Universal Enterprise Controller

User Guide

Universal Products

Version 3.2.0

uec-user-3203

Universal Enterprise Controller

User Guide

Universal Products 3.2.0

Document Name	Universal Enterprise Controller 3.2.0 User Guide				
Document ID	uec-user-3203				
Products	**z/OS**	**UNIX**	**Windows**	**OS/400**	**HP NonStop**
Universal Enterprise Controller	√		√		
Universal Event Subsystem	√		√		
UECLoad	√		√		

Stonebranch Documentation Policy

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

SAP Certified Integration

Summary of Changes

Changes for Universal Enterprise Controller 3.2.0 User Guide (uec-user-3203)
November 2, 2009

Universal Products 3.2.0.9

- Removed information describing support of the zFS file system for Universal Enterprise Controller in:
  - Chapter 3 Universal Enterprise Controller
  - Chapter 7 UEC Database Administration

Changes for Universal Enterprise Controller 3.2.0 User Guide (uec-user-3202)
September 8, 2009

- Moved the UEC and UECLoad configuration options chapters to the following chapters in the new Universal Enterprise Controller 3.2.0 Reference Guide:
  - Chapter 2 Universal Enterprise Controller Configuration Options
  - Chapter 3 UECLoad Configuration Options
- Modified Chapter 3 Universal Enterprise Controller:
  - Added Section 3.4.6 Configuration Options.
  - Added Section 3.5.2 Stopping UEC.
  - Added Section 3.5.3 Configuration Options.
- Modified Chapter 5 UECLoad Utility:
  - Added Section 5.2.1 UECLoad for z/OS.
- Modified Chapter 7 UEC Database Administration:
  - Added Section 7.1.1 Database Files.
  - Added Section 7.1.2 Database Management, including:
    - Automated Database Cleanup

    - Memory Management
  - Modified Section 7.1.3 Database Recovery.

Universal Enterprise Controller 3.2.0.4

- Added information for the following configuration options in Chapter 3 Universal Enterprise Controller:
  - SAP_POLLING_INTERVAL
  - TMP_DIRECTORY
  - UNIX_DB_DATA_SET
- Specified that UNVDB ddname is not used with zFS data sets in Table 3.1 Universal Enterprise Controller for z/OS – DD Statements in JCL Procedure.
- Modified Section 3.3.2 SAP System Polling.

Changes for Universal Enterprise Controller 3.2.0 User Guide (uec-user-3201) September 5, 2008

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

Universal Activity Monitor 3.2.0.1

- Specified the addition of job views (for Universal Command and Universal Data Mover) and file views for the Universal Activity Monitor.

Changes for Universal Enterprise Controller 3.2.0 User Guide (uec-user-320) May 16, 2008

Universal Activity Monitor 3.2.0.0

- Added support for the following features:
  - SSL and Certificates
    UEC now offers SSL configuration for Agent communication.
  - Report Agent Status at UEC Startup
    UEC can now issue alerts reporting the status of all Agents when UEC first starts in order to synchronize Agent status with automation and scheduling systems.
  - UECLOAD Improvements
    UECLOAD has been enhanced to work with almost all UEC database tables and fields. UEC databases can be loaded, updated and exported with the UECLOAD batch and command line program.
  - Individual Agent Polling Interval
    The interval on which UEC polls for the health and status of an Agent can be set in the Agent definition as opposed to a global system setting. Production Agents can be polled more frequently and lower priority Agents less frequently.

- Universal Activity Monitor
  Universal Activity Monitor (UAM) provides a graphical user interface into a near real-time view of the operational state of the Universal Product infrastructure, workload, and alerts. The infrastructure views show the operational status of the Universal Product infrastructure. The workload views show the status of job execution and file transfers. The alert views show exception conditions that may impact production workload.
- Universal Event Subsystem
  Universal Event Subsystem (UES) provides a continuous, near real-time record of all Universal Product events relating to workload and infrastructure activity, status and exceptions. UES data may be archived in long-term storage to provide an historical record of workload related activity for auditing, reporting and diagnostic purposes.
- Universal Management Console
  Universal Management Console (UMC) provides for the administration of Universal Product component configurations that are installed throughout the enterprise. UMC can manage the update of a single product configuration file on a remote Agent or the update of all of product configuration files on thousands of Agents. All configuration changes are logged for audit and reporting purposes.

  Universal Product configurations can be placed in managed mode. A managed configuration can only be updated with UMC. No configuration changes can be performed on the host on which the components are installed.

- Added the following configuration options in Chapter 2 Universal Enterprise Controller Configuration Options:
  - CERTIFICATE_REVOCATION_LIST
  - COMM_SESSIONS_PER_THREAD
  - COMM_THREADS
  - COMMIT_COMPLETE_EXPIRATION
  - COMMIT_INCOMPLETE_EXPIRATION
  - CONVERT
  - DELETE_EVENTS_ON_BROKER
  - DNS_POLLING_INTERVAL
  - HOSTNAME_RETRY_COUNT
  - JOB_THREADS
  - KEEP_MONITOR_EVENTS
  - LOG_MESSAGES_DIRECTORY
  - MONITOR_EVENT_EXPIRATION
  - MOUNT_POINT
  - MOUNT_POINT_MODE
  - PERSISTENT_EVENT_EXPIRATION
  - SAF_KEY_RING
  - SAF_KEY_RING_LABEL
  - SSL_IMPLEMENTATION
  - TRACE_DIRECTORY
  - TRACE_TABLE

- Modified the following configuration option in Chapter 2 Universal Enterprise Controller Configuration Options:
  - Added Clean-Up Routines for PERSISTENT_EVENT_EXPIRATION
- Added Section 2.3 Universal Configuration Manager.
- Added the following configuration options in Chapter 3 UECLoad Configuration Options:
  - ARCFILE
  - END_TIME
  - EXPORT_DELETE
  - FORMAT
  - START_TIME
- Added the following examples in Chapter 5 UECLoad Utility:
    - Export Events into ARC Format (z/OS)
    - Retrieve Archived File and Export into XML (z/OS)
- Added the following examples in Section 5.3 Examples of UECLoad:
  - Export Events
  - Retrieve Archived File and Export
- Added Chapter 7 UEC Database Administration.
- Added Configuration File Keyword as a specification method for Windows options.

Changes for Universal Enterprise Controller 3.1.1 User Guide (uec-user- 31111) February 28, 2007

- Added List of Figures and List of Tables.
- Added Appendix A Customer Support.

Universal Enterprise Controller 3.1.1.5

- Added a note regarding high dispatch priority to Section 2.2.4 Starting and Stopping the Controller of Section 2.2 z/OS in Chapter 2 Configuration.

Changes for 3.1.1 Release November 8, 2005

- Ability to set trace_directory config option in Windows.
- Introduction of independent broker polling interval.

Changes for 3.1.1 Release
April 30, 2005

- Major performance enhancements to reduce CPU utilization and increase Universal Broker Monitor responsiveness.

