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

[Default is *stdin.]

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

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

10.5 FRMFILES

Description

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

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

Usage

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

Values

file is the name of an additional file to copy.

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

Valid values for *library* are:

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

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

Valid values for *member* are:

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

[Default is *first.]

10.6 FRMMBR

Description

The FRMMBR option specifies the name of a member in the file specified by FRMFILE.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	n/a					
STRUCP Parameter	FRMMBR (*member*)	√				

Values

member is the name of a member in the file.

Valid values for *member* are:

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

[Default is *first.]

10.7 HELP

Description

The HELP option writes a description of the command options and their format.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-h		√	√	√	
Command Line, Long Form	-help		√	√	√	
STRUCP Parameter	n/a					

Values

(There are no values for this option.)

10.8 INPUTMODE

Description

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

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

The default is to use the CPY_MODE option.

Usage

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

Values

'option' is the mode for opening the file.

10.9 MESSAGE_LEVEL

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

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

Values

level indicates either of the following level of messages:

- **trace**
  Writes trace messages used for diagnostic purposes.
  Note: Use **trace** only as directed by Stonebranch, Inc. Customer Support.
- **audit**
  Writes audit, informational, warning, and error messages.
- **info**
  Writes informational, warning, and error messages.
- **warn**
  Writes warning and error messages.
- **error**
  Writes error messages only.

IBM i

[Default is *info*.]

HP NonStop, UNIX, and Windows

[Default is *warn*.]

10.10 MODE

Description

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-m *mode*		√	√	√	
Command Line, Long Form	-mode *mode*		√	√	√	
STRUCP Parameter	n/a					

Values

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

Valid values for *mode* are:

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

[Default is binary.]

UNIX

There is no difference between binary and text.

HP NonStop

If the specified output file is within the Guardian file space, **TEXT** mode will generate an EDIT file with a file code of 101. If the specified output file is within the OSS file space, or the mode is set to **BINARY**, a C file with a file code of 180 will be generated.

10.11 OUTPUT

Description

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

Note: See FILE in the Command Operands sections of Chapter 9 Universal Copy for operating system-specific file naming conventions.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-o *filename*		√	√	√	
Command Line, Long Form	-output *filename*		√	√	√	
STRUCP Parameter	n/a					

Values

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

[Default is standard output.]

10.12 OUTPUTMODE

Description

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

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

The default is to use the CPY_MODE option.

Usage

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

Values

mode is the mode for opening the file.

10.13 REPLACE

Description

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

Usage

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

Values

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

Valid values for *option* are:

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

[Default is yes.]

10.14 TOFILE

Description

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	n/a					
STRUCP Parameter	TOFILE([*library/*] *file*)	√				

Values

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

Valid values for *file* are:

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

[Default is **stdout*.]

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

Valid values for library are:

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

10.15 TOMBR

Description

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	n/a					
STRUCP Parameter	TOMBR(*member*)	√				

Values

member is the name of a member in the file.

[Default is the file name.]

10.16 TRANSACTIONAL

Description

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-t *option*		√	√	√	
Command Line, Long Form	-transactional *option*		√	√	√	
STRUCP Parameter	n/a					

Values

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

Valid values for *option* are:

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

[Default is no.]

10.17 VERSION

Description

The VERSION option writes the program version and copyright information.

Usage

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

Values

HP NonStop, UNIX, and Windows

There are no values for this option.

IBM i

Valid values for *option* are:

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

[Default is no.]

CHAPTER 11

Universal Database Dump

11.1 Overview

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

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

UDBDUMP and the Universal Database Load (UDBLOAD) utility (see Chapter 13 Universal Database Load) are provided to enable recovery from a corrupted Berkeley database. Databases can potentially become corrupt due to system and address spaces ending abnormally.

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

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

11.1.1 Usage

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

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

11.2 Universal Database Dump for z/OS

This section describes Universal Database Dump (UDBDUMP) utility, specific to the IBM z/OS operating system.

11.2.1 JCL Procedure

Figure 11.1, below, illustrates the Universal Database Dump for z/OS JCL procedure (**UDBDPRC**, located in the **SUNVSAMP** library) that is provided to simplify the execution JCL and future maintenance.

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

Figure 11.1 Universal Database Dump for z/OS – JCL Procedure

11.2.2 DD Statements used in JCL Procedure

Table 11.1, below, describes the DD statements used in the Universal Database Dump for z/OS JCL procedure illustrated in Figure 11.1.

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

Table 11.1 Universal Database Dump for z/OS – DD Statements in JCL Procedure

11.2.3 JCL

Figure 11.2, below, illustrates the Universal Database Dump for z/OS JCL.

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

Figure 11.2 Universal Database Dump for z/OS – JCL

11.2.4 Configuration Options

Table 11.2 identifies the UDBDUMP for z/OS configuration options. It describes only those options relevant for database recovery.

For details on all options, see the Oracle documentation on the **db_dump** utility at URL:

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

Each **Option Name** is a link to detailed information about that option.

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

Table 11.2 Universal Database Dump for z/OS - Configuration Options

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

11.2.5 Command Line Syntax

Figure 11.4, below, illustrates the command line syntax of UDBDUMP for z/OS. It identifies only those options that are relevant for database recovery.

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

Figure 11.3 Universal Database Dump for z/OS - Command Line Syntax

11.3 Universal Database Dump for Windows and UNIX

This section describes Universal Database Dump (UDBDUMP) utility, specific to the Windows and UNIX operating systems.

11.3.1 Configuration Options

Table 11.3 identifies the UDBDUMP for Windows and UNIX configuration options. It describes only those options relevant for database recovery.

For details on all options, see the Oracle documentation on the **db_dump** utility at URL:

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

Each **Option Name** is a link to detailed information about that option.

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

Table 11.3 Universal Database Dump for Windows UNIX - Configuration Options

11.3.2 Command Line Syntax

Figure 11.4, below, illustrates the syntax of UDBDUMP for Windows and UNIX. It identifies only those options that are relevant for database recovery.

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

Figure 11.4 Universal Database Dump for Windows and UNIX - Command Line Syntax

CHAPTER 12

Universal Database Dump Configuration Options

12.1 Overview

This chapter provides detailed information on the configuration options available for use with Universal Database Dump (UDBDUMP). UDBDUMP is the Berkeley **db_dump** utility tailored specifically for Stonebranch Solutions databases.

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

Information on how these options are used is documented in Chapter 11 Universal Database Dump.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

12.2 Configuration Options List

Table 12.1, below, identifies the Universal Database Dump configuration options.

Option Name	Description	Page
DATABASE_FILE	Database file to be dumped.	270
DUMP_OPTIONS	Controls database dump behavior, including the format of the dump output and the extent to which data is recovered from a possibly corrupt database file.	271

Table 12.1 Universal Database Dump Configuration Options

12.3 DATABASE_FILE

Description

The DATABASE_FILE option specifies the database file to be dumped.

DATABASE_FILE is the last option specified on the command line.

z/OS

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	*database*			√	√	√

Values

database is the database file to be dumped.

12.4 DUMP_OPTIONS

Description

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

There are two forms of the DUMP_OPTIONS option:

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

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

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

Usage

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

Values

(There are no values used with this option.)

CHAPTER 13

Universal Database Load

13.1 Overview

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

The Universal Database Load (UDBLOAD) utility is the Berkeley **db_load** utility tailored specifically for Stonebranch Solutions databases.

UDBLOAD and the Universal Database Dump (UDBDUMP) utility (see Chapter 11 Universal Database Dump) are provided to enable recovery from a corrupted Berkeley databases. Databases can potentially become corrupt due to system and address spaces ending abnormally.

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

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

13.1.1 Usage

UDBLOAD invokes the Berkeley **db_load** utility. The UDBLOAD command line options are passed to **db_load**. UDBLOAD reads the database dump file and loads the contents into the specified database file.

The database dump file is created with UDBDUMP.

Note: By default, the load operation overwrites the specified database. To simply update the database, omit the OVERWRITE configuration option.

You should back up the database file prior to performing any load operation.

13.2 Universal Database Load for z/OS

This section describes Universal Database Load (UDBLOAD) utility, specific to the IBM z/OS operating system.

13.2.1 JCL Procedure

Figure 13.1, below, illustrates the Universal Database Load for z/OS JCL procedure (**UDBLPRC**, located in the **SUNVSAMP** library) that is provided to simplify the execution JCL and future maintenance.

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

Figure 13.1 Universal Database Load for z/OS – JCL Procedure

13.2.2 DD Statements used in JCL Procedure

Table 13.1, below, describes the DD statements used in the Universal Database Load for z/OS JCL illustrated in Figure 13.1.

ddname	Description
STEPLIB	Load library in which program UDBLOAD program is located.
UNVIN	Database dump file produced by Universal Database Dump.
SYSPRINT	UDBLOAD standard output ddname.
SYSOUT	UDBLOAD standard error ddname.
SYSIN	UDBLOAD standard input.

Table 13.1 Universal Database Load for z/OS – DD Statements in JCL Procedure

13.2.3 JCL

Figure 13.2, below, illustrates the Universal Database Load for z/OS JCL.

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

Figure 13.2 Universal Database Load for z/OS – JCL

13.2.4 Configuration Options

Table 13.2 identifies the UDBLOAD for z/OS configuration options. It describes only those options relevant for database recovery.

For details on all options, see the Oracle documentation on the **db_dump** utility at URL:

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

Each **Option Name** is a link to detailed information about that option.

Option Name	Description
DATABASE_FILE	Database file to be loaded.
OVERWRITE	Specification to overwrite the database file, not update it.

Table 13.2 Universal Database Load for z/OS - Configuration Options

Note: UDBLOAD accepts configuration options only on the PARM keyword of the **EXEC** statement.

13.2.5 Command Line Syntax

Figure 13.4, below, illustrates the command line syntax of UDBLOAD for z/OS. It identifies only those options that are relevant for database recovery.

```
[-o]
database
```

Figure 13.3 Universal Database Load for z/OS - Command Line Syntax

13.3 Universal Database Load for Windows and UNIX

This section describes Universal Database Dump (UDBLOAD) utility, specific to the Windows and UNIX operating systems.

13.3.1 Configuration Options

Table 13.3 identifies the UDBLOA for Windows and UNIX configuration options. It describes only those options relevant for database recovery.

For details on all options, see the Oracle documentation on the **db_dump** utility at URL:

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

Each **Option Name** is a link to detailed information about that option.

Option Name	Description
DATABASE_FILE	Database file to be loaded.
OVERWRITE	Specification to overwrite the database file, not update it. (Update is the default load operation.)

Table 13.3 Universal Database Load for Windows and UNIX - Configuration Options

13.3.2 Command Line Syntax

Figure 13.4, below, illustrates the syntax of UDBLOAD for Windows and UNIX. It identifies only those options that are relevant for database recovery.

```
udb_load
[-o]
database < dump
```

Figure 13.4 Universal Database Load for Windows and UNIX - Command Line Syntax

CHAPTER 14

Universal Database Load Configuration Options

14.1 Overview

This chapter provides detailed information on the configuration options available for use with Universal Database Load (UDBLOAD). UDBLOAD is the Berkeley **db_1oad** utility tailored specifically for Stonebranch Solutions databases.

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

Information on how these options are used is documented in Chapter 13 Universal Database Load.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

14.2 Configuration Options List

Table 14.1, below, identifies the Universal Database Load configuration options.

Option Name	Description	Page
DATABASE_FILE	Database file to be dumped.	279
OVERWRITE	Specification to recover as many records as possible from a possibly corrupt database file.	280

Table 14.1 Universal Database Load Configuration Options

14.3 DATABASE_FILE

Description

The DATABASE_FILE option specifies the database file to be loaded.

DATABASE_FILE is the last option specified on the command line.

z/OS

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	*database*			√	√	√

Values

database is the database file to be loaded.

14.4 OVERWRITE

Description

The OVERWRITE option specifies that the database file is to be overwritten, not updated.

To simply update the database, omit this option.

OVERWRITE is specific to Universal Database Load, not to the Berkeley **db_load** utility.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-o			√	√	√
Command Line, Long Form	n/a					

Values

(There are no values specified for this option.)

CHAPTER 15

Universal Display Log File

15.1 Overview

Universal Display Log File (**UDSPLOGF**) is a command for the IBM i environment.

Universal Display Log File reads job log output files that were created as a result of API **QMHCTLJL** or command **DSPJOBLOG**. The job log is formatted and written to standard output.

Optionally, **UDSPLOGF** can delete the job log file members after writing. File member deletion is controlled by the REMOVE_MEMBERS option. The default behavior is to leave the members unaltered.

Note: Universal Display Log File became available for the IBM i environment with PTF 0UC0114 (maintenance level 1.2.1.10).

15.2 Usage

Universal Display Log File consists of a command line program followed by a list of configuration options.

This section describes the configuration options and their command line syntax.

15.2.1 Stonebranch Solutions for IBM i Commands

The names of the Stonebranch Solutions for IBM i commands that are installed in the IBM i **QSYS** library are tagged with the Stonebranch Solutions for IBM i **v**ersion / **r**elease / **m**odification number, **420**. The names of the commands installed in the Stonebranch Solutions for IBM i product library, **UNVPRD420**, are untagged.

To maintain consistency across releases, you may prefer to use the untagged names in your production environment. The **UCHGRLS** (Change Release Tag) program lets you change the tagged command names in **QSYS** to the untagged command names in **UNVPRD420**.

(See the Stonebranch Solutions 4.2.0 Installation Guide for detailed information on **UCHGRLS**.)

This chapter references the IBM i commands by their untagged names. If you are using commands with tagged names to run Universal Display Log File, substitute the tagged names for the untagged names in these references.

15.2.2 Configuration Options

Table 15.1, below, identifies all Universal Display Log configuration options.

Each **Option Name** is a link to detailed information about that option.

Option Name	Description
PRIMARY_FILE	Name of the primary output file.
PRIMARY_MEMBER	Name of the primary output file member.
REMOVE_MEMBERS	Controls the deletion of job log output file members.
SECONDARY_FILE	Name of the secondary output file.
SECONDARY_MEMBER	Name of the secondary output file member.

Table 15.1 Universal Display Log File - Configuration Options

15.2.3 Command Line Syntax

Figure 15.1, below, illustrates the command line syntax of Universal Display Log File.

```
UDSPLOGF
[PRMRYFILE(filename[library]) [PRMRYMBR(member)] ]
[SCNDRYFILE(filename[library]) [SCNDRYMBR(member)] ]
[REMOVE({yes|no})]
```

Figure 15.1 Universal Display Log File - Command Line Syntax

CHAPTER 16

Universal Display Log File Configuration Options

16.1 Overview

This chapter provides detailed information on the configuration options available for use with Universal Display Log File.

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

Information on how these options are used is documented in Chapter 15 Universal Display Log File.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

16.2 Configuration Options List

Table 16.1, below, identifies the Universal Display Log File configuration options.

Option Name	Description	Page
PRIMARY_FILE	Name of the primary output file	286
PRIMARY_MEMBER	Name of the primary output file member	287
REMOVE_MEMBERS	Controls the deletion of job log output file members	288
SECONDARY_FILE	Name of the secondary output file	289
SECONDARY_MEMBER	Name of the secondary output file member	290

Table 16.1 Universal Display Log File - Configuration Options

16.3 PRIMARY_FILE

Description

The PRIMARY_FILE option specifies the name of the primary output file.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
UDSPJOGF Parameter	PRMRYFILE (*filename*[*library*])	√				

Values

filename is the name of the primary output file.

filename can be qualified by a *library* name.

16.4 PRIMARY_MEMBER

Description

The PRIMARY_MEMBER option specifies the name of the primary output file member.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
UDSPJOGF Parameter	PRMRYMBR (*member*)	√				

Values

member is the name of the primary output file member.

16.5 REMOVE_MEMBERS

Description

The REMOVE_MEMBERS option controls the deletion of job log output file members.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
UDSPJOGF Parameter	REMOVE (**option*)	√				

Values

**option* is the specification for controlling the deletion of the job log output file members.

Valid values for **option* are:

- **yes**
  Specified members will be deleted as a result of running this command.
- **no**
  Specified job log output members will not be deleted as a result of running this command.

[Default value is *no*.]

16.6 SECONDARY_FILE

Description

The SECONDARY_FILE option specifies the name of the secondary output file.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
UDSPJOGF Parameter	SCNDRYFILE (*filename*[*library*])	√				

Values

filename is the name of the secondary output file.

filename can be qualified by a *library* name.

16.7 SECONDARY_MEMBER

Description

The SECONDARY_MEMBER option specifies the name of the secondary output file member.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
UDSPJOGF Parameter	SCNDRYMBR (*member*)	√				

Values

member is the name of the secondary output file member.

CHAPTER 17

Universal Encrypt

17.1 Overview

The Universal Encrypt (**UENCRYPT**) utility encrypts the contents of command files into an unintelligible format for privacy reasons.

Stonebranch Solutions programs have the ability to read command line arguments from command files. Command files can be used to save private information, such as user identifiers and passwords. These files can be stored as clear text or encrypted text.

Command files that contain private information must be protected by using local file system security. This ensures that only authorized accounts have read access, regardless of whether or not the command files are encrypted.

Universal Encrypt adds an additional layer of security by encrypting the contents of command files. However, it should not be mistaken as a replacement for file system security.

The encrypted command file can be decrypted only by Stonebranch Solutions programs. No decrypt command is provided to decrypt the command file.

17.1.1 Usage

Universal Encrypt reads a command file from its standard input and writes an encrypted command file to its standard output.

The encrypted command file is a text file that can be used on any operating system by any Stonebranch Solutions component. Lines starting with a hash (**#**) character in column one are comments; blank lines are ignored.

Universal Encrypt performs operations specified by the command options.

17.2 Universal Encrypt for z/OS

This section describes Universal Encrypt program JCL, configuration options, and command line syntax for the z/OS operating system.

17.2.1 JCL

Figure 17.1, below, illustrates the Universal Encrypt for z/OS JCL.

```
//UENCRYPT EXEC PGM=UENCRYPT
//STEPLIB  DD  DISP=SHR,DSN=UNV.SUNVLOAD
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
//UNVIN    DD  DISP=SHR,DSN=MY.CLEAN.CMDFILE
//UNVOUT   DD  DISP=SHR,DSN=MY.ENCRYPT.CMDFILE
//SYSIN    DD  *
  -KEY DF#98AD@ -AES YES,/*
```

Figure 17.1 Universal Encrypt for z/OS – JCL

17.2.2 DD Statements used in JCL

Table 17.1, below, describes the DD statements used in the Universal Encrypt for z/OS JCL illustrated in Figure 17.1.

ddname	Description
STEPLIB	Load library in which program UENCRYPT is located.
SYSPRINT	UENCRYPT standard output ddname.
SYSOUT	UENCRYPT standard error ddname.
UNVIN	Clear text command file to encrypt.
UNVOUT	Encrypted command file.
SYSIN	UENCRYPT standard input from which parameters are read.

Table 17.1 Universal Encrypt for z/OS – DD Statements in JCL

17.2.3 Configuration

Universal Encrypt operations are controlled by the configuration options specified either on the PARM keyword of the **EXEC** statement or in the SYSIN ddname.

17.2.4 Configuration Options

Table 17.2, below, identifies the Universal Encrypt for z/OS configuration options.

Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AES	Specification for whether or not AES encryption is used.
CODE_PAGE	Character code page used to translate text data received and encrypted.
ENCRYPTION_KEY	Encryption key used by the encryption algorithm.
HELP	Prints a description of the command options and their format.
VERSION	Prints the program version and copyright information.

Table 17.2 Universal Encrypt for z/OS - Configuration Options

17.2.5 Command Line Syntax

Figure 17.2 illustrates the command line syntax – using the long form of command line options – of Universal Encrypt for z/OS.

```
[-key key]
[-codepage codepage]
[-aes {yes|no}]

uencrypt
{ -help | -version }
```

Figure 17.2 Universal Encrypt for z/OS - Command Line Syntax

17.3 Universal Encrypt for Windows and UNIX

This section describes Universal Encrypt configuration options and command line syntax for the Windows and UNIX operating systems.

17.3.1 Configuration Options

Table 17.3 identifies the Universal Encrypt for Windows and UNIX configuration options.

Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AES	Specification for whether or not AES encryption is used.
CODE_PAGE	Character code page used to translate text data received and encrypted.
ENCRYPTION_KEY	Encryption key used by the encryption algorithm.
HELP	Writes a description of the command options and their format.
NLS_DIRECTORY	Directory name where Universal Encrypt can find its code page tables.
VERSION	Writes the program version and copyright information.

Table 17.3 Universal Encrypt for Windows and UNIX - Configuration Options

17.3.2 Command Line Syntax

Figure 17.3, below, illustrates the command line syntax – using the long form of configuration options – of Universal Encrypt for Windows and UNIX.

```
uencrypt
[-key key]
[-codepage codepage]
[-nlsdir directory]
[-aes {yes|no}]

uencrypt
{ -help | -version }
```

Figure 17.3 Universal Encrypt for Windows and UNIX - Command Line Syntax

17.4 Universal Encrypt for IBM i

This section describes Universal Encrypt program configuration options and command line syntax for the IBM i operating system.

17.4.1 Stonebranch Solutions for IBM i Commands

The names of the Stonebranch Solutions for IBM i commands that are installed in the IBM i **QSYS** library are tagged with the Stonebranch Solutions for IBM i **v**ersion / **r**elease / **m**odification number, **420**. The names of the commands installed in the Stonebranch Solutions for IBM i product library, **UNVPRD420**, are untagged.

To maintain consistency across releases, you may prefer to use the untagged names in your production environment. The **UCHGRLS** (Change Release Tag) program lets you change the tagged command names in **QSYS** to the untagged command names in **UNVPRD420**.

(See the Stonebranch Solutions 4.2.0 Installation Guide for detailed information on **UCHGRLS**.)

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

17.4.2 Configuration Options

Table 17.4 identifies the Universal Encrypt for IBM i configuration options.