Changes for 3.1.0 Release
October 31, 2004

- Added DNS cache timeout value.
- Added OS authentication.
- Added new broker query configuration parameters.
- Modified the default trace lines so that it's now 500000.
- Changed the default message level to WARN for z/OS.
- Added the SSL Protocol configuration parameters ssl_cipher_list, certificate, ca_certificates, private_key, private_key_password.
- Removed the Drawing problem section and changed Browser and Java Plugin requirement to 1.4.2_01
- Added sections on the use of the UECLoad utility program.

Changes for 1.1.0 Release
March 19, 2004

- Corrected the ACL remote user parameter syntax.
- Added Stoneman tip for ACL definitions for U-Control.
- Changed the section "Browser and Java Plugin Requirements" tested java plug-in from 1.3.1_06 to 1.3.1_07, which is the release of the SUN JVM that fixed the drawing and font problems inherent in 1.3.1_06 and older.
- Added the update_interval configuration parameter to the UEC Runtime Options and Command Option Reference section.

Documentation Update
November 24, 2003

- Moved installation instructions to a separate document, the Universal Products Installation Guide. Installation and configuration instructions were previously contained in this document in a single chapter. That chapter now contains information for configuring the application only.

Changes for 1.1.0 Release
September 3, 2003

- The Stopping a Component section for the Broker Monitor has been updated.

Changes for 1.1.0 Release
February 24, 2003

- The overview section has been updated to provide a clearer understanding of how UEC works and interacts with the Broker Monitor and UEC Administration applications.
- The section on enabling the Broker Monitor and UEC Administration to be accessed from the web has been updated.
- The UEC Administration overview has been updated to reflect the changes since the last release.
- A section has been added on using the Broker Monitor.
- A troubleshooting section has been added.
- Additional information added to the Windows installation chapter.

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.

Format

Starting with the Universal Products 3.2.0 release, the Universal Enterprise Controller User Guide has been reformatted and restructured.

Most importantly, links to information in a new companion document, Universal Enterprise Controller 3.2.0 Client Applications, have been created in this user guide.

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

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

Conventions

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

Typeface and Fonts

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

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

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

Stoneman's Tip

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

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.
- **UNUX** is synonymous with operating systems based on AT&T and BSD origins and the Linux operating system.

These names do not imply software support in any manner. Refer to the Universal Products 3.2.0 Installation Guide for a detailed list of supported operating systems.

Document Organization

The document is organized into the following chapters:

- Universal Enterprise Controller Overview (Chapter 1)
  Overview of the information contained in this document.
- UEC Features (Chapter 2)
  Description of Universal Enterprise Controller features.
- Universal Enterprise Controller (Chapter 3)
  Information about configuring Universal Enterprise Controller.
- Universal Event Subsystem (Chapter 4)
  Information about the Universal Event Subsystem of Universal Enterprise Controller.
- UECLoad Utility (Chapter 5)
  Information about the UECLoad utility of the Universal Enterprise Controller
- Troubleshooting (Chapter 6)
  Information about troubleshooting Universal Enterprise Controller.
- UEC Database Administration (Chapter 7)
  Information about Universal Enterprise Controller 3.2.0 database administration.
- Customer Support (Appendix A)
  Customer support contact information for Universal Enterprise Controller.

Chapter 1
Universal Enterprise Controller Overview

1.1 Overview

Universal Enterprise Controller (UEC) is a Universal Products server application, for Windows and z/OS operating systems, that monitors the status of all Universal Agent installations in your enterprise.

(A Universal Agent is a single Universal Products installation comprised of one Universal Broker and one or more Universal Products.)

UEC sends out alerts to any connected agent-monitoring applications whenever:

- Universal Broker is unreachable.
- Universal Broker is not responding.
- Universal Agent component enters an orphaned or disconnected state.

These alerts are posted to the:

- Event Log (when running under Windows)
- Console (when running under z/OS)

Automation tools can be used in conjunction with these messages to perform operations based on agent failures.

1.2 Universal Enterprise Controller System

Figure 1.1, below, illustrates the Universal Enterprise Controller system.

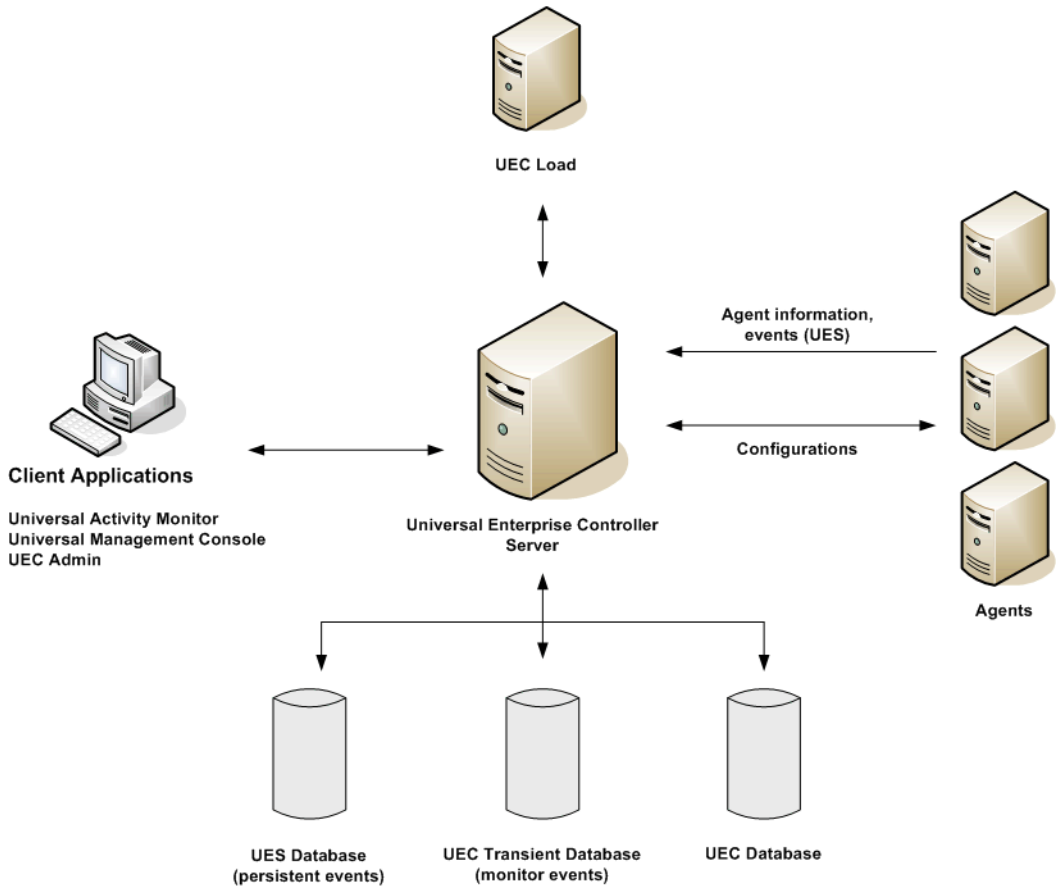

Figure 1.1 Universal Enterprise Controller - System

See Chapter 3 Universal Enterprise Controller for detailed information on executing and configuring UEC.

1.3 Additional UEC Functionality

As illustrated in Figure 1.1, UEC also provides the following additional functionality:

- Universal Event Subsystem
- UECLoad Utility
- UEC Client Applications (for Windows)

1.3.1 Universal Event Subsystem

The Universal Event Subsystem (UES) is a subsystem of Universal Enterprise Controller that records, routes, and manages event messages generated by Universal Product components.

1.3.2 UECLoad Utility

The UECLoad utility permits UEC users to add, delete, and view agents in the UEC database.

Via UECLoad, a user can add or delete individual agents or supply a Universal Agents definition file (**`deffile`**) with definitions to be added or deleted from UEC.

UECLoad also can be used to export existing agent definitions (which later can be used as a definition file to re-create the agent definitions) and event records (from UES).

1.3.3 UEC Client Applications (for Windows)

Under the Windows operating system, UEC connects to three client applications:

1. UEC Administration
2. Universal Activity Monitor
3. Universal Management Console

These client applications are run as stand-alone applications.

See the Universal Enterprise Controller 3.2.0 Client Applications guide for detailed information on running these applications.

UEC Administration

The UEC Administration utility is used to administer the list of Universal Agents that UEC will monitor. It also is used to administer UEC users and their permissions. With UEC Administration, the user can add, modify, and delete users, agents, groups, and SAP systems.

Note: A user must have UEC administrative rights – granted via UEC Administration – in order to use UEC Administration.

Upon installation of UEC, a default user ID (**admin**) and password (**admin**) are created having UEC administrative rights. It is recommended that you create another user with UEC administrative rights and then delete the default user.

Universal Activity Monitor

The Universal Activity Monitor (UAM) connects to UEC. It displays information about the current status, posted alerts, and job (for Universal Command and Universal Data Mover) and file activity for all Universal Agents being monitored by UEC throughout an enterprise.

When a Universal Agent or SAP system is added to the Universal Enterprise Controller (UEC), via the UEC Administration application, UEC is able to collect information about that agent or SAP system.

Authorized users are able to use the UAM interface to stop running any Universal Products component (if it is a component of a Universal Agent being polled by UEC).

Universal Management Console

The Universal Management Console (UMC) provides a graphical user interface for reconfiguring Universal Agents.

UMC provides two important features for this reconfiguration:

1. Reconfigure agents remotely, from a single machine.
2. Reconfigure multiple agents simultaneously.

Chapter 2
UEC Features

2.1 Overview

This chapter provides information on Universal Enterprise Controller features that apply to all operating systems.

- Configuration
- Universal Configuration Manager
- Network Data Transmission
- Message and Audit Facilities
- X.509 Certificates

2.2 Configuration

Product configuration consists of specifying options that control product behavior and resource allocation.

- An example of configurable product behavior is whether or not data transferred over the network is compressed.
- An example of configurable resource allocation is the directory location in which the product creates its log files.

Configuration can be done either by:

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

Each option is comprised of a pre-defined parameter, which identifies the option, and one or more values. The format of the parameter depends on the method being used to specify the option.

Although there are many configurable product options, Universal Products, in general, are designed to require minimal configuration and administration. The default options will work very well in most environments. When local requirements do require a change in product configuration, there are multiple methods available to configure the products in order to meet your needs.

2.2.1 Configuration Sources

Configuration options are read from the following sources:

1. PARM keyword of the started procedure (z/OS only)
2. Universal Enterprise Controller configuration file

The order of precedence is the same as the list above; with PARM options overriding values stored in the configuration file.

z/OS

Configuration files are members of a PDSE. The data set record format is fixed or fixed block with a record length of 80. No line numbers can exist in columns 72-80. All 80 columns are processed as data.

All configuration files are installed in the **UNVCONF** library.

The UEC configuration file is in the SMP/E target data set **UNVCONF** as member **UECCFG00**.

Windows

Although configuration files can be edited with any text editor (for example, Notepad), the Universal Configuration Manager application 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 (see Section 2.3 Universal Configuration Manager).

2.2.2 Configuration File Syntax

Configuration files are text files that can be edited with any available text editor.

The following rules apply for configuration file syntax:

- Options are specified in a keyword / value format.
- Keywords can start in any column.
- Keywords must be separated from values by at least one space or tab character.
- Keywords are not case sensitive.
- Keywords cannot contain spaces or tabs.
- Values can contain spaces and tabs, but if they do, they must be enclosed in single (**'**) or double (**"**) quotation marks. Repeat the enclosing characters to include them as part of the value.
- Values case sensitivity depends on the value being specified. For example:
  - Directory and file names are case sensitive.
  - Pre-defined values (such as **yes** and **no**) are not case sensitive.
- Each keyword / value pair must be on one line.
- Characters after the value are ignored.
- Newline characters are not permitted in a value.
- Values can be continued from one line to the next either by ending the line with a:
  - Plus (**+**) character, to remove all intervening spaces.
  - Minus (**-**) character, to preserve all intervening spaces between the end of the line being continued and the beginning of the continuing line.

  Ensure that the line continuation character is the last character on a line.
- Comment lines start with a hash (**#**) character.
- Blank lines are ignored.

Note: If an option is specified more than once in a configuration file, the last option specified is used.

2.2.3 Configuration Options

See the Universal Enterprise Controller 3.2.0 Reference Guide for detailed information on all UEC configuration options.

2.3 Universal Configuration Manager

The Universal Configuration Manager is a Universal Products graphical user interface application that enables you to configure all of the Universal Products that have been installed on a Windows operating system.

It is the recommended method of specifying configuration data that will not change with each command invocation. Universal Configuration Manager helps protect the integrity of the configuration file by validating all changes to configuration option values.

The configuration data for a Universal Products for Windows system is stored in the configuration file.

2.3.1 Availability

Universal Configuration Manager is installed automatically on the Windows operating system as part of every Universal Enterprise Controller for Windows installation.

It is available to all user accounts in the Windows Administrator group.

Windows Vista

When opening the Universal Configuration Manager for the first time on Windows Vista, two new operating system features, the Program Compatibility Assistant (PCA) and User Account Control (UAC), may affect its behavior.