Each **Option Name** is a link to detailed information about that option.

Option Name	Description
AES	Specification for whether or not AES encryption is used.
CODE_PAGE	Character code page used to translate text data.
ENCRYPTION_KEY	Encryption key used to encrypt the input file.
INPUT_FILE	Input file that is to be encrypted.
OUTPUT_FILE	File to which the encrypted input file is written.
VERSION	Writes the program version information and copyright.

Table 17.4 Universal Encrypt for IBM i - Configuration Options

17.4.3 Command Line Syntax

Figure 17.4 illustrates the command line syntax – using the STRUEN parameter form of configuration options – of Universal Encrypt for IBM i.

```
STRUEN
[INFILE(input_file) [INMBR(member)] ]
[OUTFILE(output_file) [OUTMBR(member)] ]
[KEY(key)]
[AES(*{yes|no})]
[CODEPAGE(codepage)]

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

Figure 17.4 Universal Encrypt for IBM i - Command Line Syntax

Note: Options entered into plain text files or encrypted files must be in short form or long form syntax (see COMMAND_FILE_PLAIN and COMMAND_FILE_ENCRYPTED in the Universal Command 4.2.0 Reference Guide.)

17.5 Universal Encrypt for HP NonStop

This section describes Universal Encrypt configuration options and command line syntax for the HP NonStop operating system.

**Currently, HP NonStop runs Universal Encrypt 2.1.1.
This section provides information for that version.**

17.5.1 Configuration Options

Table 17.5, below, identifies the Universal Encrypt configuration options for HP NonStop.

Each **Option Name** is a link to detailed information about that option.

Option Name	Description
ENCRYPTION_KEY	Encryption key to use in the encryption algorithm
HELP	Writes a description of the command options and their format
VERSION	Writes the program version and copyright information

Table 17.5 Universal Encrypt for HP NonStop - Configuration Options

17.5.2 Command Line Syntax

Figure 17.5, below, illustrates the command line syntax – using the long form of configuration options – of Universal Encrypt for HP NonStop.

```
uencrypt
[-key key]

uencrypt
{ -help | -version }
```

Figure 17.5 Universal Encrypt for HP NonStop - Command Line Syntax

CHAPTER 18

Universal Encrypt Configuration Options

18.1 Overview

This chapter provides detailed information on the configuration options available for use with Universal Encrypt.

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

Information on how these options are used is documented in Chapter 17 Universal Encrypt.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

18.2 Configuration Options List

Table 18.1, below, identifies the Universal Encrypt configuration options.

Option Name	Description	Page
AES	Specification for whether or not AES encryption is used	300
CODE_PAGE	Character code page used to translate text data received and encrypted	301
ENCRYPTION_KEY	Encryption key used by the encryption algorithm	302
HELP	Writes a description of the command options and their format	303
INPUT_FILE	Input file that is to be encrypted	304
NLS_DIRECTORY	Directory name where Universal Encrypt can find its code page tables	305
OUTPUT_FILE	File to which the encrypted input file is written	306
VERSION	Writes the program version and copyright information	307

Table 18.1 Universal Encrypt Configuration Options

18.3 AES

Description

The AES option specifies whether or not AES encryption is used.

Note: Although the default - for backward compatibility - is not to use AES encryption, the use of AES encryption is highly recommended.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-a *option*			√	√	√
Command Line, Long Form	-aes *option*			√	√	√
STRUEN Parameter	AES(**option*)	√				

Values

option is the specification for whether or not to use AES encryption.

Valid values for *option* are:

- **yes**
  Use AES 256-bit encryption in CBC mode.
- **no**
  Use DES 56-bit encryption.

[Default is NO.]

18.4 CODE_PAGE

Description

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-t *codepage*			√	√	√
Command Line, Long Form	-codepage *codepage*			√	√	√
STRUEN Parameter	CODEPAGE(*codepage*)	√				

Values

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

codepage references a Universal Translate Table (UTT) file provided with the product (see Section 36.4 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 36.3 Character Code Pages for a complete list of character code pages provided by Stonebranch Inc. for use with Stonebranch Solutions.

18.5 ENCRYPTION_KEY

Description

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

Note: The key specified by this ENCRYPTION_KEY option also must be provided to the Stonebranch Solutions component command for which the command file is intended.

If this option is not used, Universal Encrypt uses a default 8-byte key.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-K *key*		√	√	√	√
Command Line, Long Form	-key *key*		√	√	√	√
STRUEN Parameter	KEY(*key*)	√				

Values

key is the key used to encrypt the command file.

key can contain from 1 to 32 characters long. However, it is recommended that *key* contain at least 8 characters.

HP NonStop

key contains from 1 to 8 characters. If less than eight characters are provided, *key* is padded on the right with null characters (binary zeros). It is recommended that all eight bytes be provided.

18.6 HELP

Description

The HELP option writes a description of the command options and their format.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-h		√	√	√	√
Command Line, Long Form	-help		√	√	√	√
STRUEN Parameter	n/a					

Values

(There are no values used with this option.)

18.7 INPUT_FILE

Description

The INPUT_FILE option specifies the input file that is to be encrypted.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	n/a					
STRUEN Parameter	INFILE(*input_file*) [INMBR (*member*)]	√				

Values

input_file is the name of the file to be encrypted.

Valid values for *input_file* are:

- **STDIN*
  Input file is read from standard input. Standard input is allocated to the workstation for interactive jobs and to file **QINLINE** for batch jobs. If executed as an interactive job, an ILE session terminal is displayed, from which input can be entered at the terminal.
- *file_name*
  Input file is read from the specified file.

  file_name can be qualified by a library name. Otherwise, library list ***LIBL** is searched for the first occurrence of the file name.

  A member name can be used for further qualification by specifying the **INMBR** parameter.

18.8 NLS_DIRECTORY

Description

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-nlsdir *directory*			√	√	
STRUEN Parameter	n/a					

Values

directory is the name of the directory.

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

18.9 OUTPUT_FILE

Description

The OUTPUT_FILE option specifies the file to which the encrypted input file is written.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	n/a					
STRUEN Parameter	OUTFILE(*output_file*) [OUTMBR(*member*)]	√				

Values

output_file is the name of the file which the input file is written.

Valid values for *output_file* are:

- **STDOUT*
  Encrypted input file is written to stdout. Standard output is allocated to the terminal if STRUEN is executed interactively. The ILE session terminal is displayed to view the output. Standard output is allocated to file **QPRINT** if STRUEN is executed in batch.
- *file_name*
  Encrypted input file is written to the specified file.

  file_name can be qualified by a library name. Otherwise, the library list ***LIBL** is searched for the first occurrence of the file name. If the file is not found, it is created as a physical source file with a record length of 266. If ***LIBL** is specified or implied, the file is created in **QGPL**.

  A member name can be used for further qualification by specifying the **OUTMBR** parameter.

18.10 VERSION

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-v		√	√	√	√
Command Line, Long Form	-version		√	√	√	√
STRUEN Parameter	VERSION(*option*)	√				

Values

HP NonStop, UNIX, Windows, and z/OS

There are no values for this option.

IBM i

Valid values for *option* are:

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

[Default is no.]

CHAPTER 19

Universal Event Log Dump

19.1 Overview

Universal Event Log Dump (UELD) is a utility that selects records from one of the Windows event logs and writes them to a specified output file.

All records from a log can be dumped, or event records can be selected according to the date and time that they were generated.

UELD can be run any time as a stand-alone application. It also is designed to work with Universal Command, which provides centralized control from any operating system and additional options for redirecting output.

19.2 Usage

Universal Event Log Dump (UELD) consists of the command line program (**`ueld`**) followed by a list of configuration options. This section describes the command line input; that is the configuration options and their syntax on the command line.

19.2.1 Configuration

Configuration consists of:

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

Configuration options are read from the following sources:

1. Command line
2. Configuration file

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

The configuration file, **`ueld.conf`**, provides the simplest method of specifying configuration options whose values will not change with each command invocation. These default values are used if the options are not read from one or more other sources.

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

19.2.2 Configuration Options

This section describes the configuration options used to execute UELD.

Configuration Options Categories

Table 19.1, below, categorizes the options into logical areas of application.

Category	Description
Local	Options required for local broker registration.
Log	Event records to select from which event log and what actions should be taken.
Messages	Utility message options.
Miscellaneous	Options use to display command help and program versions.
Output	How the event log records are printed.

Table 19.1 Universal Event Log Dump - Configuration Options Categories

The Universal Event Log Dump options for each category are summarized in the following tables.

Each **Option Name** is a link to detailed information about that option.

Local Category Options

Option Name	Description
INSTALLATION_DIRECTORY	Base directory where product is installed.
NLS_DIRECTORY	UMC and UTT file directory.

Log Category Options

Option Name	Description
BACKUP_LOG	Causes the log to be backed up before it is cleared.
CLEAR_ LOG	Causes the records in the log to be deleted from the log.
END_TIME	Ending date and time.
LOG_TYPE	Event log to be dumped.
REMOTE_SERVER	Name of a remote computer from which event log records should be retrieved.
START_TIME	Starting date and time.

Message Category Options

Option Name	Description
LOG_DIRECTORY	Log file directory.
MESSAGE_DESTINATION	Location where messages are written.
MESSAGE_LANGUAGE	Universal Message Catalog (UMC) file that will be used to write messages.
MESSAGE_LEVEL	Level of messages that will be displayed.

Miscellaneous Category Options

Option Name	Description
HELP	Writes a description of the command options and their format.
VERSION	Writes the program version and copyright information.

Output Category Options

Option Name	Description
OUTPUT_FILE	Complete path to the file that will be used to store the selected event log records.
PAGE_HEADERS	Enables or disables the printing of page headers.
PAGE_LENGTH	Number of lines that should be printed on each page.

19.2.3 Command Line Syntax

Figure 19.1 illustrates the command line syntax – using the long form of command line options – of UELD.

```
ueld
[-logtype {system|application|security}]
[-clear [-backup filename] ]
[-stime startdate [,starttime] ]
[-etime enddate [,endtime] ]
[-server servername]
[-file filename]
[-header {yes|no}]
[-length pagelength]
[-level {trace|audit|info|warn|error}]
[-dest {stderr|logfile}]
[-lang language]

ueld
{ -help | -version }
```

Figure 19.1 Universal Event Log Dump - Command Line Syntax

Configuration consists of setting default options and preferences. This section describes

CHAPTER 20

Universal Event Log Dump Configuration Options

20.1 Overview

This chapter provides detailed information on the configuration options available for use with Universal Event Log Dump.

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

Information on how these options are used is documented in Chapter 19 Universal Event Log Dump.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

20.2 Configuration Options List

Table 20.1 identifies the Universal Event Log Dump configuration options.

Option Name	Description	Page
BACKUP_LOG	Causes the log to be backed up before it is cleared.	315
CLEAR_LOG	Causes the records in the log to be deleted from the log	316
END_TIME	Ending date and time	317
HELP	Writes a description of command options and their format	318
LOG_DIRECTORY	Log file directory	320
LOG_TYPE	Event log to be dumped	321
MESSAGE_DESTINATION	Location where messages are printed.	322
MESSAGE_LANGUAGE	Universal Message Catalog (UMC) file that will be used to write messages.	323
MESSAGE_LEVEL	Level of messages that will be displayed	324
INSTALLATION_DIRECTORY	Base directory where product is installed	319
NLS_DIRECTORY	UMC and UTT file directory	325
OUTPUT_FILE	Complete path to the file that will be used to store the selected event log records	326
PAGE_HEADER	Enables or disables the printing of page headers	327
PAGE_LENGTH	Number of lines that should be printed on each page.	328
REMOTE_SERVER	Name of a remote computer from which event log records should be retrieved	329
START_TIME	Starting date and time	330
VERSION	Writes the program version and copyright information	331

Table 20.1 Universal Event Log Dump - Configuration Options

20.3 BACKUP_LOG

Description

The BACKUP_LOG option causes the log (the System event log or the log specified by the LOG_TYPE option) to be backed up before it is cleared via the CLEAR_ LOG option.

BACKUP_LOG is valid only when used with CLEAR_ LOG.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-b *filename*				√	
Command Line, Long Form	-backup *filename*				√	
Configuration File Keyword	n/a					

Values

filename is the name of the log to be backed up.

20.4 CLEAR_ LOG

Description

The CLEAR_LOG option causes the records in the log (the System event log or the log specified by the LOG_TYPE option) to be deleted from the log.

When CLEAR_LOG used with other options (except the BACKUP_LOG option), the log first is dumped, then cleared.

Note: Administrator access is required to dump any of the event logs.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-c				√	
Command Line, Long Form	-clear				√	
Configuration File Keyword	n/a					

Values

(There are no values for this option.)

20.5 END_TIME

Description

The END_TIME option specifies the ending date and time of records to be dumped.

If END_TIME is not used, Universal Event Log Dump will select records up to and including the last (that is, the most recent) record in the log specified by the LOG_TYPE option.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-e *enddate* [,*endtime*]				√	
Command Line, Long Form	-etime *enddate* [,*endtime*]				√	
Configuration File Keyword	n/a					

Values

enddate is the ending date of records to be dumped.

Event log records for the current date can be selected by using an asterisk (*).

If Event Log Dump will run over consecutive days, a rolling date can be specified with an asterisk (*) followed by a negative value. For example, **-2* selects records that were generated prior to 2 days before the current date.

endtime, optionally, specifies the ending time of records to be dumped.

If *endtime* is omitted, a value of *23:59* is used.

Note: *enddate* and *endtime* must match the short date and short time styles, respectively, as specified in the Control Panel.

If a 12-hour time format is used, and *endtime* is specified, *enddate* and *endtime* together must be enclosed in double (**"**) quotation marks (for example, *"12/31/1999, 11:59 PM"*). This ensures that the value will be read correctly from the command line.

20.6 HELP

Description

The HELP option writes a description of the command options and their format.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-h				√	
Command Line, Long Form	-help				√	
Configuration File Keyword	n/a					

Values

(There are no values for this option.)

20.7 INSTALLATION_DIRECTORY

Description

The INSTALLATION_DIRECTORY option specifies the Universal Event Log Dump base installation directory

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	n/a					
Configuration File Keyword	installation_directory *directory*				√	

Value

directory is the name of the Universal Event Log Dump base installation directory.

Specify a full path name.

20.8 LOG_DIRECTORY

Description

The LOG_DIRECTORY option specifies the directory name where log files are created. Log file creation is specified by the MESSAGE_DESTINATION option.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	n/a					
Configuration File Keyword	log_directory *directory*				√	

Value

directory is the directory where log files are created.

20.9 LOG_TYPE

Description

The LOG_TYPE option specifies the event log to be dumped.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-t *logtype*				√	
Command Line, Long Form	-logtype *logtype*				√	
Configuration File Keyword	logtype *logtype*				√	

Values

logtype is the event log to be dumped.

Valid values for *logtype* are:

- **system**
  Dumps records from the System event log.
- **application**
  Dumps records from the Application event log.
- **security**
  Dumps records from the Security event log.

  Note:Administrator access is required for this dump.

[Default is system.]

20.10 MESSAGE_DESTINATION

Description

The MESSAGE_DESTINATION option specifies the location where messages are to be written.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-d *destination*				√	
Command Line, Long Form	-dest *destination*				√	
Configuration File Keyword	message_dest *destination*				√	

Values

destination is the location where messages are to be written.

Valid values for destination are:

- **stderr**
  Writes the messages to the console.
- **logfile**
  Write the messages to a log file. The log file location is located beneath the **`ue1d`** folder, in the log folder. The current log file name is **`unv.1og`**.

  Past generation log files are named **`unvNNNN.1og`**, where **`NNNN`** equals the generation number. Currently, five generations are kept.

[Default value is stderr.]

20.11 MESSAGE_LANGUAGE

Description

The MESSAGE_LANGUAGE option specifies the Universal Message Catalog (UMC) file that will be used to write messages.

Each UMC file contains messages for a specific language.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-L *language*				√	
Command Line, Long Form	-lang *language*				√	
Configuration File Keyword	language *language*				√	

Values

language is the UMC file that will be used to write messages.

The first three characters of the language name are used as a three-character suffix in the UMC file base name (for example, **`uelmceng.umc`**). All UMC files have a **`.umc`** extension.

[Default is `uelmceng.umc`.]

20.12 MESSAGE_LEVEL

Description

The MESSAGE_LEVEL option specifies the level of messages to be written.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l *level*				√	
Command Line, Long Form	-level *level*				√	
Configuration File Keyword	n/a					

Values

level is the level of messages to be written.

Valid values for *level* are:

- **trace**
  Writes trace messages used for debugging. The trace file name is **`ueld.trc`**. It is created in the directory where the **`ueld`** program is located. Use only as directed by Stonebranch Technical Support.

  Note: Use **trace** only as directed by Stonebranch, Inc. Customer Support.
- **audit**
  Writes audit, informational, warning and error messages.
- **info**
  Writes informational, warning and error messages.
- **warn**
  Writes warning and error messages.
- **error**
  Writes error messages only.

[Default is warn.]

20.13 NLS_DIRECTORY

Description

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

Usage

Method	Syntax *	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	n/a					
Configuration File Keyword	nls_directory *directory*				√	

Values

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

20.14 OUTPUT_FILE

Description

The OUTPUT_FILE option specifies the complete path to the file that will be used to store the selected event log records.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-f *filename*				√	
Command Line, Long Form	-file *filename*				√	
Configuration File Keyword	n/a					

Values

filename is the complete path to the file that will be used to store the selected event log records.

20.15 PAGE_HEADERS

Description

The PAGE_HEADERS option enables or disables the writing of page headers.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-r *option*				√	
Command Line, Long Form	-header *option*				√	
Configuration File Keyword	header *option*				√	

Values

option is the specification for enabling or disabling the writing of page headers.

Valid values for *option* are:

- **yes**
  Display column headings at the top of each page.
- **no**
  Display report columns without headings.

{Default value is no.]

20.16 PAGE_LENGTH

Description

The PAGE_LENGTH option specifies the number of lines (records) to be written on each page.

If PAGE_LENGTH is not used, and PAGE_HEADERS is set to *YES*, a header will be written only at the top of the first page (since there is, in effect, only one page).

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-n *pagelength*				√	
Command Line, Long Form	-length *pagelength*				√	
Configuration File Keyword	length *pagelength*				√	

Values

pagelength is the number of lines (records) to be written on each page.

20.17 REMOTE_SERVER

Description

The REMOTE_SERVER option specifies the name of a remote computer from which event log records should be retrieved.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-S *servername*				√	
Command Line, Long Form	-server *servername*				√	
Configuration File Keyword	n/a					

Values

servername is the name of a remote computer from which event log records should be retrieved.

servername must be specified using the Universal Naming Convention (UNC) format, where the computer name is preceded by two back slashes (for example, **\\RMT1**).

20.18 START_TIME

Description

The START_TIME option specifies the starting date and time of records to be dumped.

If START_TIME is not used, Universal Event Log Dump will start with the first (that is, the earliest) record in the log specified by the LOG_TYPE option.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-s *startdate* [,*starttime*]				√	
Command Line, Long Form	-stime *startdate* [,*starttime*]				√	
Configuration File Keyword	n/a					

Values

startdate is the starting date of records to be dumped.

Event log records for the current date can be selected by using an asterisk (*).

If Event Log Dump will run over consecutive days, a rolling date can be specified with an asterisk (*) followed by a negative value. For example, **-2* selects records that were generated in the previous 2 days.

starttime, optionally, specifies the starting time of records to be dumped.

If *starttime* is omitted, a value of *00:00* (midnight) is used.

Note: *enddate* and *endtime* must match the short date and short time styles, respectively, as specified in the Control Panel.

If a 12-hour time format is used, and *endtime* is specified, *enddate* and *endtime* together must be enclosed in double (**"**) quotation marks (for example, *"01/01/2000, 12:00 AM"*). This ensures that the value will be read correctly from the command line.

20.19 VERSION

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-v				√	
Command Line, Long Form	-version				√	
Configuration File Keyword	n/a					

Values

(There are no values for this option.)

CHAPTER 23

Universal Products Install Merge

23.1 Overview

The Universal Products Install Merge (UPIMERGE) utility merges options and values from one Stonebranch Solutions component configuration file or component definition file with another.

UPIMERGE runs automatically during Stonebranch Solutions installation upgrades on UNIX and Windows. During the install, UPIMERGE combines options and values from existing configuration and component definition files with the options and values in the most recent versions of those files (delivered with the distribution package).

The result of each merge is a single file, with preserved options and values residing alongside any new options and values that were introduced to support new Stonebranch Solutions features.

The Stonebranch Solutions (UNIX and Windows) and Universal Enterprise Controller (Windows only) distribution packages also install UPIMERGE. This makes UPIMERGE available at any time for recovering archived options and values and merging them with the most recent options and values.

When used to update a Stonebranch Solutions configuration or component definition file, UPIMERGE must run with a user account that has write access to the output file. This typically means administrative access (that is, root on UNIX, Administrator on Windows).

23.2 Usage

As input, UPIMERGE typically uses an archived configuration file or component definition file. However, it can use any file as input, provided that the file is in standard keyword / value format.

UPIMERGE output is a file containing the options and values from the input file, merged with those in the output file. For each option in the output file, UPIMERGE replaces its value with the value of a matching option in the input file. If the input file contains options not defined in the output file, UPIMERGE adds those options to the end of the output file.

UPIMERGE does not attempt to sequence the entries that it adds to the output file. Thus, the order of options in the output file may not match the order of the same options in the input file.

If the input file contains more than one entry for an option, UPIMERGE adds every entry to the output file. The application will use the value of the last entry that appears in the output file.

UPIMERGE does not update any comments in the output file. Options that reside only in the output file are either commented out or left as is, depending on the command line parameters specified.

Note: UPIMERGE has no effect on a Stonebranch Solutions application's behavior if the local Universal Broker is in managed mode. In that environment, configurations and component definitions reside in a database file, not a text file. Use the I-Management Console application to manage configurations for managed installations.

23.2.1 Configuration Options

Table 23.1, below, identifies the Universal Products Install Merge configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
BACKUP_DESTINATION	Creates a copy of the original DESTINATION_FILE prior to the merge.
COMMAND_FILE_ENCRYPTED	Name of a file encrypted with Universal Encrypt that contains command options.
COMMAND_FILE_PLAIN	Name of a plain text file that contains command options.
COMPONENT_TYPE	Notifies UPIMERGE that the SOURCE_FILE is a component definition file that contains settings for the specified Stonebranch Solutions server component. You cannot use this option with CONFIGURATION_TYPE. UPIMERGE ignores this option if INSTALLATION_DIRECTORY is omitted.
CONFIGURATION_TYPE	Notifies UPIMERGE that the SOURCE_FILE is a configuration file that contains settings for the specified Stonebranch Solutions application. You cannot use this option with COMPONENT_TYPE. UPIMERGE ignores this option if INSTALLATION_DIRECTORY is omitted.
DESTINATION_FILE	Name of a file used to store the result of the merge.
ENCRYPTION_KEY	Key used to encrypt the file specified by COMMAND_FILE_ENCRYPTED.
HELP	Writes a description of the command options and their format.
INSTALLATION_DIRECTORY	Primary location in which the Stonebranch Solutions server component identified by COMPONENT_TYPE, or the Stonebranch Solutions application identified by CONFIGURATION_TYPE, resides.
KEEP_NOMATCH	Controls merge behavior when an option in DESTINATION_FILE has no match in SOURCE_FILE.
MESSAGE_LEVEL	Level of messages to write.
SOURCE_FILE	Name of a file used as input to the merge. If this parameter is omitted, UPIMERGE assumes input is redirected via stdin.
VERSION	Writes program version and copyright information.

Table 23.1 Universal Products Install Merge - Configuration Options

23.2.2 Command Line Syntax

Figure 23.1, below, illustrates the syntax – using the long form of command line options – of Universal Products Install Merge.