With these two features enabled, the expected Universal Configuration Manager behavior is as follows:

1. Universal Configuration Manager may issue the following error:

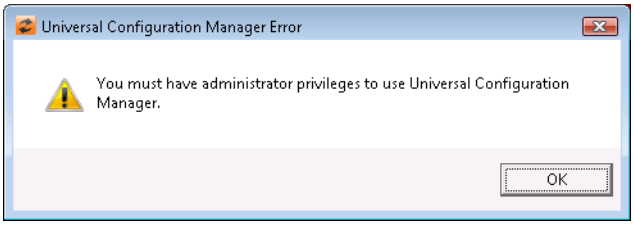

Figure 2.1 Universal Configuration Manager Error dialog - Windows Vista

2. Click **OK** to dismiss the error message.

   The Windows Vista Program Compatibility Assistant (PCA) displays the following dialog:

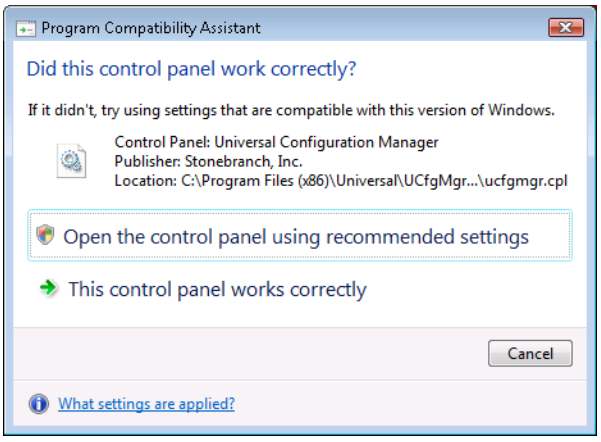

Figure 2.2 Windows Vista - Program Compatibility Assistant

3. To continue, select **`Open the control panel using recommended settings`**. This instructs the PCA to "shim" (Microsoft term) the Configuration Manager, establishing it as an application that requires elevated privileges.

   Windows Vista User Account Control (UAC) then displays a prompt seeking permission to elevate the logged-in account's access token.

4. Select **`Continue`** to give the account full administrative privileges.

   Subsequent attempts to open Universal Configuration Manager should result only in the UAC prompt.

2.3.2 Accessing the Universal Configuration Manager

To access the Universal Configuration Manager:

1. Click the **Start** icon at the lower left corner of your Windows operating system screen to display the Start menu.
2. Click (Settings/) Control Panel on the Start menu to display the Control Panel screen.
3. Select the Universal Configuration Manager icon to display the Universal Configuration Manager screen (see Figure 2.3).

Windows XP, Windows Vista, Windows Server 2008

Newer versions of Windows support a Control Panel view that places applet icons within categories. This "category view" may affect the location of the Universal Configuration Manager icon.

For example, the Windows XP Category View places the Universal Configuration Manager icon under the **Other Control Panel Options** link. Windows Vista and Windows Server 2008 place the icon within the **Additional Options** category.

If you have trouble locating the Universal Configuration Manager icon, simply switch to the Classic View to display all Control Panel icons at the same time.

64-bit Windows Editions

The Windows Control Panel places icons for all 32-bit applets under the **View x86 Control Panel Icons** (or, on newer versions, the **View 32-bit Control Panel Icons**) category, even when the Classic View is enabled.

When using the Category View, look for the 32-bit Control Panel applet icons in the **Additional Options** category.

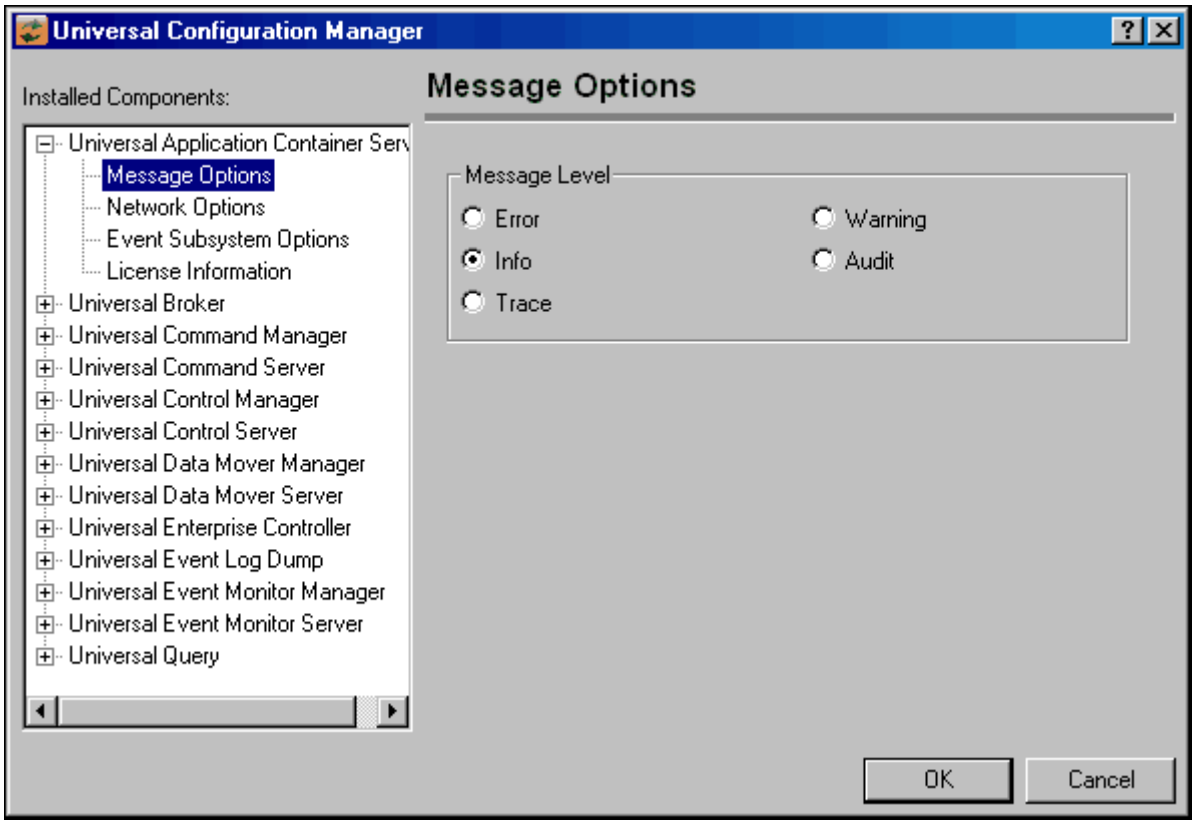

Figure 2.3 Universal Configuration Manager

Each Universal Configuration Manager screen contains two sections:

1. Left side of the screen displays the Installed Components tree, which lists:
   - Universal Products components currently installed on your system.
   - Property pages available for each component (as selected), which include one or more of the following:
     - Configuration options
     - Access control lists
     - Licensing information
     - Other component-specific information
2. Right side of the screen displays information for the selected component / page.

(By default, Universal Configuration Manager displays the first property page of the first component in the Installed Components tree.)

2.3.3 Navigating through Universal Configuration Manager

To display general information about a component, click the component name in the Installed Components list.

To display the list of property pages for a component, click the **+** icon next to the component name in the Installed Components list.

To display a property page, click the name of that page in the Installed Components list.

If a property page has one or more of its own pages, a **+** icon displays next to the name of that property page in the Installed Components list. Click that **+** icon to display a list of those pages.

In Figure 2.3, for example:

- List of property pages is displayed for Universal Broker.
- Message Options property page has been selected, and information for that property is displayed on the right side of the page.
- No **+** icons next to any of the property pages indicates that they do not have one or more of their own property pages.

2.3.4 Modifying / Entering Data

On the property pages, modify / enter data by clicking radio buttons, selecting from drop-down lists, and/or typing in data entry fields.

Some property pages provide panels that you must click in order to:

- Modify or adjust the displayed information.
- Display additional, modifiable information.

Note: You do not have to click the **OK** button after every modification or entry, or on every property page on which you have modified and/or entered data. Clicking **OK** just once, on any page, will save the modifications and entries made on all pages – and will exit Universal Configuration Manager (see Section 2.3.5 Saving Data.)

Rules for Modifying / Entering Data

The following rules apply for the modification and entry of data:

- Quotation marks are not required for configuration values that contain spaces.
- Edit controls (used to input free-form text values) handle conversion of any case sensitive configuration values. Except where specifically noted, values entered in all other edit controls are case insensitive.

2.3.5 Saving Data

To save all of the modifications / entries made on all of the property pages, click the **OK** button at the bottom of any property page. The information is saved in the configuration file, and Universal Broker is automatically refreshed.

Clicking the **OK** button also exits the Universal Configuration Manager. (If you click **OK** after every modification, you will have to re-access Universal Configuration Manager.)

To exit Universal Configuration Manager without saving any of the modifications / entries made on all property pages, click the **Cancel** button.

2.3.6 Accessing Help Information

Universal Configuration Manager provides context-sensitive help information for the fields and panels on every Universal Products component options screen.

To access Help:

1. Click the question mark (**?**) icon at the top right of the screen.
2. Move the cursor (now accompanied by the **?**) to the field or panel for which you want help.
3. Click the field or panel to display Help text.
4. To remove the displayed Help text, click anywhere on the screen.

Windows Vista, Windows Server 2008

The Universal Configuration Manager's context-sensitive help is a WinHelp file, which Windows Vista and Windows Server 2008 does not support.

Microsoft offers the 32-bit WinHelp engine as a separate download from its website. If you require access to the Universal Configuration Manager's context-sensitive help, simply download and install the WinHelp engine.

2.3.7 Universal Enterprise Controller Component

Figure 2.4 illustrates the Universal Configuration Manager screen for the Universal Enterprise Controller.

The Installed Components list identifies all of the UEC property pages.

The text describes the selected component, Universal Enterprise Controller.

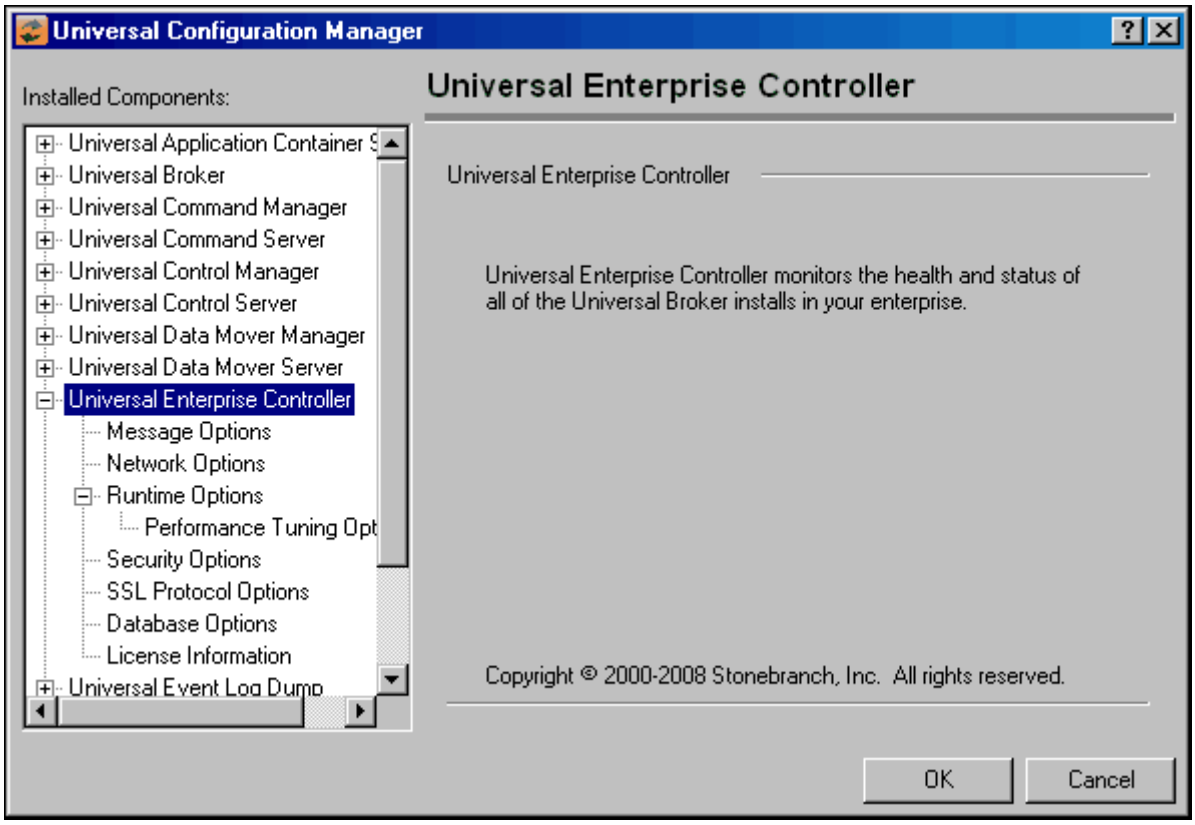

Figure 2.4 Universal Configuration Manager - Universal Enterprise Controller

2.4 Network Data Transmission

Distributed systems, such as Universal Enterprise Controller, communicate over data networks. All Stonebranch products communicate using the TCP/IP protocol. The UDP protocol is not used for any product data communication over a network.

The Universal Products suite can utilize one of two network protocols:

1. Secure Socket Layer version 3 (SSLv3) provides the highest level of security available. SSL is a widely used and accepted network protocol for distributed software applications that are required to address all aspects of secure data transfer on private and public networks.
2. Universal Products version 2 (UNVv2) legacy protocol is provided for backward compatibility with previous versions of Universal Products.

The following sections discuss each of the protocols.

In addition to the network protocol used to transmit data, Universal Products application protocol is discussed as well.

2.4.1 Secure Socket Layer Protocol

Universal Products implement the SSL protocol using the OpenSSL library or the IBM z/OS System SSL library, available on the z/OS operating system. The most recent SSL standard is version3. A subsequent version was produced changing the name to Transport Layer Security version 1 (TLSv1). TLSv1 is the actual protocol used by Universal Products. TLSv1 is more commonly referred to simply as SSL and the term SSL is used throughout the rest of this documentation to mean TLSv1 unless otherwise noted.