```
upimerge
-dest filename
[ -source filename ]
[ -installdir dirname { -cfgtype type [-comptype type] | -comptype type
   [-cfgtype type] } ]
[ -keep_nomatch {yes | no} ]
[ -bkup_dest {yes | no} ]
[ -file filename | -encryptedfile filename [-key key] ]
[ -level { trace | audit | info | warn | error }[,{ time | notime }]  ]

upimerge
{ -help | -version }
```

Figure 23.1 Universal Products Install Merge - Command Line Syntax

CHAPTER 24

Universal Products Install Merge Configuration Options

24.1 Overview

This chapter provides detailed information on the configuration options available for use with Universal Products Install Merge (UPIMERGE).

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

Information on how these options are used is documented in Chapter 23 Universal Products Install Merge.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

24.2 Configuration Options List

Table 24.1, below, identifies the Universal Products Install Merge configuration options.

Option Name	Description	Page
BACKUP_DESTINATION	Creates a copy of the original DESTINATION_FILE prior to the merge.	338
COMMAND_FILE_ENCRYPTED	Name of a file encrypted with Universal Encrypt that contains command options.	340
COMMAND_FILE_PLAIN	Name of a plain text file that contains command options.	341
COMPONENT_TYPE	Notifies UPIMERGE that the SOURCE_FILE is a component definition file that contains settings for the specified Universal server component. You cannot use this option with CONFIGURATION_TYPE. UPIMERGE ignores this option if INSTALLATION_DIRECTORY is omitted.	342
CONFIGURATION_TYPE	Notifies UPIMERGE that the SOURCE_FILE is a configuration file that contains settings for the specified Stonebranch Solutions application. You cannot use this option with COMPONENT_TYPE. UPIMERGE ignores this option if INSTALLATION_DIRECTORY is omitted.	344
DESTINATION_FILE	Name of a file used to store the result of the merge.	346
ENCRYPTION_KEY	Key used to encrypt the file specified by COMMAND_FILE_ENCRYPTED.	347
HELP	Writes a description of the command options and their format.	348
INSTALLATION_DIRECTORY	Primary location in which the Stonebranch Solutions server component identified with COMPONENT_TYPE or the Stonebranch Solutions application identified by CONFIGURATION_TYPE resides.	349
KEEP_NOMATCH	Controls merge behavior when an option in DESTINATION_FILE has no match in SOURCE_FILE.	350
MESSAGE_LEVEL	Level of messages to write.	352
SOURCE_FILE	Name of a file used as input to the merge. If this parameter is omitted, UPIMERGE assumes input is redirected via stdin.	353
VERSION	Writes the program version and copyright information.	355

Table 24.1 Universal Products Install Merge Configuration Options

24.3 BACKUP_DESTINATION

Description

The BACKUP_DESTINATION option instructs UPIMERGE to create a copy of the original configuration or component definition file specified by DESTINATION_FILE before merging the contents of SOURCE_FILE into it.

UPIMERGE creates the backup file in the same directory as the original configuration or component definition file.

Windows

UPIMERGE generates a file name with a format of **xxxnnnn.tmp**, where:

- **xxx** is a prefix based on the name of the Stonebranch Solutions configuration or component definition file specified for DESTINATION_FILE. (UPIMERGE uses a default of **uim** if it does not recognize the file name.)
- **nnnn** is a unique alpha-numeric identifier.

UNIX

The format of the file name UPIMERGE generates is implementation-dependent.

On most systems, the format is **xxxnnnnnn**, where

- **xxx** is a prefix based on the name of the Stonebranch Solutions configuration or component definition file specified for DESTINATION_FILE. (UPIMERGE uses a default of **uim** if it does not recognize the file name.)
- **nnnnnn** is a unique alpha-numeric identifier.

On some older UNIX systems, the format is undefined, but the name is still unique.

UPIMERGE reports the backup file it generates with message ID UNV4412I.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-bkup_dest *option*			√	√	
Environment Variable	UIMBKUPDEST=*option*			√	√	

Values

option specifies whether UPIMERGE creates a copy of the original configuration or component definition file.

Valid values for *option* are:

- **yes**
  Create a backup of the file specified with DESTINATION_FILE before merging the contents of SOURCE_FILE into it.
- **no**
  Merge the contents of SOURCE_FILE into DESTINATION_FILE, without saving a copy of the original DESTINATION_FILE.

[Default is no.]

24.4 COMMAND_FILE_ENCRYPTED

Description

The COMMAND_FILE_ENCRYPTED option specifies the file containing encrypted values for command line option parameters.

Command files specify an additional source of command line options. Storing options in a file is useful in situations where it is not desirable to specify them on the command line. The application reads the file and processes the options exactly like those specified on the command line. The options must be in their respective command line formats.

UPIMERGE can process encrypted or plain text command files (see the COMMAND_FILE_PLAIN option). Encrypted command files are an excellent place to store sensitive data that you may want to hide from the command line. As an added measure of security, Stonebranch, Inc. recommends that you configure the file system's security to protect any command file that contains sensitive data from unauthorized read access.

Use the Universal Encrypt utility provided with Stonebranch Solutions to encrypt a plain text command file. (For information on Universal Encrypt, see Chapter 17 Universal Encrypt). If Universal Encrypt used a key to encrypt the file, use that same key to read the file using the ENCRYPTION_KEY option.

Note: If UPIMERGE encounters the COMMAND_FILE_ENCRYPTED and COMMAND_FILE_PLAIN options on its command line, it uses the file specified for COMMAND_FILE_PLAIN.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-x *filename*			√	√	
Command Line, Long Form	-encryptedfile *filename*			√	√	
Environment Variable	n/a					

Values

filename is the name of the encrypted file that contains the command line options and their values.

24.5 COMMAND_FILE_PLAIN

Description

The COMMAND_FILE_PLAIN option specifies the file containing plain text values for command line option parameters.

Command files specify an additional source of command line options. Storing options in a file is useful in situations where it is not desirable to specify them on the command line. The application reads the file and processes the options exactly like those specified on the command line. The options must be in their respective command line formats.

UPIMERGE can process encrypted or plain text command files (see the COMMAND_FILE_ENCRYPTED option). Stonebranch, Inc. recommends that you configure the file system's security to protect command files from unauthorized access.

Note: If UPIMERGE encounters the COMMAND_FILE_ENCRYPTED and COMMAND_FILE_PLAIN options on its command line, it uses the file specified for COMMAND_FILE_PLAIN.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-f *filename*			√	√	
Command Line, Long Form	-file *filename*			√	√	
Environment Variable	n/a					

Values

filename is the name of the file that contains the command line options and their values.

24.6 COMPONENT_TYPE

Description

The COMPONENT_TYPE option notifies UPIMERGE that SOURCE_FILE is a component definition file that contains settings for the specified Universal server component.

UPIMERGE uses this value together with the INSTALLATION_DIRECTORY value to set options in a component definition file that depend on the Stonebranch Solutions server component's installed location.

Although COMPONENT_TYPE is not required to merge component definition files, using it ensures that UPIMERGE executes any custom merge logic necessary for a particular component's definition file.

Note: You cannot use COMPONENT_TYPE with the CONFIGURATION_TYPE option.

UPIMERGE ignores COMPONENT_TYPE if the INSTALLATION_DIRECTORY is omitted.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-comptype *type*			√	√	
Environment Variable	UIMCOMPTYPE=*type*			√	√	

Values

type identifies a Stonebranch Solutions server component.

Valid values for *type*, and the Stonebranch Solutions server component that each value represents, are:

- uac Universal Application Container Server
- ucmd Universal Command Server
- uctl Universal Control Server
- udm Universal Data Mover Server
- uemd Universal Event Monitor Server (demand-driven)
- uems Universal Event Monitor Server (event-driven)

[There is no default.]

24.7 CONFIGURATION_TYPE

Description

The CONFIGURATION_TYPE option notifies UPIMERGE that the file specified with the SOURCE_FILE option is a configuration file that contains settings for the specified Stonebranch Solutions application.

UPIMERGE uses this value together with the INSTALLATION_DIRECTORY value to set options in a configuration file that depend on the Stonebranch Solutions application's installed location.

Although CONFIGURATION_TYPE is not required to merge configuration files, using it ensures that UPIMERGE executes any custom merge logic necessary for a particular application's configuration file.

Note: You cannot use CONFIGURATION_TYPE with the COMPONENT_TYPE option.

UPIMERGE ignores CONFIGURATION_TYPE if the INSTALLATION_DIRECTORY option is omitted.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-cfgtype *type*			√	√	
Environment Variable	UIMCFGTYPE=*type*			√	√	

Values

type specifies the Stonebranch Solutions configuration file to merge.

Valid values for *type*, and the Stonebranch Solutions application that each value represents, are:

- uacl Universal Access Control List
- uacs Universal Application Container Server
- ubroker Universal Broker
- ucmd Universal Command Manager
- ucmds Universal Command Server
- uctl Universal Control Manager
- uctls Universal Control Server
- udm Universal Data Mover Manager
- udms Universal Data Mover Server
- uec Universal Enterprise Controller
- ueld Universal Event Log Dump Utility
- uem Universal Event Monitor Manager
- uems Universal Event Monitor Server
- uquery Universal Query

24.8 DESTINATION_FILE

Description

The DESTINATION_FILE option identifies the configuration or component definition file that UPIMERGE uses to store the results of the merge with SOURCE_FILE.

During the merge process, UPIMERGE replaces all values in DESTINATION_FILE that have a match in SOURCE_FILE. If SOURCE_FILE contains configuration or component definition options not defined in DESTINATION_FILE, UPIMERGE adds those options to the end of the output file.

UPIMERGE does not attempt to sequence the entries it adds to DESTINATION_FILE, which means the order of options in the output file may not match the order of the same options specified in SOURCE_FILE. In addition, UPIMERGE does not update any comments in the output file. Finally, if the input file contains more than one entry for a given option, UPIMERGE adds every occurrence of that entry to DESTINATION_FILE. The application uses the value of the last entry that appears in the output file.

Options that reside only in DESTINATION_FILE are either commented out or left as-is, depending on the value of the KEEP_NOMATCH option.

The file name specified for DESTINATION_FILE must exist (even if it is empty) before executing UPIMERGE.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-dest *filename*			√	√	
Environment Variable	UIMDESTFILE=*filename*			√	√	

Values

filename is the name of the file that contains the results of the merge with SOURCE_FILE.

filename can contain a complete path to the output file or a path relative to the current directory.

24.9 ENCRYPTION_KEY

Description

The ENCRYPTION_KEY option specifies the key used to encrypt the file specified with the COMMAND_FILE_ENCRYPTED option.

This key acts much like a password, preventing unauthorized users from decrypting the encrypted command file.

If Universal Encrypt used a key to encrypt the file, UPIMERGE requires the same key to decrypt the file. (For information on Universal Encrypt, see Chapter 17 Universal Encrypt).

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-K *key*			√	√	
Command Line, Long Form	-key *key*			√	√	
Environment Variable	UIMKEY=*key*			√	√	

Values

key is the key used to encrypt the command file.

24.10 HELP

Description

The HELP option writes a description of the command options and their format.

Usage

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

Values

(There are no values used with this option.)

24.11 INSTALLATION_DIRECTORY

Description

The INSTALLATION_DIRECTORY option specifies the installed location of the Stonebranch Solutions application or server component specified with the CONFIGURATION_TYPE or COMPONENT_TYPE option, respectively.

INSTALLATION_DIRECTORY is a command line-only parameter that UPIMERGE uses to set values for options in DESTINATION_FILE that depend upon the installed location of a specified Stonebranch Solutions application or server component.

It does NOT specify the installed location of the UPIMERGE utility.

Note: INSTALLATION_DIRECTORY requires the CONFIGURATION_TYPE or the COMPONENT_TYPE option.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-installdir *dirname*			√	√	
Environment Variable	UIMINSTALLDIR=*dirname*			√	√	

Values

dirname is the primary directory into which the Stonebranch Solutions application or server component specified by CONFIGURATION_TYPE or COMPONENT_TYPE resides.

24.12 KEEP_NOMATCH

Description

The KEEP_NOMATCH option instructs UPIMERGE to comment out any option in DESTINATION_FILE that does not exist in SOURCE_FILE.

The primary purpose of KEEP_NOMATCH is to prevent the introduction of new values for existing configuration options during a product upgrade.

For example, the typical sequence of events for an upgrade is as follows:

1. The install program archives a product's current configuration file to a well-known location.
2. The install program places a new configuration file from the distribution package into the well-known Stonebranch Solutions configuration file directory.
3. The install program passes the name of the archive file to UPIMERGE as SOURCE_FILE.
4. The install program passes the name of the newly-installed configuration file to UPIMERGE as DESTINATION_FILE.

If the archive file contains no value for a given configuration option – MESSAGE_LEVEL, for example – that means the corresponding program currently runs with the application-defined default. If the newly-installed configuration file were to contain an entry that sets MESSAGE_LEVEL to **audit**, and UPIMERGE kept this option by default, the result would be a change to the application's behavior (assuming the application-defined default for the option is something other than **audit**).

Note: As an additional precaution against a product upgrade changing an application's behavior, packaged configuration files have most options commented out. This approach allows product upgrades to "announce" the availability of new options while preserving an existing configuration.

On the other hand, a situation may arise after the install where it is desirable to introduce new options and/or values into a product's configuration. In this case, simply execute UPIMERGE with KEEP_NOMATCH set to **yes**.

While the discussion above focused on configuration files, UPIMERGE uses the same approach with component definition files.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-keep_nomatch *option*			√	√	
Environment Variable	UIMKEEPNOMATCH=*option*			√	√	

Values

option specifies whether UPIMERGE should "keep" or comment out any setting contained in DESTINATION_FILE that has no match in SOURCE_FILE.

Valid values for *option* are:

- **yes**
  Keep any option in DESTINATION_FILE without a match in SOURCE_FILE.
- **no**
  Comment out any option in DESTINATION_FILE with no match in SOURCE_FILE.

[Default value is no.]

24.13 MESSAGE_LEVEL

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l *level*			√	√	
Command Line, Long Form	-level *level*			√	√	
Environment Variable	UIMLEVEL=*level*			√	√	

Values

level indicates either of the following level of messages:

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

  Note:Use **trace** only as directed by Stonebranch, Inc. Customer Support.
- **audit**
  Writes audit, informational, warning, and error messages.
- **info**
  Writes informational, warning, and error messages.
- **warn**
  Writes warning and error messages.
- **error**
  Writes error messages only.

24.14 SOURCE_FILE

Description

The SOURCE_FILE option identifies the file that UPIMERGE merges into the configuration or component definition file specified by the DESTINATION_FILE option.

During the merge process, UPIMERGE replaces all values in DESTINATION_FILE that have a match in SOURCE_FILE.

If SOURCE_FILE contains configuration or component definition options not defined in DESTINATION_FILE, UPIMERGE adds those options to the end of the output file. If SOURCE_FILE contains more than one entry for a given option, UPIMERGE adds every occurrence of that entry to DESTINATION_FILE. The application uses the value of the last entry that appears in the output file.

Use of SOURCE_FILE is optional; it also is possible to provide input options and values to UPIMERGE using standard input (that is, stdin) redirection.

Figure 24.1, below, illustrates an example of stdin redirection:

```
upimerge -dest outfile.txt <infile.txt
```

Figure 24.1 Using stdin redirection with UPIMERGE

Note: It is possible to execute UPIMERGE without the redirected input file. However, the program may appear unresponsive. In this situation, UPIMERGE is actually waiting for an end-of-file indicator to signal the end of the redirected input.

Windows

Supply the end-of-file indicator by pressing <Ctrl+Z> <Enter>.

UNIX

Supply the end-of-file indicator by pressing <Ctrl+D>.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-source *filename*			√	√	
Environment Variable	UIMSOURCE=*filename*			√	√	

Values

filename is the name of the file that contains options and values that UPIMERGE merges into DESTINATION_FILE.

filename can contain a complete path to the input file or a path relative to the current directory.

24.15 VERSION

Description

The VERSION option writes the program version and copyright information.

Usage

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

Values

(There are no values for this option.)

CHAPTER 21

Universal Message Translator

21.1 Overview

Universal Message Translator (UMET) translates error messages into return (exit) codes based on a user-defined translation table.

Every command ends with a return code that indicates the success or failure of the command execution. Typically, a return code of 0 indicates success; all other codes indicate failure.

However, a small number of commands do not set their return code under failure conditions; instead, they issue error messages. Based on the user-defined translation table, Universal Message Translator translates these error messages into return codes.

21.2 Usage

UMET requires two input files:

1. Message Input file (user-specified or standard input) containing the error messages that are to be translated into a return codes.
2. Translation Table file containing the user-defined translation table that controls the error message-to-return code translation process.

To perform a translation, UMET:

1. Reads the messages in the input file.
2. Matches each line against the translation table entries.
3. Exits with an return code from the best match in the translation table.

If no match is found, UMET ends with return code 0.

UMET performs operations specified by the configuration options. This section describes each option and their syntax.

21.2.1 Translation Table

The translation table specifies:

- Text to search for.
- Return code associated with the text.
- Precedence when multiple matches are found.

Translation Table Format

The translation table consists of one or more lines.

Each line is either:

- Comment line (# in column one)
- Blank line (ignored)
- Translation table entry

Translation table entries consist of two fields separated by spaces or tabs. An entry cannot be continued onto multiple lines.

Translation Table Fields

The translation table entry fields are:

Field	Description
Message Mask	Selects which messages to match in the input file. The mask must be enclosed in double (**"**) quotation marks. Mask characters include the asterisks (*****) and the question mark (**?**). The asterisk matches 0 or more characters and the question mark matches one character. If an asterisk, question mark, or quotation mark is required in the message text, it must be preceded with a back slash (****). If a back slash is required in the message text, it must be preceded by another back slash.
Exit Code	Specifies an integer value that UMET exits with if this entry is the resulting match. The exit code is in the range of –99999 to 99999.

Figure 21.1 Universal Message Translator – Translation Table

21.2.2 Matching Algorithm

The input file is read line by line. For each line, the line is compared to each entry in the translation table. All the matching entries are saved.

After the entire input file is read, the matched entries from the translation table are sorted in ascending order by their line number in the translation table. The first entry in this sorted list is the resulting translation table entry. The exit code from the resulting translation table entry is used as the return code of UMET. If no matching entry is found, UMET exits with 0.

21.3 Universal Message Translator for z/OS

This section describes the Universal Message Translator (UMET) for the z/OS operating system.

21.3.1 JCL

Figure 21.2, below, illustrates the Universal Message Translator for z/OS JCL.

```
//UMET     EXEC PGM=UMET,PARM='-TABLE TABLE'
//STEPLIB  DD  DISP=SHR,DSN=UNV.SUNVLOAD
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
//TABLE    DD  DISP=SHR,MY.TRANS.TABLE
//SYSIN    DD  DISP=SHR,MY.MSG.FILE
```

Figure 21.2 Universal Message Translator for z/OS – JCL

UMET options are passed in with the PARM keyword on the EXEC statement.

21.3.2 DD Statements used in JCL Procedure

Table 21.1, below, describes the DD statements used in the Universal Message Translator for z/OS JCL illustrated in Figure 21.2.

ddname	Description
STEPLIB	Load library in which program UMET is located.
SYSPRINT	UMET standard output ddname.
SYSOUT	UMET standard error ddname.
TABLE	Translation table specified by the –table option on the PARM keyword.
SYSIN	Standard input ddname from which the message file is read.

Table 21.1 Universal Message Translator for z/OS – DD Statements in JCL

21.3.3 Configuration Options

Table 21.2 identifies the UMET for z/OS configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
HELP	Writes a description of the configuration options and their format.
IGNORE_CASE	Specification that matching of message masks to the input file is not case sensitive.
MESSAGE_FILE	Input message file name.
MESSAGE_LEVEL	Level of messages that will be written.
TRANSLATION_TABLE	Translation table file name.
VERSION	Writes the program version and copyright information.

Table 21.2 Universal Message Translator for z/OS - Configuration Options

21.3.4 Command Line Syntax

Figure 21.3, below, illustrates the syntax – using the long form of command line options – of UMET for z/OS.

```
umet
-table table
[-file messages]
[-ignorecase]
[-level {verbose|info|warn|error}]

umet
{ -help | -version }
```

Figure 21.3 Universal Message Translator for z/OS - Command Line Syntax

21.4 Universal Message Translator for Windows and UNIX

This section describes Universal Message Translator (UMET) for the Windows and UNIX operating systems.

21.4.1 Configuration Options

Table 21.3, below, identifies the UMET for Windows and UNIX configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
HELP	Writes a description of the command options and their format.
IGNORE_CASE	Specification that matching of message masks to the input file is not case sensitive.
MESSAGE_FILE	Input message file name.
MESSAGE_LEVEL	Level of messages that will be written.
TRANSLATION_TABLE	Translation table file name.
VERSION	Writes the program version and copyright information.

Table 21.3 Universal Message Translator for Windows and UNIX - Configuration Options

21.4.2 Command Line Syntax

Figure 21.4, below, illustrates the syntax — using the long form of command line options — of UMET for Windows and UNIX.

```
umet
-table table
[-file messages]
[-ignorecase]
[-level {verbose|info|warn|error}]

umet
{ -help | -version }
```

Figure 21.4 Universal Message Translator for Windows and UNIX - Command Line Syntax

21.5 Universal Message Translator for IBM i

This section describes Universal Message Translator (UMET) program for the IBM i operating system.

21.5.1 Return Codes

The resulting return code from the translation process is converted into an IBM i escape message.

The escape message ID and message severity depend on the return code value. as identified in Table 21.4, below.

Return Code	Message ID	Message Severity
1 – 10	UNV0344	10
11 – 20	UNV0345	20
21 – 30	UNV0346	30
31 and higher	UNV0347	40

Table 21.4 Universal Message Translator for IBM i - Return Codes

21.5.2 Stonebranch Solutions for IBM i Commands

The names of the Stonebranch Solutions for IBM i commands that are installed in the IBM i **QSYS** library are tagged with the Stonebranch Solutions for IBM i **v**ersion / **r**elease / **m**odification number, **420**. The names of the commands installed in the Stonebranch Solutions for IBM i product library, **UNVPRD420**, are untagged.

To maintain consistency across releases, you may prefer to use the untagged names in your production environment. The **UCHGRLS** (Change Release Tag) program lets you change the tagged command names in **QSYS** to the untagged command names in **UNVPRD420**.

(See the Stonebranch Solutions 4.2.0 Installation Guide for detailed information on **UCHGRLS**.)

This section references the IBM i commands by their untagged names. If you are using commands with tagged names to run Universal Message Translator, substitute the tagged names for the untagged names in these references.

21.5.3 Configuration Options

Table 21.5, below, identifies the UMET for IBM i configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
IGNORE_CASE	Specification that matching of message masks to the input file is not case sensitive.
MESSAGE_FILE	Input message file name.
MESSAGE_LEVEL	Level of messages that will be written.
TRANSLATION_TABLE	Translation table file name.

Table 21.5 Universal Message Translator for IBM i - Configuration Options

21.5.4 Command Line Syntax

Figure 21.5 illustrates the syntax — using the STRUME parameter form of command line options — of UMET for IBM i.

```
STRUME
TBL([library/]{umetbl|filename}) [TBLMBR(member)]
[MSGFILE([library/]{*stdin|filename}) [MSGMBR(member)] ]

**Additional Options**
[MSGLEVEL(*{verbose|info|warn|error})]
[IGNORECASE({yes|no})]
```

Figure 21.5 Universal Message Translator for IBM i - Command Line Syntax

21.6 Universal Message Translator for HP NonStop

This section describes Universal Message Translator (UMET) for the HP NonStop operating system.

21.6.1 Configuration Options

Table 21.6, below, identifies the UMET for HP NonStop configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
HELP	Writes a description of the command options and their format.
IGNORE_CASE	Specification that matching of message masks to the input file is not case sensitive.
MESSAGE_FILE	Input message file name.
MESSAGE_LEVEL	Level of messages that will be written.
TRANSLATION_TABLE	Translation table file name.
VERSION	Writes the program version and copyright information.

Table 21.6 Universal Message Translator for HP NonStop - Configuration Options

21.6.2 Command Line Syntax

Figure 21.6, below, illustrates the syntax – using the long form of command line options – of UMET for HP NonStop.

```
umet
-table table
[-file messages]
[-ignorecase]
[-level {verbose|info|warn|error}]

umet
{ -help | -version }
```

Figure 21.6 Universal Message Translator for HP NonStop - Command Syntax

CHAPTER 22

Universal Message Translator Configuration Options

22.1 Overview

This chapter provides detailed information on the configuration options available for use with Universal Message Translator.

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

Information on how these options are used is documented in Chapter 21 Universal Message Translator.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

22.2 Configuration Options List

Table 22.1 identifies the Universal Message Translator configuration options.

Option Name	Description	Page
HELP	Writes a description of the command options and their format	367
IGNORE_CASE	Specification that matching of message masks to the input file is not case sensitive	368
MESSAGE_FILE	Input message file name	369
MESSAGE_LEVEL	Level of messages that will be displayed	370
TRANSLATION_TABLE	Translation table file name	371
VERSION	Writes the program version and copyright information	371

Table 22.1 Universal Message Translator Configuration Options

22.3 HELP

Description

The HELP option writes a description of the command options and their format.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-h		√	√	√	√
Command Line, Long Form	-help		√	√	√	√
STRUME Parameter	n/a					

Values

(There are no values for this option.)

22.4 IGNORE_CASE

Description

The IGNORE_CASE option specifies that the matching of message masks to the input file is not case sensitive.

(For example, if IGNORE_CASE is used, the word **`Error`** `matches` **ERROR**).

If this option is not used, the matching of message masks is case sensitive.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-i	√	√	√	√	√
Command Line, Long Form	-ignorecase	√	√	√	√	√
STRUME Parameter	IGNORECASE(*option*)	√				

Values

HP NonStop, UNIX, Windows, and z/OS

There are no values for this option.

IBM i

Valid values for *option* are:

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

[Default is no.]

22.5 MESSAGE_FILE

Description

The MESSAGE_FILE option specifies the name of the input message file.

Note: If this option is not used, Universal Message Translator reads its input from standard input.

z/OS

If this option is not used, Universal Message Translator reads the input file from SYSIN ddname.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-f *messages*	√	√	√	√	√
Command Line, Long Form	-file *messages*	√	√	√	√	√
STRUME Parameter	MSGFILE(*messages* [*library*]) [MSGMBR(*member*)]	√				

Values

messages is the name of the input message file.

z/OS

messages is the ddname to which the input message file is allocated.

IBM i

Valid values for *messages* are:

- ***stdin**
  Standard input is used to specify the messages. Standard input is allocated to the workstation for interactive jobs and to file **QINLINE** for batch jobs. If executed as an interactive job, an ILE session terminal is displayed from which input can be entered at the terminal.

  This value is not valid for the command line form. Simply do not specify the command line option to read from standard input.
- *filename*
  Name of a file. The file name can be qualified by a library name. If not, the library list ***LIBL** is searched for the first occurrence of the file name.

  A member name can be used for further qualification by specifying the **MSGMBR** parameter.

22.6 MESSAGE_LEVEL

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l *level*	√	√	√	√	√
Command Line, Long Form	-level *level*	√	√	√	√	√
STRUME Parameter	MSGLEVEL (**level*)	√		√	√	√

Values

level indicates either of the following level of messages:

- **verbose**
  Writes messages that provide information on the message matching process. The messages are helpful for verifying or debugging a translation table.
  (Information, warning, and error messages also are printed.)
- **info**
  Writes informational, warning, and error messages.
- **warn**
  Writes warning and error messages.
- **error**
  Writes error messages only.

IBM i and z/OS

[Default is info.]

HP NonStop, UNIX, and Windows

[Default is warn.]

22.7 TRANSLATION_TABLE

Description

The TRANSLATION_TABLE option specifies the name of the translation table file.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-t *table*	√	√	√	√	√
Command Line, Long Form	-table *table*	√	√	√	√	√
STRUME Parameter	TBL(*table* [*library*]) [TBLMBR(*member*)]	√				

Values

table is the name of the translation table file.

z/OS

table is the ddname to which the translation table is allocated.

IBM i

Valid values for *table* are:

- **umetbl**
  Default file name for the STRUME parameter.
  The command line form does not default. An error is generated if no value is specified.
- *filename*
  Translation table is read from a file. The file name can be qualified by a library name. Otherwise, the library list ***LIBL** is searched for the first occurrence of the file name.
  A member name can be used for further qualification by specifying the **TBLMBR** parameter.

22.8 VERSION

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-v		√	√	√	√
Command Line, Long Form	-version		√	√	√	√
STRUME Parameter	n/a					

Values

(There are no values for this option.)

CHAPTER 25

Universal Query

25.1 Overview

The Universal Query (UQUERY) utility queries any Universal Broker for Broker-related and active component-related information.

25.1.1 Usage

UQUERY returns information for a Universal Broker that is installed on the host, as specified by configuration options on the command line or in a configuration file. Information regarding the components managed by a particular Broker also can be requested.

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

25.2 Universal Query for z/OS

This section describes Universal Query (UQUERY) for the z/OS operating system.

25.2.1 JCL Procedure

Figure 25.1, below, illustrates the Universal Query for z/OS JCL procedure (**UQRYPRC**, located in the **SUNVSAMP** library) that is provided to simplify the execution JCL and future maintenance.

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

Figure 25.1 Universal Query for z/OS – JCL Procedure

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

(See the Stonebranch Solutions 4.2.0 Installation Guide for information regarding the customization of Language Environment parameters.)

25.2.2 DD Statements used in JCL Procedure

Table 25.1, below, describes the DD statements used in the Universal Query for z/OS JCL procedure illustrated in Figure 25.1.

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

Table 25.1 Universal Query for z/OS – DD Statements in JCL Procedure

25.2.3 JCL

Figure 25.2, below, illustrates the Universal Query for z/OS JCL using the **UQRYPRC** procedure illustrated in Figure 25.1.

```
//jobname  JOB CLASS=A,MSGCLASS=X
//STEP1    EXEC UQRYPRC
//SYSIN    DD  *
 -i dallas
/*
```

Figure 25.2 Universal Query for z/OS – JCL

Job step STEP1 executes the procedure **UQRYPRC**.

The command options are specified on the **SYSIN** DD.

25.2.4 Configuration Options

UQUERY for z/OS operations are controlled by the configuration options, which are specified either on the command lie (via the PARM keyword of the EXEC statement or in the SYSIN ddname) or the configuration file.

Table 25.2, below, identifies the UQUERY configuration options for z/OS.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
CODE_PAGE	Character code page used to translate text data received and transmitted over the network.
COMMAND_ID	Requests that Universal Query return information for all records that match the specified command ID.
COMPONENT_ID	Requests that Universal Query return information only for the specified component ID.
HELP	Writes a description of the configuration options and their format.
MANAGERS	Specification for whether or not Universal Query requests manager component information from the queried Broker.
MESSAGE_LANGUAGE	Universal Message Catalog (UMC) file used to write messages.
MESSAGE_LEVEL	Level of messages to write.
OUTBOUND_IP	Sets the host or IP address that UQUERY binds to when initiating outgoing connections.
PING	Information Universal Query requests from Universal Broker.
REMOTE_HOST	IP address of the remote computer.
REMOTE_PORT	TCP port number on the remote computer on which Universal Broker is accepting connections.
REPORT	Format in which Universal Broker information is written.
SYSTEM_ID	Local Universal Broker with which the Universal Query must register.
VERSION	Writes the program version and copyright information.

Table 25.2 Universal Query for z/OS - Configuration Options

25.2.5 Command Line Syntax

Figure 25.3, below, illustrates the command line syntax — using the command line, long form of the configuration options — of UQUERY for z/OS.

```
uquery
-host hostaddress
[-port port]
[-system_id ID]
[-ping {yes|no}]
[-report {normal|fixed}]
[-codepage codepage]
[-level {trace|audit|info|warn|error}]
[-lang language]
[-managers {yes|no}]
[-outboundip host]
[-cmdid ID]
[-component ID]

uquery
{ -help | -version }
```

Figure 25.3 Universal Query for z/OS - Command Line Syntax

25.3 Universal Query for UNIX and Windows

This section describes Universal Query (UQUERY) for the UNIX and Windows operating systems.

25.3.1 Configuration Options

Table 25.3, below, identifies the UQUERY configuration options for UNIX and Windows.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
BIF_DIRECTORY *	Broker Interface File (BIF) directory where the Universal Broker interface file is located.
CODE_PAGE	Character code page used to translate text data received and transmitted over the network.
COMMAND_ID	Requests that Universal Query return information for all records that match the specified command ID.
COMPONENT_ID	Requests that Universal Query return information only for the specified component ID.
HELP	Writes a description of the command options and their format.
MANAGERS	Specification for whether or not Universal Query requests manager component information from the queried Broker.
MESSAGE_LANGUAGE	Universal Message Catalog (UMC) file used to write messages.
MESSAGE_LEVEL	Level of messages to write.
NLS_ DIRECTORY	Directory where the Universal Query message catalog and code page tables are located.
OUTBOUND_IP	Sets the host or IP address that UQUERY binds to when initiating outgoing connections.
PING	Information Universal Query requests from Universal Broker.
PLF_DIRECTORY *	Program Lock File (PLF) directory where the program lock files are located.
REMOTE_HOST	IP address of the remote computer.
REMOTE_PORT	TCP port number on the remote computer on which Universal Broker is accepting connections.
REPORT	Format in which Universal Broker information is written.
VERSION	Writes the program version and copyright information.
* Valid for UNIX only.	

Table 25.3 Universal Query for UNIX and Windows - Configuration Options

25.3.2 Command Line Syntax

Figure 25.4, below, illustrates the command line syntax — using the command line, long form of the configuration options — of UQUERY for Windows and UNIX.

```
uquery
-host hostaddress
[-port port]
[-ping {yes|no}]
[-report {normal|fixed}]
[-bif_directory directory]    (NOTE: This option is valid only for UNIX.)
[-plf_directory directory]    (NOTE: This option is valid only for UNIX.)
[-codepage codepage]
[-level {trace|audit|info|warn|error}]
[-lang language]
[-managers {yes|no}]
[-outboundip host]
[-cmdid ID]
[-component ID]

uquery
{ -help | -version }
```

Figure 25.4 Universal Query for UNIX and Windows - Command Line Syntax

Windows

The Universal Configuration Manager also can be used to obtain the same information reported by the command line version of Universal Query.

25.4 Universal Query for IBM i

This section describes Universal Query (UQUERY) for the IBM i operating system.

25.4.1 Stonebranch Solutions for IBM i Commands

The names of the Stonebranch Solutions for IBM i commands that are installed in the IBM i **QSYS** library are tagged with the Stonebranch Solutions for IBM i **v**ersion / **r**elease / **m**odification number, **420**. The names of the commands installed in the Stonebranch Solutions for IBM i product library, **UNVPRD420**, are untagged.

To maintain consistency across releases, you may prefer to use the untagged names in your production environment. The **UCHGRLS** (Change Release Tag) program lets you change the tagged command names in **QSYS** to the untagged command names in **UNVPRD420**.

(See the Stonebranch Solutions 4.2.0 Installation Guide for detailed information on **UCHGRLS**.)

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

25.4.2 Configuration Options

Table 25.5, below, identifies the UQUERY for IBM i configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
CODE_PAGE	Character code page used to translate text data received and transmitted over the network.
COMMAND_ID	Requests that Universal Query return information for all records that match the specified command ID.
COMPONENT_ID	Requests that Universal Query return information only for the specified component ID.
HELP	Writes a description of the configuration options and their format.
MANAGERS	Specification for whether or not Universal Query requests manager component information from the queried Broker.
MESSAGE_LANGUAGE	Universal Message Catalog (UMC) file used to write messages.
MESSAGE_LEVEL	Level of messages to write.
PING	Information Universal Query requests from Universal Broker.
PLF_DIRECTORY	Program Lock File (PLF) directory where the program lock files are located.
REMOTE_HOST	IP address of the remote computer.
REMOTE_PORT	TCP port number on the remote computer on which Universal Broker is accepting connections.
REPORT	Format in which Universal Broker information is written.
VERSION	Writes the program version and copyright information.

Table 25.4 Universal Query for IBM i - Configuration Options

25.4.3 Command Line Syntax

Figure 25.5, below, illustrates the command line syntax — using the STRUQR parameter form of command line options — of UQUERY for IBM i.

```
STRUQR
HOST(hostaddress)
[PORT(port)]
[PING(*{yes|no})]
[REPORT(*{normal|fixed})]
[CODEPAGE(codepage)]
[MSGLANG(language)]
[MSGLEVEL(*{trace|audit|info|warn|error}]
          (NOTE: trace turns on the trace function.)
[OUTBOUNDIP(host|blank_line)]
[CMDID(ID)]
[COMPID(ID)]
[MANAGERS(*{yes|no})]
[PLFDIR(ifs_directory)]

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

Figure 25.5 Universal Query for IBM i - Command Line Syntax

25.5 Universal Query for HP NonStop

This section describes Universal Query (UQUERY) for the HP NonStop operating system.

25.5.1 Configuration Options

Table 25.5, below, identifies the UQUERY for HP NonStop configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
CODE_PAGE	Character code page used to translate text data received and transmitted over the network.
HELP	Writes a description of the configuration options and their format.
MESSAGE_LANGUAGE	Universal Message Catalog (UMC) file used to write messages.
MESSAGE_LEVEL	Level of messages to write.
PING	Information Universal Query requests from Universal Broker.
REMOTE_HOST	IP address of the remote computer.
REMOTE_PORT	TCP port number on the remote computer on which Universal Broker is accepting connections.
REPORT	Format in which Universal Broker information is written.
VERSION	Writes the program version and copyright information.

Table 25.5 Universal Query for HP NonStop - Configuration Options

25.5.2 Command Line Syntax

Figure 25.6, below, illustrates the command line syntax — using the command line, long form of the configuration options — of UQUERY for HP NonStop.

```
uquery
-host hostaddress
[-port port]
[-ping {yes|no}]
[-report {normal|fixed}]
[-codepage codepage]
[-level {trace|audit|info|warn|error}]
[-lang language]

uquery
{ -help | -version }
```

Figure 25.6 Universal Query for HP NonStop - Command Line Syntax

CHAPTER 26

Universal Query Configuration Options

26.1 Overview

This chapter provides detailed information on the configuration options available for use with Universal Query.

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

Information on how these options are used is documented in Chapter 25 Universal Query.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

26.2 Configuration Options List

Table 26.1 identifies the Universal Query configuration options.

Option Name	Description	Page
BIF_DIRECTORY	Broker Interface Directory that specifies the location of the Universal Broker interface file.	387
COMMAND_ID	Requests that Universal Query return information for all records that match the specified command ID.	388
COMPONENT_ID	Requests that Universal Query return information only for the specified component ID.	388
CODE_PAGE	Character code page used to translate text data received and transmitted over the network.	388
HELP	Writes a description of the command options and their format.	391
MANAGERS	Specification for whether or not Universal Query requests manager component information from the queried Broker.	392
MESSAGE_LANGUAGE	Universal Message Catalog (UMC) file used to write messages.	393
MESSAGE_LEVEL	Level of messages to write.	394
NLS_DIRECTORY	Directory where the Universal Query message catalog and code page tables are located.	396
OUTBOUND_IP	Sets the host or IP address that UQUERY binds to when initiating outgoing connections.	397
PING	Information Universal Query requests from Universal Broker.	398
PLF_DIRECTORY	Program Lock File directory that specifies the location of the Universal Query program lock file.	399
REMOTE_HOST	IP address of the remote computer.	400
REMOTE_PORT	TCP port number on the remote computer on which Universal Broker is accepting connections.	401
REPORT	Format in which Universal Broker information is printed.	402
SYSTEM_ID	Local Universal Broker with which the Universal Query must register.	403
VERSION	Writes the program version and copyright information.	402

Table 26.1 Universal Query Configuration Options

26.3 BIF_DIRECTORY

Description

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

Usage

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

Values

directory is the name of the BIF directory.

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

26.4 CODE_PAGE

Description

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

Usage

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

Values

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

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

See Section 36.3 Character Code Pages for a complete list of character code pages provided by Stonebranch Inc. for use with Stonebranch Solutions.

See Section 36.4 UTT Files for information on UTT files.

Default

The default code page is different for different operating systems:

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

26.5 COMMAND_ID

Description

The COMMAND_ID option specifies a command ID that should be used by Universal Query when searching for component records.

When COMMAND_ID is used, Universal Query will return Broker-specific information and component information for all records that match the specified command ID.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-C *ID*			√	√	√
Command Line, Long Form	-cmdid *ID*			√	√	√
Environment Variable	UQRYCMDID=*ID*	√		√	√	√
Configuration File Keyword	n/a					
STRUQR Parameter	CMDID(*ID*)	√				

Values

ID is the command ID used by Universal Query.

[There is no default.]

26.6 COMPONENT_ID

Description

The COMPONENT_ID option specifies a component ID that should be used by Universal Query when searching for component records.

When COMPONENT_ID is used, Universal Query will return Broker-specific information and component information only for the specified component ID.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-c *ID*			√	√	√
Command Line, Long Form	-component *ID*			√	√	√
Environment Variable	UQRYCOMPONENT=*ID*	√		√	√	√
Configuration File Keyword	n/a					
STRUQR Parameter	COMPID(*ID*)	√				

Values

ID is the component ID used by Universal Query.

[There is no default.]

26.7 HELP

Description

The HELP option writes a description of the command options and their format.

Usage

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

Values

(There are no values for this option.)