The SSL protocol addresses the major challenges of communicating securely over a potentially insecure data network. The following sections discuss the issue of data privacy and integrity, and peer authentication.

Data Privacy and Integrity

People with sufficient technical knowledge and access to network resources can watch or capture data transmitting across the network. What they do with the data is up to them.

Data sent over the network that should remain private must be encrypted in a manner that unauthorized persons cannot determine what the original data contained regardless of their level of expertise, access to network resources, amount of data captured, and amount of time they have. The only party that should be able to read the data is the intended recipient.

As data is transmitted over the network, it passes through media and hardware of unknown quality that may erroneously change bits of data without warning. Additionally, although data may be encrypted, there is nothing stopping a malicious person from changing the data while it is transmitted over the network. The changed data may or may not be detected by the recipient depending on what changed and how it is processed. It may be accepted as valid data, but the information it represents is now erroneous

Data integrity must be protected from errors in transmission and malicious users. Data integrity checks insures that what was sent is exactly what is received by the recipient. Without integrity checks, there is no guarantee.

Encryption algorithms are used to encrypt data into an unreadable format. The encryption process is computationally expensive. There are a variety of encryption algorithms some of which perform better than others. Some algorithms offer a higher level of security than others. Typically, the higher level of security requires more computational resources.

Message digest algorithms are used to produce a Message Authentication Code (MAC) that uniquely identifies a block of data. The sender computes a MAC for the data being sent based on a shared secret key the sender and receiver hold. The sender sends the data and the MAC to the receiver. The receiver computes a new MAC for the received data based on the shared secret key. If the two MAC's are the same, data integrity is maintained, else the data is rejected as it has been modified. Message digest algorithms are often referred to as MAC's and can be used synonymously in most contexts.

The SSL standard defines a set of encryption and message digest algorithms referred to cipher suites that insure data privacy and data integrity. Cipher suites pair encryption algorithms with appropriate message digest algorithms. The two algorithms cannot be specified individually.

Universal Products supports a subset of the complete SSL cipher suites defined by the standard. The cipher suite name is formatted as an encryption algorithm abbreviation followed by the message digest algorithm abbreviation.

Table 2.1, below, identifies the supported cipher suites.

Cipher Suite Name	Description
RC4-SHA	128-bit RC4 encryption with SHA-1 message digest
RC4-MD5	128-bit RC4 encryption with MD5 message digest
AES256-SHA	256-bit AES encryption with SHA-1 message digest
AES128-SHA	128-bit AES encryption with SHA-1 message digest
DES-CBC3-SHA	128-bit Triple-DES encryption with SHA-1 message digest
DES-CBC-SHA	128-bit DES encryption with SHA-1 message digest
NULL-SHA	No encryption with SHA-1 message digest
NULL-MD5	No encryption with MD5 message digest

Table 2.1 Supported SSL cipher suites

Universal Products support one additional cipher suite name that is not part of the SSL protocol. The NULL-NULL cipher suite turns SSL off completely and instead uses the Universal Products Protocol (UNVv2) described below.

Peer Authentication

When communicating with a party across a data network, how do you insure that the party you are communicating with (your peer) is who you believe? A common form of network attack is a malicious user representing themselves as another user or host.

Peer authentication insures that the peer is truly who they identify themselves as. Peer authentication applies to users, computer programs and hardware systems.

SSL uses X.509 certificates and public and private keys to identify an entity. An entity may be a person, a program, or a system. A complete description of X.509 certificates is beyond the scope of this documentation. Section 2.6 X.509 Certificates provides an overview to help get the reader oriented to the concepts, terminology and benefits.

For additional details, the following website is recommended:

http://www.faqs.org/rfcs/rfc3280.html

2.4.2 Universal Products Protocol

The Universal Products protocol (UNVv2) is a proprietary protocol that securely and efficiently transports data across data networks. UNVv2 is used in Universal Products prior to version 3 and will be available in future versions.

UNVv2 addresses data privacy and integrity. It does not address peer authentication.

Data Privacy and Integrity

Data privacy is insured with data encryption algorithms. UNVv2 utilizes 128-bit RC4 encryption for all data encryption.

Data integrity is insured with message digest algorithms. UNVv2 utilizes 128-bit MD5 MAC's for data integrity. UNVv2 referred to data integrity as data authentication.

Encryption and integrity may be enabled and disabled on an individual bases.

Encryption keys are generated using a proprietary key agreement algorithm. A new key is created for each and every network session.

2.4.3 Universal Products Application Protocol

Universal Product components use an application-layer protocol to exchange data messages. The protocol has the following characteristics:

- Low-Overhead
- Secure
- Extensible
- Configurable Options

The following sections refer to two categories of data transmitted by Universal Products:

- Control data (or messages) consists of messages generated by Universal Products components in order to communicate with each other. The user of the product has no access to the control data itself.
- Application data (or messages) consists of data that is transmitted as part of the requested work being executed. For example, standard input and output data of jobs Universal Command executes. The data is created by the job and read or written by Universal Command on behalf of the job.

Low-Overhead

The protocol is lightweight, in order to minimize its use of network bandwidth. The product provides application data compression options, which reduces the amount of network data even further.

There are two possible compression methods:

- **`ZLIB`** method offers the highest compression ratios with highest CPU utilization.
- **`HASP`** method offers the lowest compression ratios with lowest CPU utilization.

Note: Control data is not compressed. Compression options are available for application data only.

Secure

The protocol is secure. All control data exchanged between Universal Products components are encrypted with a unique session key and contain a MAC. The encryption prevents anyone from analyzing the message data and attempting to circumvent product and customer policies. Each session uses a different encryption key to prevent "play back" types of network attacks, where messages captured from a previous session are replayed in a new session. This applies to both network protocols: SSL and UNVv2.

The security features used in the control messages are not optional. They cannot be turned off. The security features are optional for application data sent over the network.

The data encryption options affect the application data being sent over the network. Special fields, such as passwords, are always encrypted. The encryption option cannot be turned off for such data.

Extensible

The message protocol used between the Universal Products components is extensible. New message fields can be added with each new release without creating product component incompatibilities. This permits different component versions to communication with each other with no problems. This is a very important feature for distributed systems, since it is near impossible to upgrade hundreds of servers simultaneously.

New encryption and compression algorithms can be added in future releases without loosing backward compatibility with older releases. After a network connection is made, connection options are negotiated between the two Universal Products programs. The options negotiated include which encryption and compression algorithms are used for the session. Only algorithms that both programs implement are chosen in the negotiation process. The negotiation process permits two different program versions to communicate.

2.4.4 Configurable Options

The network protocol can be configured in ways that affect compress, encryption, code pages, and network delays.

The following configuration options are available on many of the Universal Products:

CODE_PAGE

The CODE_PAGE option specifies the code page translation table used to translate network data from and to the local code page for the system on which the program is executing.

A codepage table is text file that contain a two-column table. The table maps local single byte character codes to two-byte UNICODE character codes.