26.8 MANAGERS

Description

The MANAGERS option specifies whether or not Universal Query requests manager component information from the Broker being queried.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-managers *option*			√	√	√
Environment Variable	UQRYMANAGERS=*option*	√		√	√	√
Configuration File Keyword	managers *option*	√		√	√	√
STRUQR Parameter	MANAGERS(**option*)	√				

Values

option is the specification for whether or not Universal Query requests manager component information.

Valid values for option are:

- **yes**
  Request manager component information.
- **no**
  Do not request manager component information.

[Default is yes.]

26.9 MESSAGE_LANGUAGE

Description

The MESSAGE_LANGUAGE option specifies the Universal Message Catalog (UMC) file that will be used to write messages.

Each UMC file contains messages for a specific language.

Usage

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

Values

language is the UMC file that will be used to write messages.

The first three characters of the language name are used as a three-character suffix in the UMC file base name (for example, **`uelmceng.umc`**). All UMC files have a **`.umc`** extension.

z/OS

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

IBM i

The first three characters of the language name are used as a three-character suffix in the UMC member base name **`UCMMC`**. UMC files are located in the source physical file **`UNVPRD420/UNVNLS`**.

[Default is `uelmceng.umc`.]

26.10 MESSAGE_LEVEL

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l *level*	√	√	√	√	√
Command Line, Long Form	-level *level*	√	√	√	√	√
Environment Variable	UQRYLEVEL=*level*	√		√	√	
Configuration File Keyword	message_level *level*	√		√	√	√
STRUQR Parameter	MSGLEVEL(**level*)	√				

Values

level indicates either of the following level of messages:

- **trace**
  Writes trace messages used for diagnostic purposes.
  Note: Use **trace** only as directed by Stonebranch, Inc. Customer Support.
- **audit**
  Writes audit, informational, warning, and error messages.
- **info**
  Writes informational, warning, and error messages.
- **warn**
  Writes warning and error messages.
- **error**
  Writes error messages only.

IBM i and z/OS

[Default is info.]

HP NonStop, UNIX, and Windows

[Default is warn.]

Trace Files

UNIX, Windows, and z/OS

Trace file name is **uquery.trc**. It is created in the working directory of Universal Query.

HP NonStop

Trace file name is **UQRYTRC**. It is created in the working subvolume of Universal Query.

IBM i

Trace file name is ***CURLIB/UNVTRCUQR(UQRxxxxxx)**, where **xxxxxx** is the job number of the job invoking Universal Command.

26.11 NLS_ DIRECTORY

Description

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

Usage

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

Values

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

Full path names are recommended.

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

Defaults

UNIX

[Default is `/opt/universal/nls`.]

Windows

[Default is `..\nls`.]

26.12 OUTBOUND_IP

Description

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

By default, no host or IP address is specified for this option.

Usage

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

Values

host is the host or IP address that Universal Query binds to when initiating outgoing connections.

IBM i

Valid values for *host* are:

- *host*
  Host or IP address that Universal Query binds to when initiating outgoing connections.
- (blank line)
  No value is used.

26.13 PING

Description

The PING option specifies the type of information that Universal Query requests from Universal Broker.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-P *option*	√	√	√	√	√
Command Line, Long Form	-ping *option*	√	√	√	√	√
Environment Variable	UQRYPING=*option*	√	√	√	√	
Configuration File Keyword	ping *option*	√	√	√	√	√
STRUQR Parameter	PING(**option*)	√				

Values

option is the specification for the type of information that Universal Query requests.

Valid values for *option* are:

- **yes**
  Universal Broker information only is returned.
- **no**
  Universal Broker information and Universal Broker active component information is returned.

[Default is no.]

26.14 PLF_DIRECTORY

Description

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

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

IBM i

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-plf_directory *directory*			√		
Environment Variable	UQRYPLFDIRECTORY=*directory*	√		√		
Configuration File Keyword	n/a					
STRUQR Parameter	PLFDIR(*directory*)	√				

Values

directory is the name of the PLF directory.

A full path name must be specified.

Defaults

UNIX

[Default is `/var/opt/universal/tmp`.]

IBM i

[Default is `/tmp`.]

26.15 REMOTE_HOST

Description

The REMOTE_HOST option specifies the IP address of the remote computer.

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-i *hostaddress*	√	√	√	√	√
Command Line, Long Form	-host *hostaddress*	√	√	√	√	√
Environment Variable	UQRYHOST=*hostaddress*	√	√	√	√	
Configuration File Keyword	host *hostaddress*	√	√	√	√	√
STRUQR Parameter	HOST(*hostaddress*)	√				

Values

hostaddress is the IP address of the remote computer.

The format of *hostaddress* can be either:

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

26.16 REMOTE_PORT

Description

The REMOTE_PORT option specifies the TCP port on the remote computer on which Universal Broker is accepting connections.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-p *port*	√	√	√	√	√
Command Line, Long Form	-port *port*	√	√	√	√	√
Environment Variable	UQRYPORT=*port*	√	√	√	√	
Configuration File Keyword	port *port*	√	√	√	√	√
STRUQR Parameter	PORT(*port*)	√				

Values

port is the TCP port on which Universal Broker is accepting connections.

The format of *port* can be either:

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

[Default is *7887*.]

26.17 REPORT

Description

The REPORT option specifies the format in which the Universal Broker information is written.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-R *format*	√	√	√	√	√
Command Line, Long Form	-report *format*	√	√	√	√	√
Environment Variable	UQRYREPORT=*format*	√	√	√	√	
Configuration File Keyword	report *format*	√	√	√	√	√
STRUQR Parameter	REPORT(**format*)	√				

Values

format is the format in which the Universal Broker information is written.

Valid values for *report* are:

- **normal**
  Universal Broker information is written one field per line.
- **fixed**
  Universal Broker information is written in a table format with limited number of columns.

[Default is normal.]

26.18 SYSTEM_ID

Description

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

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

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-system_id *ID*					√
Environment Variable	UQRYSYSTEMID=*ID*					√
Configuration File Keyword	n/a					
STRUQR Parameter	n/a					

Values

ID is the system identifier of the local Universal Broker.

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

26.19 VERSION

Description

The VERSION option writes the program version and copyright information.

Usage

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

Values

HP NonStop, UNIX, Windows, and z/OS

There are no values for this option.

IBM i

Valid values for *option* are:

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

[Default is no.]

CHAPTER 27

Universal Return Code

27.1 Overview

The Universal Return Code utility is a Windows utility that performs the function of ending a process with a return code that is equal to its command line argument.

The return code of a Windows batch script is the return code of the last command executed. Universal Return Code can be used as the last command to set the return code of the batch script.

27.2 Usage

The Universal Return Code program is **`urc.exe`**.

It exits with its integer command line argument as its return code.

27.2.1 Command Line Syntax

Figure 27.1, below, illustrates the syntax - using the command line, long form of the configuration options - of Universal Return Code.

```
urc
return_code

urc
{ -help | -version }
```

Figure 27.1 Universal Return Code – Command Line Syntax

27.2.2 Configuration Options

The command line arguments to Universal Return Code are:

Argument	Description
return_code	Integer-value return code of Universal Return Code.
-? | -h | -help	Write command instructions.
-v | -version	Write version information.

CHAPTER 28

Universal Spool List

28.1 Overview

The Universal Spool List (**USLIST**) utility provides the ability to list Universal Spool database records. Universal Spool List must be executed on the system on which the database is located.

The functions that Universal Spool List provide are required for possible database clean-up or problem resolution by Stonebranch, Inc. Customer Support.

28.1.1 Usage

The Universal Spool List utility reads requested records from a specified database. The selected records are written to standard output.

Universal Spool List performs operations specified by the command options.

28.2 Universal Spool List for z/OS

This section describes Universal Spool List for the z/OS operating system.

28.2.1 Databases

Universal Spool databases are implemented as HFS data sets. The HFS data sets **UNVDB** and **UNVSPOOL** contain an HFS file system that contains the Universal Spool database files.

Table 28.1, below, identifies the database files and the HFS data sets in which they reside.

Database Name	Data Set	File Name
Universal Broker Component Database	UNVDB	bcomponent.db
Universal Server Component Database	UNVDB	scomponent.db
Universal Server Spool Databases	UNVSPOOL	spool.stdin.*COMPID*.db spool.stdout.*COMPID*.db spool.stderr.*COMPID*.db
Note: In the Universal Server Spool Databases file names, *COMPID* is the component ID assigned to the Server instance.		

Table 28.1 Universal Spool – Databases

28.2.2 JCL Procedure

Figure 28.1, below, illustrates the Universal Spool List for z/OS JCL procedure (**USLLSPRC**, located in the **SUNVSAMP** library) that is provided to simplify the execution JCL and future maintenance.

```
//USLLSPRC PROC UPARM=,               -- USLIST options
//             UNVPRE=#SHLQ.UNV,
//             CFGPRE=#PHLQ.UNV,
//             DBPRE=#PHLQ.UNV
//*
//PS1      EXEC PGM=USLIST,PARM='ENVAR(TZ=EST5EDT)/&UPARM'
//STEPLIB  DD  DISP=SHR,DSN=&UNVPRE..SUNVLOAD
//*
//UNVNLS   DD  DISP=SHR,DSN=&UNVPRE..SUNVNLS
//UNVCONF  DD  DISP=SHR,DSN=&CFGPRE..UNVCONF
//UNVDB    DD  DISP=SHR,DSN=&DBPRE..UNVDB
//UNVSPOOL DD  DISP=SHR,DSN=&DBPRE..UNVSPOOL
//UNVTRACE DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
```

If zFS data sets are being used instead of the default HFS data sets, the UNVDB and UNVSPOOL ddnames cannot be used to allocate the zFS data sets. The ddnames must be removed along with the DBPRE procedure parameter.

Figure 28.1 Universal Spool List for z/OS – JCL Procedure

28.2.3 DD Statements used in JCL Procedure

Table 28.2, below, describes the DD statements used in the Universal Spool List for z/OS JCL procedure illustrated in Figure 28.2.

ddname	Description
STEPLIB	Load library in which program USLIST program is located.
UNVNLS	Universal National Language Support library.
UNVCONF	Stonebranch Solutions configuration library.
UNVDB	Universal Broker Database HFS data set.
UNVSPOOL	Universal Spool Database HFS data set.
UNVTRACE	Application trace ddname.
SYSOUT	USLIST standard error ddname.
SYSPRINT	USLIST standard output ddname.

Table 28.2 Universal Spool List for z/OS – DD Statements in JCL Procedure

28.2.4 JCL

Figure 28.2, below, illustrates the Universal Spool List for z/OS JCL.

```
//STEP1    EXEC PGM=USLIST
//STEPLIB  DD  DISP=SHR,DSN=UNV.SUNVLOAD
//UNVNLS   DD  DISP=SHR,DSN=UNV.SUNVNLS
//UNVCONF  DD  DISP=SHR,DSN=&CFGPRE..UNVCONF
//UNVDB    DD  DISP=SHR,DSN=UNV.UNVDB
//UNVSPOOL DD  DISP=SHR,DSN=UNV.UNVSPOOL
//UNVTRACE DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 command options
/*
```

If zFS data sets are being used instead of the default HFS data sets, the UNVDB and UNVSPOOL ddnames cannot be used to allocate the zFS data sets. The ddnames must be removed along with the DBPRE procedure parameter.

Figure 28.2 Universal Spool List for z/OS – JCL

28.2.5 Configuration Options

Table 28.3, below, identifies the Universal Spool List for z/OS configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
COMPONENT	Component identifier for which records will be selected to write.
HELP	Writes a description of the command options and their format.
LIST	Type of database from which to select record to write.
MESSAGE_LEVEL	Level of messages that will be written.
MOUNT_POINT	HFS directory in which the HFS databases allocated to ddnames UNVDB and UNVSPOOL are mounted.
VERSION	Writes the program version and copyright information.

Table 28.3 Universal Spool List for z/OS - Configuration Options

28.2.6 Command Line Syntax

Figure 28.4, below, illustrates the command line syntax — using the command line, long form of the configuration options — of Universal Spool List for z/OS.

```
uslist
[-list {ubroker|ucmd|urmtcfg|stdin|stderr|stdout}]
[-component cid]
[-mount_point directory]
[-level {audit|info|warn|error}]

uslist
{ -help | -version }
```

Figure 28.3 Universal Spool List for z/OS - Command Line Syntax

28.2.7 zFS Support

Universal Spool List (**USLIST**) and Universal Spool Remove (**USLRM**) obtain the HFS data set names from **UNVDB** and **UNVSPOOL** ddnames. zFS data sets cannot be provided via ddnames.

To obtain the zFS or HFS data set names, **USLIST** and **USLRM** allocate and parse the Universal Broker's configuration member, **UBRCFG00**, for the UNIX_DB_DATA_SET and UNIX_SPOOL_DATA_SET options. If these options are not found, **USLIST** and **USLRM** assume that HFS data sets are being used and refer the **UNVDB** and **UNVSPOOL** ddnames for the HFS data set names.

Note: Previous versions of **USLIST** and **USLRM** did not allocate the Universal Broker's configuration file.

USLIST and **USLRM** have always mounted the HFS data sets if they were not already mounted. This is the case with zFS data sets as well.

The Stonebranch Solutions configuration PDSE is allocated to **UNVCONF** in the **USLLSPRC** and **USLRMPRC** JCL procedures in the **SUNVSAMP** library. The only other changes to **USLLSPRC** and **USLRMPRC** is the addition of a comment that the **UNVDB** and **UNVSPOOL** ddnames must be commented out to use zFS data sets.

USLIST and **USLRM** now write messages UNV2264I and UNV2265I, which provide information on the Broker's database and spool, respectively. The messages also provide the file system type, data set name, and mount point.

28.3 Universal Spool List for Windows and UNIX

This section describes Universal Spool List for the Windows and UNIX operating systems.

The Universal Spool List utility can be used to read the databases listed in Table 28.4.

Database Name	Database File Name
Universal Broker Component Database	bcomponent.db
Universal Server Component Database	scomponent.db
Universal Server Spool Databases	spool.stdin.*COMPID*.db spool.stdout.*COMPID*.db spool.stderr.*COMPID*.db
Universal Event Monitor Event Definition Database	ueme.db
Universal Event Monitor Event Handler Database	uemh.db
Universal Event Monitor Spool Database	uems.db

Table 28.4 Universal Spool List for Windows and UNIX - Databases

The *COMPID* in the Universal Server Spool Databases file name in this table is the component ID assigned to the Server instance.

UNIX

By default, the database files are located in the **/var/opt/universal/spool** directory.

The program file is located in the Universal Spool installation directory **bin** directory, which defaults to **/opt/universal/uspool-4.2.0/bin**.

Windows

By default, the database files are stored in the **C:\Program Files\Universal\spool** directory.

The Universal Spool List program file, **uslist.exe**, is located in the **bin** subdirectory of the Universal Spool installation directory, which defaults to **C:\Program Files\Universal\uspool**.

28.3.1 Configuration Options

Table 28.5, below, identifies the Universal Spool List for Windows and UNIX configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
BROKER_SPOOL_DIR	Directory location in which the Universal Broker Component database is located.
COMPONENT	Component identifier for which records will be selected to write.
HELP	Writes a description of the command options and their format.
ID	Lists the contents of a specific record from the Universal Event Monitor event definition, event handler, or spool databases.
LIST	Type of database from which to select record to written.
MESSAGE_LEVEL	Level of messages that will be written.
UCMD_SPOOL_DIR	Directory location in which the Universal Server Component database is located.
VERSION	Writes the program version and copyright information.

Table 28.5 Universal Spool List for Windows and UNIX - Configuration Options

28.3.2 Command Line Syntax

Figure 28.4, below, illustrates the command line syntax — using the command line, long form of the configuration options — of Universal Spool List for Windows and UNIX.

```
uslist
[-list {ubroker|ucmd|ueme|uemh|uems|urmtcfg|stdin|stderr|stdout}]
[-component cid]
[-id id]
[-brokerspooldir directory]
[-ucmdspooldir directory]
[-level {audit|info|warn|error}]

uslist
{ -help | -version }
```

Figure 28.4 Universal Spool List for Windows and UNIX - Command Line Syntax

28.4 Universal Spool List for IBM i

This section describes Universal Spool List for IBM i operating system.

The Universal Spool List utility can be used to read the databases listed in Table 28.6, below.

Database Name	Database File Name
Universal Broker Component Database	**UBR_CMP_DB**
Universal Server Component Database	**SRV_CMP_DB**
Universal Server Spool Databases *	**SIcompid** (STDIN) **SOcompid** (STDIN) **SEcompid** (STERR)
I-Management Console Remote Configuration Database	UNVCFG_DB
The **compid** in the Universal Server Spool Databases file names in this table is the component ID (in hexadecimal) assigned to the Server instance.	

Table 28.6 Universal Spool List for IBM i - Databases

The spool files are located in library **UNVSPL420**.

28.4.1 Configuration Options

Table 28.7, below, identifies the Universal Spool List for IBM i configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
COMPONENT	Component identifier for which records will be selected to write.
ID	Lists the contents of a specific record from the Universal Event Monitor event definition, event handler, or spool databases.
LIST	Type of database from which to select record to written.
MESSAGE_LEVEL	Level of messages that will be written.
VERSION	Writes the program version and copyright information.

Table 28.7 Universal Spool List for IBM i – Configuration Options

28.4.2 Command Line Syntax

Figure 28.5, below, illustrates the command line syntax — using the ULSTSE parameter form of command line options — of Universal Spool List for IBM i.

```
ULSTSE
[LIST(*{ubroker|ucmd|urmtcfg|stdin|stderr|stdout})]
[COMPONENT(cid)]
[ID(id)]
[LEVEL(*{audit|info|warn|error})]

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

Figure 28.5 Universal Spool List for IBM i – Command Line Syntax

28.5 Universal Spool List Output

28.5.1 Universal Broker Component

Table 28.8, below, identifies the information written for a requested Universal Broker component.

Field Name	Description
ID	Component identifier.
Name	Component definition name. Either **ucmd** or **uctl**.
Desc	Component description field from the component definition.
Version	Component version and build level.
State	Component state.
Cmd ID	Command identifier provided by the manager.
Comm State	Component communication state.
Comp State Time	Date and time the component entered the communication state.
Restartable	Specification for whether or not the component is restartable (manager fault tolerant).
Srv PID	Component's process identifier.
Srv Start Time	Components start date and time.
Srv End Time	Component's end date and time.
Srv Exit Code	Component's exit code if its status is not RUNNING.
Srv Exit Status	Component's execution status.
Mgr UID	Manager's user identifier.
Mgr Work ID	Manager's work identifier.
Mgr Host Name	Manager's TCP/IP host name on which it's executing.
Mgr Port	Manager's TCP/IP port number from which it connected.

Table 28.8 Universal Spool List Output - Universal Broker Component

28.5.2 Universal Broker Component List

Table 28.9, below, identifies the columns of data in a Universal Broker component list.

Column Name	Description
ID	Component Identifier
NAME	Component name. Either **ucmd** or **uctl**.
CST	Component's communication state. Communication state values are • COM Component is completed. • DIS Communication link between the component and the manager is disconnected. The status of the manager is unknown. The network fault tolerant protocol is being used. • EST Communication link between the component and the manager is established. This is the normal operating mode. • ORH Component is executing, but the manager has terminated. The manager is orphaned. The component was started with manager fault tolerance and is waiting for a manager restart. The user process is still executing. • PEN Component has completed its work and is waiting for a manager to restart to receive the user process spool files and exit status. The component was started with manager fault tolerance. • RCG Component is in the middle of reconnecting the manager. The network fault tolerant protocol is being used. • RSG Component is in the middle of restarting with a manager. The component was started with manager fault tolerance. • RSA Manager restart request has been accepted. The manager and the component will be reestablishing their communication links. • STR Component is starting. The component usually remains in this state for a short period of time unless they are executing with manager fault tolerance and the manager is redirecting a large stdin file.
MGR-WORK-ID	Manager's work identifier. The work ID format depends on the system type on which the manager is executing.
COMMAND-ID	Command identifier specified by the manager.

Table 28.9 Universal Spool List Output - Universal Broker Component List

28.5.3 Universal Command Server Component

Table 28.10 identifies the information in a requested Universal Command Server component.

Field Name	Description
ID	Component identifier.
Cmd Line	User command the manager requested to be executed.
User ID	User identifier with which the user command is executing.
Desc	Component description.
Comm State	Component communication state.
Comp State Time	Date and time the component entered the communication state.
Restartable	Whether the component is restartable (manager fault tolerant).
Spool Retention	Number of days to retain the spool files after the component goes into completed state.
Comp Retention	Number of days to retain the component record after the component goes into completed state.
PID	User command's process identifier.
Start Time	User process's start date and time.
End Time	User process's end date and time.
Exit Code	User process's exit code if it is not RUNNING.
Exit Status	User process's exit status.
Script File	Script file being executed by the Server.

Table 28.10 Universal Spool List - Universal Command Server Component

28.5.4 Universal Command Server Component List

Table 28.11, below, identifies the columns of data in a Universal Command Server component record.

Column Name	Description
ID	Component Identifier.
CST	Component's communication state. (See **CST** in Table 28.9 Universal Spool List Output - Universal Broker Component List.)
USER-ID	Local user account with which the user process is being executed.
COMMAND	Command which the manager requested to be executed.

Table 28.11 Universal Spool List Output - Universal Command Server Component List

28.5.5 Universal Event Monitor Event Definition

Table 28.12 identifies the information that is displayed for a requested Universal Event Monitor event definition.

Field Name	Description
Event ID	Event definition identifier.
Event Type	Type of system event that the event definition is responsible for detecting and monitoring. The following event types are supported: • FILE Instructs UEM to detect the creation of a file and track its progress until it meets its specified completion criteria.
Component Name	Name of the event-driven Universal Event Monitor Server component to which the event definition is assigned. This is the UEM Server component responsible for monitoring the event.
Enabled	Indicates whether the event definition is currently recognized by its assigned UEM Server. An enabled event definition will be monitored as long as the current date and time fall within the activation and inactivation times. A disabled event definition will never become active and will never be monitored, unless it is explicitly enabled.
Active	Indicates whether the event definition is currently being monitored by its assigned UEM Server. An event definition must be enabled before it will be made active. The event will be made inactive once its inactivation time elapses.
Activation Time	Date and time at which the assigned UEM Server component will begin monitoring this event.
Inactivation Time	Date and time at which the assigned UEM Server component will stop monitoring this event.
Tracking Interval	Frequency, in seconds, with which UEM will test for the completion of any system occurrence detected for this event.
Triggered Handler	ID of a record stored in the event handler database that should be executed whenever the processing state for an occurrence of this event is set to TRIGGERED.
Expired Handler	ID of a record stored in the event handler database that should be executed whenever the processing state for this event is set to EXPIRED.
Rejected Handler	ID of a record stored in the event handler database that should be executed whenever the processing state for an occurrence of this event is set to REJECTED.
Handler Options	Parameters that UEM adds to the command line used to execute the event handler process. The event handler process receives these parameters as command line options.
Last Modified On	Date and time the event definition record was last updated.
Last Modified By	Name of the user account that last updated the event definition record.

Table 28.12 Universal Spool List - Universal Event Monitor Event Definition

28.5.6 Event Type-Specific Fields

The following sections describe the event definition fields that vary depending on the value of the Event Type parameter.

FILE Event Definitions

Table 28.13, below, identifies the fields that are displayed for events with an event type of **FILE**.

Field Name	Description
File Specification	File whose creation should be detected and whose progress should be tracked by UEM.
Minimum File Size	Smallest size a file must be before it is considered complete by UEM.
Rename File	Indication of whether or not the file will be renamed by UEM whenever the processing state of the tracked event occurrence is set to TRIGGERED.
Rename Specification	Format that UEM should use when renaming a file whose event occurrence has been set to the TRIGGERED state.

Table 28.13 FILE Event Fields

28.5.7 Universal Event Monitor Event Definition List

Table 28.14, below, identifies the items for which values are defined in a Universal Event Monitor Event Definitions list.

Column Name	Description
EVENT ID	Event Definition Identifier.
TYPE	Type of system event that the event definition is responsible for detecting and monitoring. For a complete list of supported event types, see **Event Type** in Table 28.12 Universal Spool List - Universal Event Monitor Event Definition.
ENABLED	Indication of whether or not the event definition currently is being processed by its assigned UEM Server.
ACTIVE	Indication of whether or not the event definition currently is being monitored by its assigned UEM Server.

Table 28.14 Universal Spool List - Universal Event Monitor Event Definition List

28.5.8 Universal Event Monitor Event Handler

Table 28.15 identifies the information displayed for a requested Universal Event Monitor event handler.

Field Name	Description
Handler ID	Event Handler Identifier.
Handler Type	Process which is executed on behalf of the event handler. The following process types are supported: • CMD Indicates the record contains the name of an application, along with all of its required command line parameters, that is to be executed on behalf of the event handler. • SCRIPT Indicates the record contains a set of one or more system commands that are to be executed as a single script on behalf of the event handler.
Max Acceptable Return Code	Highest value an event handler process may return to still be considered as having executed successfully.
User ID	ID of the user account in whose security context the event handler process will be executed.
Command	If the value of the Handler Type parameter is **CMD**, this field shows the command to execute. This field will not be shown if the value of the Handler Type parameter is **SCRIPT**.
Script Statements	If the value of the Handler Type parameter is **SCRIPT**, this field marks the beginning of the system commands that will be executed as a script. This field will not be shown if the value of the Handler Type parameter is **CMD**.
Script Type	Type of script statements to execute when the value of the Handler Type parameter is **SCRIPT**. This field will not be shown if the value of the Handler Type parameter is **CMD**.
Last Modified On	Date and time the event handler record was last updated.
Last Modified By	Name of the user account that last updated the event handler record.

Table 28.15 Universal Spool List - Universal Event Monitor / Event Handler

28.5.9 Universal Event Monitor Event Handler List

Table 28.16, below, identifies the items for which values are defined in a Universal Event Monitor Event Handlers list.

Column Name	Description
HANDLER ID	Event Handler Identifier.
TYPE	Describes the process which is executed on behalf of the event handler. For a complete list of supported process types, see **Handler Type** in Table 28.15 Universal Spool List - Universal Event Monitor / Event Handler.

Table 28.16 Universal Spool List - Universal Event Monitor Event Handler List

28.5.10 Universal Event Monitor Spool List

Table 28.17 identifies the items for which values are listed in Universal Event Monitor Spool List.

Column Name	Description
SERIAL NO	A sequential number that is assigned to each record as it is added to the database. This number serves to uniquely identify each occurrence of a given event definition.
EVENT ID	The ID of the event definition responsible for the detection and monitoring of the event occurrence recorded by this spool record.
PRC STATE	The processing state of the event occurrence. For a complete list of possible values, see the description of the Processing State parameter, below.
HANDLER ID	The ID of an event handler executed whenever the processing state of an event or an event occurrence enters the TRIGGERED, REJECTED, or EXPIRED state.
EXIT CODE	The value returned by the process executed on behalf of an event handler.
EXIT STATUS	Indicates whether the event handler process ended normally or whether it was terminated unexpectedly.
HANDLER STATUS	Indicates the outcome of event handler processing. For a list of possible values, see **`Handler Status`** in Table 28.18 Universal Spool List - Universal Event Monitor Spool Record.

Table 28.17 Universal Spool List - Universal Event Monitor Spool List

28.5.11 Universal Event Monitor Spool Record

Table 28.18 identifies the information displayed for a requested Universal Event Monitor spool record.

Field Name	Description
Serial No	Sequential number that is assigned to each record as it is added to the database. This number serves to uniquely identify each occurrence of a given event definition.
Event ID	ID of the event definition responsible for the detection and monitoring of the event occurrence recorded by this spool record.
Component Name	Name of the event-driven Universal Event Monitor Server component to which the event definition is assigned. This is the UEM Server component responsible for monitoring and processing the event.
Component Description	Description of the UEM Server component identified by Component Name.
Component Version	Version of the UEM Server component identified by Component Name.
Component ID	Value that uniquely identifies the instance of the UEM Server component that processed the event occurrence.
Event Type	Type of system event that the event definition is responsible for detecting and monitoring.
System Object	System event detected and monitored by the event occurrence.
Processing State	Processing state of the event occurrence. The following values are used: • TRACKING Indicates that an occurrence of a system event described by an event definition was detected, but has not yet met the completion criteria set forth by the event definition and by UEM's application logic. • TRIGGERED Indicates that an occurrence of a system event described by an event definition was detected and has completed. If a triggered event handler was specified in the event definition, that handler's process will be executed and the event handler's ID will be shown in the Handler ID field. • REJECTED Indicates than an occurrence of a system event described by an event definition was detected, but failed to complete before the date and time specified in the event definition's Inactivation Time parameter. If a rejected event handler was specified in the event definition, that handler's process will be executed and the event handler's ID will be shown in the Handler ID field. • EXPIRED Indicates that no occurrence of the system event described by an event definition was detected before the event's Inactivation Time elapsed. If an expired event handler was specified in the event definition, that handler's process will be executed and the event handler's ID will be shown in the Handler ID field. • ERROR Indicates an error occurred while processing the event occurrence.
Handler ID	ID of an event handler executed whenever the processing state of an event or an event occurrence enters the TRIGGERED, REJECTED, or EXPIRED state.

Field Name	Description
User Command	Command executed on behalf of the event handler when the Handler Type is CMD. If the Handler Type is SCRIPT, this field contains no value.
Process ID	ID of the process executed by UEM on behalf of the event handler.
User ID	Name of the user account in whose security context the event handler process was executed.
Start Time	Date and time the event handler process started.
End Time	Date and time the event handler process ended.
Exit Code	Value returned by the event handler process.
Exit Status	Indicates whether the event handler process ended normally or whether it terminated unexpectedly.
Handler Status	Indicates the outcome of event handler processing. The following values are used: • FAILED The event handler process finished abnormally, or ended normally with an exit code greater than the maximum acceptable return code specified in the event handler record. • NO HANDLER No event handler was specified for the event's processing state. • NOT AUTHORIZED An attempt to execute the event handler process failed because the user ID or password specified for the event handler was incorrect. • NOT FOUND The ID of an event handler record specified for a particular processing state was not found in the event handler database. • SHUTDOWN The Universal Event Monitor Server was stopped while the event handler process was running. • SUCCESSFUL The event handler process completed normally and exited with a value that was less than or equal to the maximum acceptable return code specified in the event handler record. • UNRECOVERABLE Information for the event handler process could not be recovered.
Last Modified On	Date and time the spool record was last updated.

Table 28.18 Universal Spool List - Universal Event Monitor Spool Record

CHAPTER 29

Universal Spool List Configuration Options

29.1 Overview

This chapter provides detailed information on the configuration options available for use with Universal Spool List.

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

Information on how these options are used is documented in Chapter 28 Universal Spool List.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

29.2 Configuration Options List

Table 29.1 identifies the Universal Spool List configuration options.

Option Name	Description	Page
BROKER_SPOOL_DIR	Directory location in which the Universal Broker Component database is located.	429
COMPONENT	Component identifier for which records will be selected to write.	430
HELP	Writes a description of command options and their format.	431
ID	Lists the contents of a specific record from the Universal Event Monitor event definition, event handler, or spool databases.	432
LIST	Type of database from which to select records to write.	433
MESSAGE_LEVEL	Level of messages to be written.	435
MOUNT_POINT	HFS directory in which the HFS databases allocated to ddnames UNVDB and UNVSPOOL are mounted.	436
UCMD_SPOOL_DIR	Directory location in which the Universal Server Component database is located.	437
VERSION	Writes the program version and copyright information.	437

Table 29.1 Universal Spool List Configuration Options

29.3 BROKER_SPOOL_DIR

Description

The BROKER_SPOOL_DIR option specifies the directory in which the Universal Broker component database (**bcomponent.db**) is located.

If this option is not used to specify the directory, the directory is read from the Universal Broker configuration file.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-b *directory*			√	√	
Command Line, Long Form	-brokerspooldir *directory*			√	√	
ULSTSE Parameter	n/a					

Values

directory is the directory in which the Universal Broker component database is located.

29.4 COMPONENT

Description

The COMPONENT option specifies the ID of a single component (job) for which records will be selected to be written.

The LIST option identifies the database of the component.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-c *cid*			√	√	√
Command Line, Long Form	-component *cid*			√	√	√
ULSTSE Parameter	COMPONENT(*cid*)	√				

Values

cid is the component ID.

29.5 HELP

Description

The HELP option writes a description of the command options and their format.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-h			√	√	√
Command Line, Long Form	-help			√	√	√
ULSTSE Parameter	n/a					

Values

(There are no values for this option.)

29.6 ID

Description

The ID option specifies the ID of a single record (from the Universal Event Monitor event definition, event handler, or spool databases) to be written.

The LIST option specifies the spool database.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-i *id*			√	√	
Command Line, Long Form	-id *id*			√	√	
ULSTSE Parameter	ID(*id*)	√				

Values

id is the ID of a specific record.

- For an event definition or event handler record, *id* is the event ID or event handler ID, respectively.
- For a spool database record, *id* is the sequential serial number automatically assigned to the record.

29.7 LIST

Description

The LIST option specifies the database from which to select records to write.

- Universal Broker
- Universal Command Server
- Universal Event Monitor
- Spool

The COMPONENT option is used to select individual component (job) records from the database.

The ID option is used to select a single record from a Universal Event Monitor database.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a			√	√	√
Command Line, Long Form	-list *option*			√	√	√
ULSTSE Parameter	LIST(**option*)	√				

Values

option is the database from which to select records.

Valid values for *option* are:

- **ubroker**
  List the contents of the Universal Broker Component database. A summary of all records is written.
- **ucmd**
  List the contents of the Universal Command Server Component database. A summary of all records is written.
- **ueme** (UNIX and Windows only)
  List the contents of the Universal Event Monitor Event Definition database. A summary of all records is written.
- **uemh** (UNIX and Windows only)
  List the contents of the Universal Event Monitor Event Handler database. A summary of all records is written. Use the ID option to list the complete contents of a single record.
- **uems** (UNIX and Windows only)
  List the contents of the Universal Event Monitor Spool database. A summary of all records is written. Use the ID option to list the complete contents of a single record.
- **urmtcfg**
  List the contents of the I-Management Console Remote Configuration database. A summary of all records is written. Use the ID option to list the complete contents of a single record.
- **stdin**
  List the standard input spool file for a specified component.
- **stderr**
  List the standard error spool file for a specified component.
- **stdout**
  List the standard output spool file for a specified component.

[Default is ubroker.]

29.8 MESSAGE_LEVEL

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l *level*			√	√	√
Command Line, Long Form	-level *level*			√	√	√
ULSTSE Parameter	LEVEL(**level*)	√				

Values

level indicates either of the following level of messages:

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

z/OS and IBM i

[Default is info.]

UNIX and Windows

[Default is warn.]

29.9 MOUNT_POINT

Description

The MOUNT_POINT option specifies the HFS directory in which the HFS database allocated to ddnames UNVDB and UNVSPOOL are mounted.

The actual mount points will be subdirectories named after the HFS data set names being mounted.

If the mount points do not exist, they are created by Universal Spool List.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-mount_point *directory*					√
ULSTSE Parameter	n/a					

Values

directory is the HFS directory in which the HFS databases are mounted.

[Default is */tmp*.]

29.10 UCMD_SPOOL_DIR

Description

The UCMD_SPOOL_DIR option specifies the directory in which the Universal Server component database (**`scomponent.db`**) is located.

If this option is not used to specify the directory, the directory of the Universal Broker component database is used (see the BROKER_SPOOL_DIR option).

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-u *directory*			√	√	
Command Line, Long Form	-ucmdspooldir *directory*			√	√	
ULSTSE Parameter	n/a					

Values

directory is the directory of the Universal Server component database.

29.11 VERSION

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-v			√	√	√
Command Line, Long Form	-version			√	√	√
ULSTSE Parameter	VERSION(**option*)	√				

Values

UNIX, Windows, z/OS

(There are no values for this option.)

IBM i

Valid values for *option* are:

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

CHAPTER 30

Universal Spool Remove

30.1 Overview

Universal Spool Remove (**USLRM**) utility provides the ability to remove component records from the Universal Command and Universal Event Monitor (UNIX and Windows only) Spool databases. Universal Spool Remove must be executed on the system upon which the database is located.

By default, spool records are not retained after they no longer are needed. Accordingly, it is not anticipated that the spool databases will become too large. However, on occasion, some records may not be cleaned up, making it necessary to remove them with the Universal Spool Remove utility.

30.1.1 Prerequisite to Running Universal Spool Remove

Before attempting to remove any records using Universal Spool Remove, ensure that the Universal Broker is not running on the local system.

While it is active, the Universal Broker, in its role as a local database administrator, actually “owns” and maintains an open reference to the spool databases. Any changes made to these databases outside of the Broker are not committed to the database while this reference is open. If Universal Spool Remove removes a spool record while the Broker is running, that same record will “reappear” the next time that the Broker commits an update to the database (for example, a new server component is started and recorded in the spool).

All spool records that are deleted as part of the regular component clean-up are permanent because those deletions are done via the Broker.

The functions that Universal Spool List program provide are required for possible database clean-up (see Chapter 29 Universal Spool List Configuration Options).

30.2 Usage

The Universal Spool Remove utility removes all records for a specified component ID from the Universal Spool databases.

Any errors encountered while records are being removed from a database will be reported, but will not result in the program being stopped.

Universal Spool Remove removes records as specified by the command options.

30.3 Universal Spool Remove for z/OS

This section describes Universal Spool Remove for the z/OS operating system.

30.3.1 Databases

Universal Spool databases are implemented as HFS data sets. The HFS data sets UNVDB and UNVSPOOL contain an HFS file system that contains the Universal Spool database files.

Table 30.1, below, identifies the database files and the HFS data sets in which they reside.

Database Name	Data Set	File Name
Universal Broker Component Database	UNVDB	bcomponent.db
Universal Server Component Database	UNVDB	scomponent.db
Universal Server Spool Databases	UNVSPOOL	spool.stdin.*COMPID*.db spool.stdout.*COMPID*.db spool.stderr.*COMPID*.db
Note: In the Universal Server Spool Databases file names, *COMPID* is the component ID assigned to the Server instance.		

Table 30.1 Universal Spool Remove for z/OS - Universal Spool Databases

30.3.2 JCL Procedure

Figure 30.1, below, illustrates the Universal Spool Remove for z/OS JCL procedure (**USLRMPRC**, located in the **SUNVSAMP** library) that is provided to simplify the execution JCL and future maintenance.

```
//USLRMPRC PROC UPARM=,               -- USLRM options
//             UNVPRE=#SHLQ.UNV,
//             CFGPRE=#PHLQ.UNV,
//             DBPRE=#PHLQ.UNV
//*
//PS1      EXEC PGM=USLRM,PARM='ENVAR(TZ=EST5EDT)/&UPARM'
//STEPLIB  DD  DISP=SHR,DSN=&UNVPRE..SUNVLOAD
//*
//UNVNLS   DD  DISP=SHR,DSN=&UNVPRE..SUNVNLS
//UNVCONF  DD  DISP=SHR,DSN=&CFGPRE..UNVCONF
//UNVDB    DD  DISP=SHR,DSN=&DBPRE..UNVDB
//UNVSPOOL DD  DISP=SHR,DSN=&DBPRE..UNVSPOOL
//UNVTRACE DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
```

If zFS data sets are being used instead of the default HFS data sets, the UNVDB and UNVSPOOL ddnames cannot be used to allocate the zFS data sets. The ddnames must be removed along with the DBPRE procedure parameter.

Figure 30.1 Universal Spool Remove for z/OS – JCL Procedure

30.3.3 DD Statements used in JCL Procedure

Table 30.2, below, describes the DD statements used in the Universal Spool Remove for z/OS JCL procedure illustrated in Figure 30.2.

ddname	Description
STEPLIB	Load library in which program USLRM program is located.
UNVNLS	Universal National Language Support library.
UNVCONF	Stonebranch Solutions configuration library.
UNVDB	Universal Broker Database HFS data set.
UNVSPOOL	Universal Spool Database HFS data set.
UNVTRACE	Application trace ddname.
SYSOUT	USLRM standard error ddname.
SYSPRINT	USLRM standard output ddname.

Table 30.2 Universal Spool Remove for z/OS – DD Statements in JCL Procedure

30.3.4 JCL

Figure 30.2, below, illustrates the Universal Spool Remove for z/OS JCL.

```
//STEP1    EXEC PGM=USLRM
//STEPLIB  DD  DISP=SHR,DSN=UNV.SUNVLOAD
//UNVNLS   DD  DISP=SHR,DSN=UNV.SUNVNLS
//UNVCONF  DD  DISP=SHR,DSN=&CFGPRE..UNVCONF
//UNVDB    DD  DISP=SHR,DSN=UNV.UNVDB
//UNVSPOOL DD  DISP=SHR,DSN=UNV.UNVSPOOL
//UNVTRACE DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 command options
/*
```

If zFS data sets are being used instead of the default HFS data sets, the UNVDB and UNVSPOOL ddnames cannot be used to allocate the zFS data sets. The ddnames must be removed along with the DBPRE procedure parameter.

Figure 30.2 Universal Spool Remove for z/OS – JCL

30.3.5 Configuration Options

Table 30.3, below, identifies the Universal Spool Remove for z/OS configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
COMPONENT	Component identifier for which records will be removed.
HELP	Writes a description of the command options and their format.
MESSAGE_LEVEL	Level of messages that will be written.
MOUNT_POINT	HFS directory in which the HFS databases allocated to ddnames UNVDB and UNVSPOOL are mounted.
VERSION	Writes the program version and copyright information.

Table 30.3 Universal Spool Remove for z/OS - Configuration Options

30.3.6 Command Line Syntax

Figure 30.4, below, illustrates the command syntax — using the command line, long form of the configuration options — of Universal Spool Remove for z/OS.

```
uslrm
-component cid
[-mount_point dir]
[-level {audit|info|warn|error}]

uslrm
{ -help | -version }
```

Figure 30.3 Universal Spool Remove for z/OS - Command Line Syntax

30.3.7 zFS Support

Universal Spool Remove (**USLRM**) and Universal Spool List Configuration Options (**USLIST**) obtain the HFS data set names from **UNVDB** and **UNVSPOOL** ddnames. zFS data sets cannot be provided via ddnames.

To obtain the zFS or HFS data set names, **USLRM** and **USLIST** allocate and parse the Universal Broker's configuration member, **UBRCFG00**, for the UNIX_DB_DATA_SET and UNIX_SPOOL_DATA_SET options. If these options are not found, **USLRM** and **USLIST** assume that HFS data sets are being used and refer the **UNVDB** and **UNVSPOOL** ddnames for the HFS data set names.

Note: Previous versions of **USLRM** and **USLIST** did not allocate the Universal Broker's configuration file.

USLRM and **USLIST** have always mounted the HFS data sets if they were not already mounted. This is the case with zFS data sets as well.

The Stonebranch Solutions configuration PDSE is allocated to **UNVCONF** in the **USLLSPRC** and **USLRMPRC** JCL procedures in the **SUNVSAMP** library. The only other changes to **USLLSPRC** and **USLRMPRC** is the addition of a comment that the **UNVDB** and **UNVSPOOL** ddnames must be commented out to use zFS data sets.

USLRM and **USLIST** now write messages UNV2264I and UNV2265I, which provide information on the Broker's database and spool, respectively. The messages also provide the file system type, data set name, and mount point.

30.4 Universal Spool Remove for Windows and UNIX

Table 30.4, below, identifies the databases from which Universal Spool Remove can remove records.

Database Name	Default File Name
Universal Broker Component Database	bcomponent.db
Universal Server Component Database	scomponent.db
Universal Server Spool Databases	spool.stdin.*COMPID*.db spool.stdout.*COMPID*.db spool.stderr.*COMPID*.db
Universal Event Monitor Spool Database	uems.db
Note: In the Universal Server Spool Databases file names, *COMPID* is the component ID assigned to the Server instance.	

Table 30.4 Universal Spool Remove for Windows and UNIX - Universal Spool Databases

UNIX

By default, the database files are located in the **/var/opt/universal/spool** directory.

The program file is located in the Universal Spool installation directory **bin** directory, which defaults to **/opt/universal/uspool/bin**.

Windows

By default, the database files are stored in the **C:\Program Files\Universal\spool** directory.

The Universal Spool Remove program file, **uslrm.exe**, is located in the **bin** subdirectory of the Universal Spool installation directory, which defaults to **C:\Program Files\Universal\uspool**.

30.4.1 Configuration Options

Table 30.5, below, identifies the Universal Spool Remove for Windows and UNIX configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
BROKER_SPOOL_DIR	Directory location in which the Universal Broker Component database is located.
COMPONENT	Component identifier for which records will be removed.
HELP	Writes a description of the command options and their format.
MESSAGE_LEVEL	Level of messages that will be written.
UCMD_SPOOL_DIR	Directory location in which the Universal Server Component database is located.
UEM_SERIALNO	Serial number of the Universal Event Monitor spool database record to remove.
VERSION	Writes the program version and copyright information.

Table 30.5 Universal Spool Remove for Windows and UNIX - Configuration Options

30.4.2 Command Line Syntax

Figure 30.4, below, illustrates the command line syntax — using the command line, long form of the configuration options — of Universal Spool Remove for Windows and UNIX.