Code pages are located in the product National Language Support (NLS) directory or library. New code pages may be created and added to the NLS directory or library. The CODE_PAGE option value is simply the name of the code page file without any file name extension if present.

CTL_SSL_CIPHER_LIST

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.

The SSL protocol uses cipher suites to specify the combination of encryption and message digest algorithms used for a session. An ordered list of acceptable cipher suites can be specified in a most to least order of preference.

An example cipher suite list is RC4-MD5,RC4-SHA,AES128-SHA. The RC4-MD5 cipher suite is the most preferred and AES128-SHA is the least preferred.

When a manager and server first connect, they perform an SSL handshake. The handshake negotiates the cipher suite used for the session. The manager and server each have a cipher suite list and the first one in common is used for the session.

Why is a list of cipher suites helpful? A distributed software solution may cross many organizational and application boundaries each with their own security requirements. Instead of having to choose one cipher suite for all distributed components, the software components can be configured with their own list of acceptable cipher suites based on their local security requirements. When a high level of security is required, the higher CPU consuming cipher suite is justified. When lower level of security is acceptable, a lower CPU consuming cipher suite may be used. As long as the manager has both cipher suites in its list, it can negotiate either cipher suite with servers of different security levels.

DATA_AUTHENTICATION

The DATA_AUTHENTICATION option specifies whether or not the network data is authenticated. Data authentication verifies that the data did not change from the point it was sent to the point it was received.

Data authentication also is referred to as a data integrity in this document.

Data authentication occurs for each message sent over the network. If a message fails authentication, the network session is terminated and both programs end with an error.

The DATA_AUTHENTICATION option is applicable to the UNVv2 protocol only. SSL always performs authentication.

DATA_COMPRESSION

The DATA_COMPRESS option specifies that network data be compressed.

Compression attempts to reduce the amount of data to a form that can be decompressed to its original form. The compression ratio is the original size divided by the compressed size. The compression ratio value will depend on the type of data. Some data compress better than others.

Two methods of compression are available:

- ZLIB method provides the highest compression ratio with the highest use of CPU
- HASP method provides the lowest compress ratio with the lowest use of CPU.

Whether or not compression is used and which compression method is used depends on several items:

- Network bandwidth. If network bandwidth is small, compression may be worth the cost in CPU.
- CPU resources. If CPU is limited, the CPU cost may not be worth the reduced bandwidth usage.
- Data compression ratio. If the data does not compress well, it is probably not worth CPU cost. If the data ratio is high, the CPU cost may be worth it.

DATA_ENCRYPTION

The DATA_ENCRYPTION option specifies whether or not network data is encrypted.

Encryption translates data into a format that prevents the original data from being determined. Decryption translates encrypted data back into its original form.

The type of encryption performed depends on the network protocol being used, SSL or UNVv2.

Data encryption does increase CPU usage. Whether or not encryption is used depends on the sensitivity of the data and the security of the two host systems and the data network between the hosts.

DATA_SSL_CIPHER_LIST

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

(See CTL_SSL_CIPHER_LIST in this section.)

DEFAULT_CIPHER

The DEFAULT_CIPHER option specifies the SSL cipher suite to use (since SSL protocol requires a cipher suite) if the DATA_ENCRYPTION option is set to NO. The default DEFAULT_CIPHER is NULL-MD5 (no encryption, MD5 message digest).

All SSL cipher suites have a message digest for good reasons. The message digest ensures that the data sent are the data received. Without a message digest, it is possible for bits of the data packet to get changed without being noticed.

KEEPALIVE_INTERVAL

The KEEPALIVE_INTERVAL option specifies how often, in seconds, a keepalive message (also commonly known as a heartbeat message) is sent between a manager and server. A keepalive message ensures that the network and both programs are operating normally. Without a keepalive message, error conditions can arise that place one or both programs in an infinite wait.

A keepalive message is sent from the server to the manager. If the server does not receive a keepalive acknowledgement from the manager in a certain period of time (calculated as the maximum of 2 x NETWORK_DELAY or the KEEPALIVE_INTERVAL), the server considers the manager or network as unusable. How the server processes a keepalive time-out depends on what fault tolerant features are being used. If no fault tolerant features are being used, the server ends with an error. The manager expects to receive a keepalive message in a certain period of time (calculated as the KEEPALIVE_INTERVAL + 2 x NETWORK_DELAY).

NETWORK_DELAY

The NETWORK_DELAY option provides the ability to fine tune Universal product'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, network bandwidth, and the network media type. 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 Products time out waiting for a packet to arrive in a specified period of time. The delay option specifies this period of time.

NETWORK_DELAY specifies the maximum acceptable delay in transmitting data between two programs. Should a data transmission take longer than the specified delay, the operation ends with a time out error. Universal Products will consider a time out error as a network fault.

The default NETWORK_DELAY value is 120 seconds. This value is reasonable for most networks and operational characteristics. If the value is too small, false network time outs could occur. If the value is too large, programs will wait a long period of time before reporting a time out problem.

SIO_MODE

The SIO_MODE option specifies whether the data transmitted over the network is processed as text data or binary data.

Text data is translated between the remote and local code pages. Additionally, end of line representations are converted

Text translation operates in two modes: direct and UCS. The default is direct. The direct translation mode exchanges code pages between Universal Products components to build direct translation tables. Direct translation is the fastest translation method when a significant amount (greater then 10K) of text data is transmitted. The code page exchange increases the amount of data sent over the network as part of the network connection negotiation. UCS translation does not require the exchange of code pages. For transactions that have little text data transmission, this is the fastest.

Binary data is transmitted without any data translation.

2.5 Message and Audit Facilities

All Universal Products have the same message facilities. Messages - in this context - are text messages written to a console, file, or system log that:

1. Document the actions taken by a program.
2. Inform users of error conditions encountered by a program.

This section describes the message and audit facilities that are common to all Universal Products. (See the individual Universal Product documentation for additional details.)

2.5.1 Message Types

There are six types (or severity levels) of Universal Products messages. (The severity level is based on the type of information provided by those messages.)

1. Audit messages document the configuration options used by the program's execution and resource allocation details. They provide complete description of the program execution for auditing and problem resolution.
2. Informational messages document the actions being taken by a program. They help determine the current stage of processing for a program. Informational messages also document statistics about data processed.
3. Warning messages document unexpected behavior that may cause or indicate a problem.
4. Error messages document program errors. They provide diagnostic data to help identify the cause of the problem.
5. Diagnostic messages document diagnostic information for problem resolution.
6. Alert messages document a notification that a communications issue, which does not disrupt the program or require action, has occurred.

The MESSAGE_LEVEL configuration option in each Universal Product component lets you specify which messages are written (see Section 2.5.3 Message Levels).

2.5.2 Message ID

Each message is prefixed with a message ID that identifies the message.