```
uslrm
{ -component cid | -uem_serialno serno }
[-brokerspooldir dir]
[-ucmdspooldir dir]
[-level {audit|info|warn|error}]

uslrm
{ -help | -version }
```

Figure 30.4 Universal Spool Remove for Windows and UNIX - Command Line Syntax

30.5 Universal Spool Remove for IBM i

This section describes Universal Spool Remove for IBM i operating system.

The Universal Spool Remove utility can be used to read the databases listed in Table 30.6, below.

Database Name	Database File Name
Universal Broker Component Database	**UBR_CMP_DB**
Universal Server Component Database	**SRV_CMP_DB**

Table 30.6 Universal Spool Remove for IBM i – Databases

The spool files are located in library **UNVUSL420**.

30.5.1 Configuration Options

Table 30.7, below, identifies the Universal Spool Remove for IBM i configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
COMPONENT	Component identifier for which records will be removed.
MESSAGE_LEVEL	Level of messages that will be written.
VERSION	Writes the program version and copyright information.

Table 30.7 Universal Spool Remove for IBM i – Configuration Options

30.5.2 Command Line Syntax

Figure 30.5, below, illustrates the command line syntax — using the URMVSE parameter form of command line options — of Universal Spool Remove for IBM i.

```
URMVSE
[COMPONENT(cid)]
[LEVEL(*{audit|info|warn|error})]

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

Figure 30.5 Universal Spool Remove for IBM i – Command Line Syntax

CHAPTER 31

Universal Spool Remove Configuration Options

31.1 Overview

This chapter provides detailed information on the configuration options available for use with the Universal Spool Remove utility.

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

Information on how these options are used is documented in Chapter 30 Universal Spool Remove.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

31.2 Configuration Options List

Table 31.1, below, identifies the Universal Spool Remove configuration options.

Option Name	Description	Page
BROKER_SPOOL_DIR	Directory location in which the Universal Broker Component database is located.	452
COMPONENT	Component identifier for which records will be removed.	453
HELP	Writes a description of the command options and their format.	454
MESSAGE_LEVEL	Level of messages that will be written.	455
MOUNT_POINT	HFS directory in which the HFS databases allocated to ddnames UNVDB and UNVSPOOL are mounted.	456
UCMD_SPOOL_DIR	Directory location in which the Universal Command Server Component database is located.	457
UEM_SERIALNO	Serial number of the Universal Event Monitor spool database record to remove.	458
VERSION	Writes the program version and copyright information.	459

Table 31.1 Universal Spool Remove Configuration Options

31.3 BROKER_SPOOL_DIR

Description

The BROKER_SPOOL_DIR option specifies the directory in which the Universal Broker component database (**bcomponent.db**) is located.

If this option is not used to specify the directory, the directory is read from the Universal Broker configuration file.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-b *directory*			√	√	
Command Line, Long Form	-brokerspooldir *directory*			√	√	
URMVSE Parameter	n/a					

Values

directory is the directory in which the Universal Broker component database is located.

31.4 COMPONENT

Description

The COMPONENT option specifies the ID of a component for which records will be removed from all databases.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-c *cid*			√	√	√
Command Line, Long Form	-component *cid*			√	√	√
URMVSE Parameter	COMPONENT (*cid*)	√				

Values

cid is the component ID.

31.5 HELP

Description

The HELP option writes a description of the command options and their format.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-h			√	√	√
Command Line, Long Form	-help			√	√	√
URMVSE Parameter	n/a					

Values

(There are no values for this option.)

31.6 MESSAGE_LEVEL

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l *level*			√	√	√
Command Line, Long Form	-level *level*			√	√	√
URMVSE Parameter	LEVEL (**level*)	√				

Values

level indicates either of the following level of messages:

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

z/OS and IBM i

[Default is info.]

UNIX and Windows

[Default is warn.]

31.7 MOUNT_POINT

Description

The MOUNT_POINT option specifies the HFS directory in which the HFS database allocated to ddnames UNVDB and UNVSPOOL are mounted.

The actual mount points will be subdirectories named after the HFS data set names being mounted.

If the mount points do not exist, they are created by Universal Spool List.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-mount_point *dir*					√
URMVSE Parameter	n/a					

Values

dir is the HFS directory in which the HFS databases are mounted.

[Default is */tmp*.]

31.8 UCMD_SPOOL_DIR

Description

The UCMD_SPOOL_DIR option specifies the directory in which the Universal Command Server component database (**scomponent.db**) is located.

If this option is not used to specify the directory, the directory of the Universal Broker component database is used (see the BROKER_SPOOL_DIR option).

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-u *dir*			√	√	
Command Line, Long Form	-ucmdspooldir *dir*			√	√	
URMVSE Parameter	n/a					

Values

dir is the directory of the Universal Command Server component database.

31.9 UEM_SERIALNO

Description

The UEM_SERIALNO option specifies the serial number of the Universal Event Monitor spool database record to remove.

Use the Universal Spool List utility to generate a complete list of all UEM spool records (see Chapter 28 Universal Spool List).

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-s *serno*			√	√	
Command Line, Long Form	-uem_serialno *serno*			√	√	
URMVSE Parameter	n/a					

Values

serno is the serial number of the database record to remove.

31.10 VERSION

Description

The VERSION option writes the program version and copyright information.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-v			√	√	√
Command Line, Long Form	-version			√	√	√
URMVSE Parameter	VERSION(**option*)	√				

Values

UNIX, Windows, z/OS

(There are no values for this option.)

IBM i

Valid values for *option* are:

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

CHAPTER 32

Universal Submit Job

32.1 Overview

The Universal Submit Job (USBMJOB) utility is a command for the IBM i environment that encapsulates the IBM Submit Job (SBMJOB) command.

USBMJOB builds on the functionality of SBMJOB by providing a job submission command that better suits the needs of a remote user issuing IBM i commands via Universal Command.

Note: Users never should call USBMJOB directly.

32.1.1 Functions

USBMJOB performs four main functions:

1. Submits commands in their own easily customized job environment.
2. Monitors submitted jobs to completion and sets a return code based on the end code and message severity codes of the submitted job.
3. Provides a facility for the remote handling of inquiry messages generated by the submitted job. Inquiry messages are received and replied to from the z/OS console via the Universal Write-to-Operator utility (available only for z/OS managers).
4. Returns spooled output generated by the submitted job on standard output. The joblog generated by the submitted job is written to standard error and, optionally, to a job log output queue.

32.2 Usage

Universal Submit Job (USBMJOB) submits a user command in its own job. USBMJOB supports all of the command parameters offered by the SBMJOB command that make sense for an unscheduled job submitted from a batch environment. This enables the user to fully customize the job environment for the user command. Internally, the SBMJOB command is called to submit the user job.

USBMJOB remains active for the duration of the submitted job. USBMJOB continuously monitors the state of the submitted job at a user-defined polling interval. In addition to monitoring for job completion, USBMJOB can detect when the submitted job is waiting for a reply to an inquiry message.

USBMJOB provides the option for inquiry messages generated from the submitted job to be sent to a remote z/OS console. Replies received from the z/OS console are sent as reply messages to the corresponding inquiry message.

When the submitted job completes, USBMJOB writes the joblog for the submitted job to standard error and, optionally, to a job log output queue. The spooled output generated by the submitted job is written to standard output.

After USBMJOB has finished processing the submitted job, it completes by issuing an escape message to the external message queue. The escape message sets the return code for the USBMJOB command. The severity code of the escape message indicates the return code.

If the user job submitted by USBMJOB completes normally (end code < 20), the severity code for the USBMJOB escape message will be 0. If the user job submitted by USBMJOB completes abnormally (end code > 10), the severity code for the USBMJOB escape message will be set to the highest severity code generated by the submitted job.

By issuing an escape message with a severity code correlated with the submitted job's end code/highest severity code, USBMJOB allows the Universal Command Server to pick up the severity code. This, in turn, allows the USBMJOB command to propagate its return code to the Universal Command Server.

Two helper commands - supplied by Stonebranch, Inc. - are called internally by USBMJOB:

- Universal Job initializer (UJOBINIT)
- Universal Message Handler (UMSGHNDLR)

Universal Job initializer (UJOBINIT) is called from within the job submitted by USBMJOB. UJOBINIT performs initialization that allows USBMJOB to redirect the joblog of the submitted job and then issues the user command.

Output

In addition to joblog redirection, USBMJOB returns the jobs spooled output to the Universal Command Manager via standard output.

When the submitted job user name and the user profile name passed to the Universal Command Manager differ, USBMJOB requires *SPLCTL authority to retrieve the spooled output. USBMJOB receives this *SPLCTL authority from the **UNVUBR420** user profile.

If *SPLCTL special authority is removed from the **UNVUBR420** user profile, USBMJOB will fail if:

1. Security is set to DEFAULT via the UNVCONF(UCMDS) configuration file and the submitted job user name and the user profile name passed to the Universal Command Manager differ.

   Example:

   ```
   ucmd -c "usbmjob cmd(dsplib abc) user(abc)" -i rmtsys -u myuser
   -w mypwd
   ```

2. Security is set to NONE via the UNVCONF(UCMDS) configuration file and a user name is specified for the submitted job.

   Example:

   ```
   ucmd -c "usbmjob cmd(dsplib abc) user(abc)" -i rmtsys
   ```

32.2.1 Configuration Options

The Universal Submit Job command performs operations specified by configuration options. The options have associated values that describe the actions to take.

The USBMJOB configuration options are separated into two categories:

1. USBMJOB-specific
2. SBMJOB encapsulated

USBMJOB-Specific Configuration Options

Table 32.1, below, identifies the USBMJOB-specific configuration options, which control the way that the submitted job is monitored and administered. Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
COMMAND	Command that runs in the submitted batch job.
COPY_SPOOL_FILES	Specification for whether or not spooled output files generated by the submitted job be copied to standard output.
ENCRYPTED_COMMAND_FILE	Name of an encrypted command file.
JOB_LOG_LIBRARY	Library into which the job log will be placed.
JOB_STATUS_POLLING_INTERVAL	Number of seconds that USBMJOB will sleep between calls to check the status of the submitted job.
KEY	Encryption key used to encrypt the encrypted command file.
REMOTE_MESSAGE_PREFIX	Text string that prefixes any remote message sent by USBMJOB.
REMOTE_REFRESH_INTERVAL	Time that a remote reply message will remain on a remote console without being replied to before it times out.
REMOTE_REPLY_COMMAND_PATH	Path (including the executable name) to the remote message handler (**uwto**).
REMOTE_REPLY_HOST	Host name of the remote system on which the **uwto** command is executed.
REMOTE_REPLY_PORT	Port of the Universal Broker on the remote system on which the Universal WTO command is executed.
REMOTE_REPLY_USER_ID	User ID for the remote system where the **uwto** command resides.
REMOTE_REPLY_USER_PWD	Password for user on the remote system where **uwto** resides.
SPECIFY_PRINT_CONTROL_CHARS	Print control characters (if any) that are to replace the spooled file's internal print control characters.
TRACE	Specification for whether or not trace information will be printed to standard error.
USE_REMOTE_REPLY_FACILITY	Specification for whether or not USBMJOB will use the remote reply facility.

Table 32.1 Universal Submit Job - USBMJOB-Specific Configuration Options

SBMJOB Encapsulated Configuration Options

The SBMJOB encapsulated configuration options have a one-to-one relationship with the IBM SBMJOB command parameters of the same name.

Option Name	USBMJOB Parameter
ALLOW DISPLAY BY WRKSBMJOB	DSPSBMJOB({***yes** | ***no**})
CODED CHARACTER SET ID	CCSID({ ***current** | ***sysval** | ***usrprf** | ***hex** | *coded_character_set_identifier* })
COPY ENVIRONMENT VARIABLES	CPYENVVAR({ ***no** | ***yes** })
COUNTRY ID	CNTRYID({ ***current** | ***sysval** | ***usrprf** | *country_id*})
CURRENT LIBRARY	CURLIB({ ***current** | ***usrprf** | ***crtdft** | *current_library_name* })
HOLD ON JOB QUEUE	HOLD({ ***jobd** | ***no** | ***yes** })
INITIAL LIBRARY LIST	INLLIBL({ ***current** | ***jobd** | ***sysval** | ***none** | *library_name...*})
INLASPGRP	INLASPGRP(*current | *jobd | *none)
INQUIRY MESSAGE REPLY	INQMSGRPY({ ***jobd** | ***rqd** | ***dft** | ***sysrpyl** })
JOB DATE	DATE({ ***jobd** | ***sysval** | *job_date* })
JOB DESCRIPTION	JOBD({ ***usrprf** | [*library/*] *job_description* })
JOB MESSAGE QUEUE FULL ACTION	JOBMSGQFL({ ***jobd** | ***sysval** | ***nowrap** | ***wrap** | ***prtwrap** })
JOB MESSAGE QUEUE MAXIMUM SIZE	JOBMSGQMX({ ***jobd** | ***sysval** | *maximum_size_of_job_message_queue* })
JOB NAME	JOB({ ***jobd** | *job_name* })
JOB PRIORITY	JOBPTY(*priority*)
JOB QUEUE	JOBQ({ ***jobd** | [*library/*] *job_queue* })
JOB SWITCHES	SWS({ ***jobd** | *switch_settings* })
LANGUAGE ID	LANGID({***current** | ***sysval** | ***usrprf** | *language_id* })
LOG CL PROGRAM COMMANDS	LOGCLPGM({***jobd** | ***no** | ***yes** })
OUTPUT PRIORITY	OUTPTY(*priority*)
OUTPUT QUEUE	OUTQ({ ***current** | ***usrprf** | ***dev** | ***jobd** | [*library/*] *output_queue* })
PRINT DEVICE	PRTDEV({ ***current** | ***usrprf** | ***sysval** | ***jobd** | *printer_device_name* })
PRINT TEXT	PRTTXT(*text*)
SORT SEQUENCE	SRTSEQ({ ***current *sysval *usrprf *hex *langidunq *langidshr** | [{ ***libl** | ***curlib** | *library_name* } /] *table_name* })
SUBMITTED FOR	SBMFOR(*job_number* / *user* / *job_name*)
SYSTEM LIBRARY LIST	SYSLIBL({ ***current** | ***sysval** })
USER	USER({ ***current** | ***jobd** | *user_name* })

Table 32.2 Universal Submit Job - SBMJOB Encapsulated Configuration Options

Note: The values for these options are passed directly to the SBMJOB command internally. Therefore, the effect these options have on a submitted job will be equal to that documented by IBM for the Submit Job command.

Refer to the documentation provided by IBM for the SBMJOB implementation being used for an accurate description of the effect that these options will have on that implementation.

32.2.2 Command Line Syntax

Figure 32.1, below, illustrates the command line syntax of Universal Submit Job.

The command line name of every USBMJOB-specific configuration option is a link to detailed information about that option.

```
USBMJOB
[CMD(command)]
[JOBLOGLIB(library)]
[POLL(seconds)]
[RMTRPY( {*yes|*no} )]
[RMTREFRESH(seconds)]
[RMTMSGPRX(prefix)]
[RMTUSER(userid)]
[RMTPWD(password)]
[ECMFILE(cmd_file) [ECMMBR(member)] [KEY(key)] ]
[RMTHOST(host)]
[RMTPORT (port)]
[MSGCMDPATH(path)]
[JOB( { *jobd | job_name })]
[JOBD( { *usrprf | [library/ ] job_description } )]
[JOBMSGQFL( {*jobd|*sysval|*nowrap|*wrap|*prtwrap} )]
[JOBQ( { *jobd | [ library/ ] job_queue } )]
[JOBPTY(priority)]
[OUTPTY(priority)]
[PRTDEV( { *current | *usrprf | *sysval | *jobd | printer_device_name } )]
[OUTQ( { *current | *usrprf | *dev | *jobd | [ library/ ] output_queue } )]
[CPYSPLF( {*yes|*no} )]
[SPLFCTLCHR ( {*none|*fcfc|*prtctl|*s36fmt} )]
[TRACE( {*yes|*no} )]
[USER( { *current | *jobd | user_name } )]
[PRTTXT(text)]
[SYSLIBL( {*current|*sysval} )]
[CURLIB( { *current | *usrprf | *crtdft | current_library_name } )]
[INLLIBL({ *current | *jobd | *sysval | *none |library_name...} )]
[LOGCLPGM( {*jobd|*no|*yes} )]
[INQMSGRPY( {*jobd|*rqd|*dft|*sysrpyl} )]
[INLASPGRP({ *current | *jobd | *none} )]
[HOLD( {*jobd|*no|*yes} )]
[DATE( {*jobd|*sysval|job_date} )]
[SWS( {*jobd|switch_settings} )]
[DSPSBMJOB( {*yes|*no} )]
[SRTSEQ( {*current*sysval*usrprf*hex*langidunq*langidshr |
         [*libl | *curlib | library/] table_name} )]
[LANGID( {*current|*usrprf|*sysval|language_id} )]
[CNTRYID( {*current|*usrprf|*sysval|country_id} )]
[CCSID( {*current|*usrprf|*sysval|*hex|coded_character_set_identifier} )]
[SBMFOR(job_number/user/job_name)]
[JOBMSGQMX( {*jobd|*sysval|*maximum_size_of_job_message_queue} )]
[CPYENVVAR( {*yes|*no} )]
```

Figure 32.1 Universal Submit Job - Command Line Syntax

Command Line Syntax Rules

Values for configuration options that contain special characters require:

- Double (**"**) quotation marks when executed from an MVS Universal Command Manager.
- Single (**'**) quotation marks when executed from an IBM i Universal Command Manager.

For example, the following is correct when executed from a z/OS Universal Command Manager:

```
MSGCMDPATH("/usr/local/universal/bin/uwto")
```

However, the following is incorrect when executed from a z/OS Universal Command Manager; it will create a syntax error:

```
RMSGCMDPATH('/usr/local/universal/bin/uwto')
```

32.3 Remote Reply Facility

Universal Submit Job can detect when messages sent by the submitted job require a reply. If the Remote Reply Facility is turned on (RMTRPY(***yes**)), USBMJOB will send all messages requiring a reply to a remote z/OS console. Replies to the inquiry messages are received from the z/OS console and sent to the IBM i message queue waiting for the reply.

The Remote Reply Facility requires an installation of Universal Command on the IBM i system where the USBMJOB command will run and an installation of Universal Command on the remote z/OS system where the inquiry messages will be sent to for reply.

- Universal Command on the IBM i must be at Universal Command 1.2.1 level 7 or greater.
- Universal Command on the remote z/OS system, where the inquiry messages will be sent, must be at level 12 or greater.

The Remote Reply Facility used by USBMJOB is comprised of a group of Stonebranch Inc. utilities that work together (Table 32.3).

Utility	Platform
Universal Submit Job	IBM i
Universal Message Handler	IBM i
Universal Command	IBM i and z/OS
Universal UWTO	z/OS Unix System Services (USS)

Table 32.3 Remote Reply Facility Utilities

Universal Submit Job and Universal Message Handler are part of the Universal Command for IBM i licensed product. These utilities were added to the product in Universal Command 1.2.1 level 7).

Universal WTO (UWTO) is a command line utility for the z/OS Unix System Services (USS) environment, as of level 12. The path to the UWTO command is specified on the **MSGCMDPATH** parameter of the USBMJOB command.

See Chapter 34 Universal Write-to-Operator for more information on the UWTO command.

32.4 Return Codes

The Universal Submit Job command completes by sending an escape message to the external message queue. The severity code of this escape message is the USBMJOB return code. The USBMJOB return code is determined in the following way:

When the submitted job completes, USBMJOB scans the job log, examining the severity code of all *ESCAPE, *NOTIFY, *STATUS, and function check messages.

- If the submitted job completed abnormally (end code > 10), the USBMJOB return code will be set to the highest severity code examined in the submitted job's job log.
- If the submitted job completes normally (end code < 20), the examined severity codes will be ignored and the USBMJOB return code will be set to 0.

Setting the return code with an escape message allows the Universal Command Server to pick up the return code for use in its return code processing.

The range of possible severity code values is 0 through 99.

If USBMJOB encounters a processing error, a severity code of 99 will be used, regardless of severity codes that may have been examined from the submitted job.

CHAPTER 33

Universal Submit Job Configuration Options

33.1 Overview

This chapter provides detailed information on the configuration options available for use with the Universal Submit Job (USBMJOB) utility.

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

Information on how these options are used is documented in Chapter 32 Universal Submit Job.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

33.2 Configuration Options List

The Universal Submit Job configuration options are separated into two categories:

1. USBMJOB-Specific Configuration Options
2. SBMJOB Encapsulated Configuration Options

33.2.1 USBMJOB-Specific Configuration Options

Table 33.1 identifies the USBMJOB-specific configuration options.

Option Name	Description	Page
COMMAND	Command that runs in the submitted batch job.	474
CPYSPLF	Specification for whether or not spooled output files generated by the submitted job be copied to standard output.	474
ENCRYPTED_COMMAND_FILE	Name of an encrypted command file.	474
JOB_LOG_LIBRARY	Library into which the job log will be placed.	475
JOB_STATUS_POLLING_INTERVAL	Number of seconds that USBMJOB will sleep between calls to check the status of the submitted job.	475
REMOTE_MESSAGE_PREFIX	Text string that will prefix any remote messages sent by USBMJOB.	476
REMOTE_REFRESH_INTERVAL	Time that a remote reply message will remain on a remote console without being replied to before it times out.	476
REMOTE_REPLY_COMMAND_PATH	Path (including the executable name) to the remote message handler (**uwto**).	476
REMOTE_REPLY_HOST	Host name of the remote system on which the **uwto** command is executed.	477
REMOTE_REPLY_PORT	Port of the Universal Broker on the remote system on which the Universal WTO command is executed.	477
REMOTE_REPLY_USER_ID	User ID for the remote system where the **uwto** command resides.	477
REMOTE_REPLY_USER_PWD	Password for the user on the remote system where the **uwto** command resides.	478
SPECIFY_PRINT_CONTROL_CHARS	Print control characters (if any) that are to replace the spooled file's internal print control characters.	478
TRACE	Specification for whether or not trace information will be printed to standard error.	478
USE_REMOTE_REPLY_FACILITY	Specification for whether or not USBMJOB will use the remote reply facility.	479

Table 33.1 Universal Submit Job Configuration Options - USBMJOB-Specific

33.2.2 SBMJOB Encapsulated Configuration Options

The SBMJOB encapsulated configuration options (Table 33.2, below) have a one-to-one relationship with the IBM SBMJOB command parameters of the same name.

Option Name	USBMJOB Parameter
ALLOW DISPLAY BY WRKSBMJOB	DSPSBMJOB({***yes** | ***no**})
CODED CHARACTER SET ID	CCSID({ ***current** | ***sysval** | ***usrprf** | ***hex** | *coded_character_set_identifier* })
COPY ENVIRONMENT VARIABLES	CPYENVVAR({ ***yes** | ***no** })
COUNTRY ID	CNTRYID({ ***current** | ***sysval** | ***usrprf** | *country_id*})
CURRENT LIBRARY	CURLIB({ ***current** | ***usrprf** | ***crtdft** | *current_library_name* })
HOLD ON JOB QUEUE	HOLD({ ***jobd** | ***no** | ***yes** })
INITIAL LIBRARY LIST	INLLIBL({ ***current** | ***jobd** | ***sysval** | ***none** | *library_name...*})
INLASPGRP	INLASPGRP({***current** | ***jobd** | ***none**})
INQUIRY MESSAGE REPLY	INQMSGRPY({ ***jobd** | ***rqd** | ***dft** | ***sysrpyl** })
JOB DATE	DATE({ ***jobd** | ***sysval** | *job_date* })
JOB DESCRIPTION	JOBD({ ***usrprf** | [*library/*] *job_description* })
JOB MESSAGE QUEUE FULL ACTION	JOBMSGQFL({ ***jobd** | ***sysval** | ***nowrap** | ***wrap** | ***prtwrap** })
JOB MESSAGE QUEUE MAXIMUM SIZE	JOBMSGQMX({ ***jobd** | ***sysval** | *maximum_size_of_job_message_queue* })
JOB NAME	JOB({ ***jobd** | *job_name* })
JOB PRIORITY	JOBPTY(*priority*)
JOB QUEUE	JOBQ({ ***jobd** | [*library/*] *job_queue* })
JOB SWITCHES	SWS({ ***jobd** | *switch_settings* })
LANGUAGE ID	LANGID({***current** | ***sysval** | ***usrprf** | *language_id* })
LOG CL PROGRAM COMMANDS	LOGCLPGM({***jobd** | ***no** | ***yes** })
OUTPUT PRIORITY	OUTPTY(*priority*)
OUTPUT QUEUE	OUTQ({ ***current** | ***usrprf** | ***dev** | ***jobd** | [*library/*] *output_queue* })
PRINT DEVICE	PRTDEV({ ***current** | ***usrprf** | ***sysval** | ***jobd** | *printer_device_name* })
PRINT TEXT	PRTTXT(*text*)
SORT SEQUENCE	SRTSEQ({ ***current** ***sysval** ***usrprf** ***hex** ***langidunq** ***langidshr** | [{ ***libl** | ***curlib** | *library_name* } /] *table_name* })
SUBMITTED FOR	SBMFOR(*job_number* / *user* / *job_name*)
SYSTEM LIBRARY LIST	SYSLIBL({ ***current** | ***sysval** })
USER	USER({ ***current** | ***jobd** | *user_name* })

Table 33.2 Universal Submit Job - SBMJOB Encapsulated Configuration Options

33.2.3 Universal Command Server Options Affecting USBMJOB

The Universal Command Server JOBLOG_COPY_KEEP configuration option controls the keeping a copy of the job log returned via standard output on the local iSeries system.

See Chapter 3 Universal Command Server Configuration Options in the Universal Command 4.2.0 Reference Guide for specific information about JOBLOG_COPY_KEEP.

33.3 Universal Submit Job (USBMJOB) Specific Options

33.3.1 COMMAND

The COMMAND option specifies a command that runs in the submitted batch job.

The command can be a maximum of 3000 characters.

USBMJOB Parameter: CMD(*command*)

33.3.2 COPY_SPOOL_FILES

The CPYSPLF option specifies whether or not spooled output files generated by the submitted job are copied to standard output.

USBMJOB Parameter: CPYSPLF({ *yes | *no })

- **YES* will cause spooled output files generated by the submitted job to be written to standard output.
- **NO* will prevent spooled output files generated by the submitted job from being written to standard output.

33.3.3 ENCRYPTED_COMMAND_FILE

The ENCRYPTED COMMAND FILE option specifies the name of an encrypted command file.

The Encrypted Command File option is used when the Remote Reply Facility is used. In order to issue inquiry commands remotely, Universal Submit Job must log on to the remote system. This is accomplished by internally issuing a command to the Universal Command Manager. Therefore, the Encrypted Command File for Universal Submit Job serves the same security purposes as the Encrypted Command File for Universal Command Manager. However, encrypted command files for Universal Submit Job should not contain commands. The command will be issued by Universal Submit Job.

Use the Universal Encrypt utility provided with Universal Command to encrypt a plain text command file. If a key was used to encrypt the file, the same key must be supplied using the KEY option.

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

USBMJOB Parameter: ECMFILE(*cmd_file*) [ECMMBR(*member*)]

33.3.4 JOB_LOG_LIBRARY

The JOB LOG LIBRARY option specifies a library into which the job log will be placed.

The job log will be sent to files **USJPnnnnnn** and **USJSnnnnnn** (**nnnnnn** is the job number):

- **USJPnnnnnn** contains the primary messages of the job log.
- **USJSnnnnnn** contains the secondary messages of the job log.

The member name for both primary and secondary joblog files is of the form **Cnnnnnn**, where **nnnnnn** is the job number of the job that USBMJOB is running under.

USBMJOB Parameter: JOBLOGLIB(*library*)

33.3.5 JOB_STATUS_POLLING_INTERVAL

The JOB STATUS POLLING INTERVAL option specifies the number of seconds that USBMJOB will sleep between calls to check the status of the submitted job.

USBMJOB Parameter: POLL(*seconds*)

33.3.6 KEY

The KEY option specifies the encryption key used to encrypt the encrypted command file specified by ENCRYPTED_COMMAND_FILE. If no encryption key is specified, a default key is used.

USBMJOB Parameter: KEY(*key*)

33.3.7 REMOTE_MESSAGE_PREFIX

The REMOTE MESSAGRE PREFIX option allows the user to specify a text string up to 12 characters in length that will prefix any remote messages sent by USBMJOB.

This prefix can make it easier to relate a remote message to it's associated job.

USBMJOB Parameter: RMTMSGPRFX(*prefix*)

33.3.8 REMOTE_REFRESH_INTERVAL

The REMOTE REFRESH INTERVAL option specifies a time interval (in seconds) that controls how long a remote reply message will remain on a remote console without being replied to before it will time out.

If the remote reply message times out, the message will be removed from the remote console. Universal Submit Job then will determine if the user job still is waiting for a reply. If it is, the remote reply message will be re-sent to the remote console.

USBMJOB Parameter: RMTREFRESH(*seconds*)

[Default is *0* (wait indefinitely for a reply).]

33.3.9 REMOTE_REPLY_COMMAND_PATH

The REMOTE REPLY COMMAND PATH option specifies the path (including the executable name) to the remote message handler (**`uwto`**). (See Chapter 34 Universal Write-to-Operator for information on the Universal WTO utility.)

USBMJOB Parameter: MSGCMDPATH(*command_path*)

33.3.10 REMOTE_REPLY_HOST

The REMOTE_REPLY HOST option specifies the host name of the Universal Broker on the remote system on which the Universal WTO command is executed. (See Chapter 34 Universal Write-to-Operator for information on the Universal WTO utility.)

USBMJOB Parameter: RMTHOST(*host*)

The format of *host* is either:

- Host name (for example, *homer*)
- Numeric address (for example, *10.20.30.40*)

33.3.11 REMOTE_REPLY_PORT

The REMOTE REPLY PORT option specifies the port of the Universal Broker on the remote system on which the Universal WTO command is executed (see Section 33.3.10 REMOTE_REPLY_HOST). (See Chapter 34 Universal Write-to-Operator for information on the Universal WTO utility.)

USBMJOB Parameter: RMTPORT(*port*)

Valid values for *port* are:

- Port number
- Service name

[Default is *7887*.]

33.3.12 REMOTE_REPLY_USER_ID

The REMOTE REPLY USER ID option specifies the user ID for the remote system where the Universal WTO command resides. (See Chapter 34 Universal Write-to-Operator for information on the Universal WTO utility.)

USBMJOB Parameter: RMTUSER(*userid*)

33.3.13 REMOTE_REPLY_USER_PWD

The REMOTE REPLY USER PASSWORD option specifies the password for the user on the remote system where the Universal WTO command resides. (See Chapter 34 Universal Write-to-Operator for information on the Universal WTO utility.)

USBMJOB Parameter: RMTPWD(*password*)

33.3.14 SPECIFY_PRINT_CONTROL_CHARS

Specifies which print control characters (if any) are to replace the spooled file's internal print control characters.

USBMJOB Parameter: SPLFCTLCHR(**characters*)

Valid values for *characters* are:

- *NONE*
- *FCFC*
- *PRTCTL*
- *S36FMT*

See the CL Reference, SC41-5722 for more information on these values.

33.3.15 TRACE

The TRACE option specifies whether or not trace information will be written to standard error.

USBMJOB Parameter: TRACE({ *yes| *no})

- **YES* will cause trace information to be written to standard error.
- **NO* will prevent trace information from being written to standard error.

Note: Use TRACE only as directed by Stonebranch, Inc. Customer Support.

33.3.16 USE_REMOTE_REPLY_FACILITY

The USER REMOTE REPLY FACILITY option specifies whether or not Universal Submit Job will use the remote reply facility.

The remote reply facility will detect messages, issued by the submitted job, that require a reply. The message then will be passed on to a remote system for a reply. When the reply is received, the reply will be sent to the message queue that is waiting for the reply.

USBMJOB Parameter: RMTRPY({ *yes | *no })

- ***YES*** will cause USBMJOB to use the remote reply facility.
- ***NO*** will cause USBMJOB to ignore message wait conditions for the submitted job.

[Default is *NO.]

CHAPTER 34

Universal Write-to-Operator

34.1 Overview

The Universal WTO (UWTO) utility is a command line utility for the z/OS UNIX System Services (USS) environment.

Universal WTO issues two types of messages to z/OS consoles:

1. Write-To-Operator (WTO) messages
2. Write-To-Operator-with-Reply (WTOR) messages

Note: UWTO became available for the z/OS USS environment with Universal Command 2.2.0, Level 12.

34.2 Usage

Universal WTO either:

- Writes a message to the z/OS console, and then ends (WTO).
- Writes a message to the z/OS console and waits for a requested reply (WTOR).

The type of message to be written (WTO or WTOR) is specified via the REPLY option. If WTOR is specified, the message is written to the console as a WTOR message and Universal WTO waits for a reply. The message reply is written to stdout.

34.2.1 Return Codes

The UWTO command ends with specific return codes indicating the success of the requested action.

Table 34.1, below, describes these return codes.

Return Code	Description
0	Process was successful.
1	Message was written to the console, but a warning was issued regarding a requested option. A detailed message is written to standard error.
2	WTOR request timed out waiting for an operator reply.
3	Error occurred when attempting to write the message. No message was written to the console.
10	Error in the command line options was detected. No processing occurred.

Table 34.1 Universal WTO - Return Codes

34.2.2 Configuration Options

Table 34.2, below, identifies the Universal WTO configuration options.

Each **Option Name** is a link to detailed information about that configuration option.

Option Name	Description
CONSOLE_ID	ID of the console to which to route the message.
CONSOLE_NAME	Name of the console to which to route the message.
HELP	Writes a description of the command options and their format.
KEY	Key to associate with the message.
MESSAGE	Text to write to the z/OS operator console.
MESSAGE_LEVEL	Level of messages to write.
REPLY	Directs UWTO is issue a WTOR message and wait for an operator reply to the message.
TIMEOUT	Number of seconds to wait for a WTOR operator reply.
VERSION	Writes the program version and copyright information.

Table 34.2 Universal WTO - Configuration Options

34.2.3 Command Line Syntax

Figure 34.1, below, illustrates the syntax – using the long form of command line options – of Universal WTO.

```
uwto
[-msg message]
[-reply {yes|no} [-timeout seconds] ]
[ -consoleid id | -consolename name ]
[-level {trace|audit|info|warn|error}]
[-key keyname]

uwto
{ -help | -version }
```

Figure 34.1 Universal WTO - Command Line Syntax

CHAPTER 35

Universal Write-to-Operator Configuration Options

35.1 Overview

This chapter provides detailed information on the configuration options available for use with Universal Write-to-Operator (Universal WTO).

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

Information on how these options are used is documented in Chapter 34 Universal Write-to-Operator.

Section 1.3 Configuration Options Information provides a guideline for understanding the information presented for each option.

35.2 Configuration Options List

Table 35.1, below, identifies the Universal WTO configuration options.

Option Name	Description	Page
CONSOLE_ID	ID of the console to which the message is routed	485
CONSOLE_NAME	Name of the console to which the message is routed	486
HELP	Writes a description of the command options and their format	487
KEY	Key to associate with the message	488
MESSAGE	Text to write to the z/OS operator console	489
MESSAGE_LEVEL	Level of messages to write	490
REPLY	Directs UWTO is issue a WTOR message and wait for an operator reply to the message	491
TIMEOUT	Number of seconds to wait for a WTOR operator reply	492
VERSION	Writes the program version and copyright information	493

Table 35.1 Universal WTO Configuration Options

35.3 CONSOLE_ID

Description

The CONSOLE_ID option specifies the console ID which the message is routed.

If CONSOLE_ID specifies an invalid console ID, the message is written to the default consoles. Universal WTO will write a warning message to standard error and end with exit code 1.

CONSOLE_ID and CONSOLE_NAME are mutually exclusive.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-consoleid *id*					√
Environment Variable	UWTOCONSOLEID=*id*					√

Values

id is the ID of the console to which the messages is routed.

Valid values for *id* are numeric values.

35.4 CONSOLE_NAME

Description

The CONSOLE_NAME option specifies the console name which the message is routed.

It provides a method to route messages based on console name.

If an invalid console name is specified, the message is written to the default consoles. Universal WTO will write a warning message to standard error and end with exit code 1.

CONSOLE_NAME and CONSOLE_ID are mutually exclusive.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-consolename *name*					√
Environment Variable	UWTOCONSOLENAME=*name*					√

Values

name specifies the name of the console to which the message is routed.

The format of *name* is 1 to 8 characters.

35.5 HELP

Description

The HELP option writes a description of the command options and their format.

Usage

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

Values

(There are no values for this option.)

35.6 KEY

Description

The KEY option specifies a key to associate with the message.

The key provides a way to identify messages to operators. The z/OS DISPLAY console commands can list messages key values.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-key *keyname*					
Environment Variable	UWTOKEY=*keyname*					√

Values

keyname is the message key.

The format of *keyname* is 1 to 8 characters.

35.7 MESSAGE

Description

The MESSAGE option specifies the text to write to the z/OS operator console.

The text is written as a WTO or WTOR message, as specified by the REPLY option.

Note: Even though Universal WTO executes in the z/OS Unix System Services environment, not all USS supported characters are supported in the console character set. Refer to IBM MCS Console documentation for character set support. Unsupported characters are translated to blanks when written to the console.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-m *message*					√
Command Line, Long Form	-msg *message*					√
Environment Variable	UWTOMSG=*message*					√

Values

message is the text message to write to the z/OS operator console.

The maximum length of *message* depends on the type of message specified by REPLY:

- WTO 770 characters
- WTOR 121 characters

Note: A WTO message that exceeds 125 characters is written as a multi-line WTO.

35.8 MESSAGE_LEVEL

Description

The MESSAGE_LEVEL option specifies the level of messages to write.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l *level*					√
Command Line, Long Form	-level *level*					√
Environment Variable	UWTOLEVEL=*level*					√

Values

level indicates either of the following level of messages:

- **trace**
  Writes traces messages used for debugging. The trace file, named **`uwto.trc`**, is created in the working directory of Universal WTO.

  Note: Use **trace** only as directed by Stonebranch, Inc. Customer Support.
- **audit**
  Writes audit, informational, warning, and error messages.
- **info**
  Writes informational, warning, and error messages.
- **warn**
  Writes warning and error messages.
- **error**
  Writes error messages only.

[Default is warn.]

35.9 REPLY

Description

The REPLY option specifies the type of message to be issued by Universal WTO:

- WTO (do not request message reply)
- WTOR (request and wait for message reply)

For a WTOR message, the length of time to wait for a reply can be limited with the TIMEOUT option. The maximum reply length is 119 characters. The reply is written to Universal WTO's standard output file.

Note: A valid operator reply to a WTOR message can be zero characters. In this case, nothing is written to standard output.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-reply *option*					√
Environment Variable	UWTREPLY=*option*					√

Values

option specifies the type of message to be issued.

Valid values for option are:

- **no**
  Issue a WTO message.
- **yes**
  Issues a WTOR message.

[Default is no.]

35.10 TIMEOUT

Description

The TIMEOUT option specifies the number of seconds to wait for a reply to a WTOR message (see the REPLY option).

If a reply is not received within the specified time, the WTOR message is deleted and Universal WTO ends with exit code 2.

Usage

Method	Syntax	IBM i	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-timeout *seconds*					√
Environment Variable	UWTOTIMEOUT=*seconds*					√

Values

seconds is the length of time to wait for a reply.

[Default is *0* seconds (wait indefinitely for a reply.]

35.11 VERSION

Description

The VERSION option writes the program version and copyright information.

Usage

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

Values

(There are no values for this option.)

CHAPTER 36

Additional Information

36.1 Overview

This chapter provides additional information related to Stonebranch Solutions Utilities.

Table 36.1, below, identifies this information and provides a link to its location in this document.

Information	Description	Page
SSL cipher suites	SSL cipher suites for use with Universal Command.	495
Character Code Pages	Character code pages provided by Stonebranch Inc. for use with Stonebranch Solutions on each supported operating system.	496
UTT Files	Universal Translate Table (UTT) files are used to translate between Unicode and the local single-byte code page	498

Table 36.1 Stonebranch Solutions Utilities - Additional Information

36.2 SSL Cipher Suites

Table 36.2 identifies all of SSL cipher suites provided by Stonebranch, Inc. for use with Universal Command.

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.
NULL-SHA	No encryption and SHA-1 message digest.
NULL-MD5	No encryption and MD5 message digest.
NULL-NULL	No encryption, no data authentication, SSL is not used.

Table 36.2 SSL Cipher Suites

36.3 Character Code Pages

Table 36.3 identifies the character code pages provided by Stonebranch Inc. for use with Stonebranch Solutions on each supported operating system.

Code Page	CCSID	z/OS	UNIX	Windows	IBM i		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	IBM i		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 36.3 Character Code Pages

36.4 UTT Files

Table 36.4 identifies the Universal Translate Table (UTT) files that are used to translate between Unicode and the local single-byte code page.

Operating System	UTT File Location
IBM i	UTT files are located in the source physical file **UNVPRD420/UNVNLS**. *codepage* is the member name of the UTT file.
z/OS	UTT files are located in the library allocated to the **UNVNLS** ddname. *codepage* is the member name of the UTT file.
UNIX	UTT files are located in the **nls** subdirectory of the installation directory. *codepage* is the base file name of the UTT file. All UTT files end with an extension of **.utt**.
Windows	UTT files are located in the NLS subdirectory of the installation directory. *codepage* is the base file name of the UTT file. All UTT files end with an extension of **.utt**.
HP NonStop	UTT files are located in the **$SYSTEM.UNVNLS** subvolume. *codepage* is the base file name of the UTT file.

Table 36.4 UTT File Locations

APPENDIX A

Customer Support

Stonebranch, Inc. provides customer support, via telephone and e-mail, for Stonebranch Solutions Utilities and all Stonebranch Solutions components.

E-MAIL

All Locations

support@stonebranch.com

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

www.stonebranch.com

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

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