The message ID format is **UNVnnnnI**, where:

- **nnnn** is the message number.
- **I** is the message severity level:
  - **A** (Audit)
  - **I** (Informational)
  - **W** (Warning)
  - **E** (Error)
  - **T** (alerT)
  - **D** (Diagnostic)

Note: The Universal Products 3.2.0 Messages and Codes document identifies all messages numerically, by product, using the **nnnn** message number.

2.5.3 Message Levels

Each Universal Product includes a MESSAGE_LEVEL configuration option that lets you select which levels (that is, severity levels) of messages are to be written.

- *Audit* specifies that all audit, informational, warning, and error messages are to be written.
- *Informational* specifies that all informational, warning, and error messages are to be written.
- *Warning* specifies that all warning and error messages are to be written.
- *Error* specifies that all error messages are to be written.
- *Trace* specifies that a trace file is created, to which data used for program analysis will be written. The trace file name and location are Universal Product dependent (see the appropriate Universal Product documentation for details).

  (Trace should be used only at the request of Stonebranch, Inc. Customer Support.)

Note: Diagnostic and Alert messages always are written, regardless of the level selected in the MESSAGE_LEVEL option.

2.5.4 Message Destinations

The location to which messages are written is the message destination.

Some Universal Products have a MESSAGE_DESTINATION configuration option that specifies the message destination. If a program is used only from the command line or batch job, it may have only one message destination, such as standard error.

Valid message destination values depend on the host operating system.

z/OS Message Destinations

Universal Products on z/OS run as batch jobs or started tasks. Batch jobs do not provide the MESSAGE_DESTINATION option. All messages are written to the SYSOUT ddname.

Started task message destinations are listed in the table below.

Destination	Description
LOGFILE	Messages are written to ddname UNVLOG. All messages written to log files include a date and time stamp and the program's USS process ID.
SYSTEM	Messages are written to the console log as WTO messages.

UNIX Message Destinations

Message destinations are listed in the table below.

Destination	Description
STDERR	Messages are written to standard error. This destination is most useful for console commands.
LOGFILE	Messages are written to a log file. Not all programs provide this destination. The recommended directory for log files is **`/var/opt/universal/log`**. This can be changed with the LOG_DIRECTORY option. All messages written to log files include a date and time stamp and the program's process ID.
SYSTEM	Messages are written to the syslog daemon. Not all programs provide this destination. Universal programs that execute as daemons write to the syslog's daemon facility. All messages include the programs process ID. If an error occurs writing to the syslog, the message is written to the system console.

Windows Message Destinations

Message destinations are listed in the table below.

Destination	Description
STDERR	Messages are written to standard error. This destination is most useful for console commands.
LOGFILE	Messages are written to a log file. Not all programs provide this destination. Log files are written to product specific log directories, which can be modified with the LOG_DIRECTORY option. All messages written to log files include a date and time stamp and the program's process ID.
SYSTEM	Messages are written to the Windows Application Event Log.

OS/400 Message Destinations

Message destinations are listed in the table below.

Destination	Description
STDERR	Messages are written to standard error. A batch job's standard error file is allocated to the print file QPRINT.
LOGFILE	Messages are written to the job's job log.
SYSTEM	Messages are written to the system operator message queue QSYSOPR.

HP NonStop Message Destinations

Message destinations are listed in the table below.

Destination	Description
STDERR	Messages are written to standard error.
LOGFILE	Messages are written to a log file. Not all programs provide this destination. Log files are written the **$SYSTEM.UNVLOG** subvolume. All messages written to log files include a date and time stamp and the program's process ID.

2.6 X.509 Certificates

A certificate is an electronic object that identifies an entity. It is analogous to a passport in that it must be issued by a party that is trusted by all who accept the certificate. Certificates are issued by trusted parties called Certificate Authorities (CA's). For example, VeriSign Inc. is a CA that most parties trust. We all have faith that a trusted CA takes the necessary steps to confirm the identity of a user before issuing the user a certificate.

Certificate technology is based on public/private key technology. There are a few different types of public/private keys: RSA, DH, and DSS. As their name denotes, the private key must be kept private, like a password. The public key can be given to anyone or even published in a newspaper.

A property of public/private keys is that data encrypted with one can be decrypted only with the other. Therefore, if someone wants to send you a secret message, they encrypt the data with your public key, which everyone has. However, since you are the only one with your private key, you are the only one who can decrypt it. If you want to send someone message, such as a request for $100,000 purchase, you can "sign" it with your private key.

Note: Signing does not encrypt the data. Once a person receives your request, that person can verify it is from you by verifying your electronic signature with your public key.

A certificate ties a statement of identity to a public key. Without the public key, the certificate is meaningless. Possession of a certificate alone does not prove your identity. You must have the corresponding private key. The two together prove your identity to any third party that trusts the CA that issued your certificate. This is a key point; if you do not trust the CA that signed a certificate, you cannot trust the certificate.

Since certificates originally were designed to be used for internet authentication, global directory technologies were developed to make them available via the internet. This directory technology is known as X.500 Directory Access Protocol. Later LDAP was introduced by Netscape to make it Lightweight Directory Access Protocol.

X.500 divides the world into a hierarchical directory. A person's identity is located by traversing down the hierarchy until it reaches the last node. Each node in the hierarchy consists of a type of object, such as a country, state, company, department, or name.

2.6.1 Sample Certificate Directory

Figure 2.5, below, provides a sample diagram of a small X.500 directory.

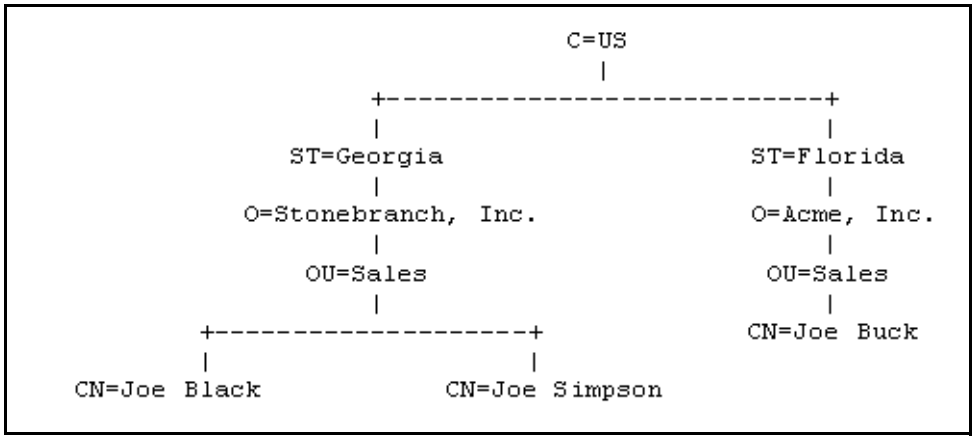

Figure 2.5 X.500 Directory (sample)

The keywords listed on each node are referred to as a Relative Distinguished Name (RDN). A person is identified by a Distinguished Name (DN). The DN value for Joe Black is **`C=US/ST=Georgia/O=Stonebranch, Inc./OU=Sales/CN=Joe Black`**.

A certificate is composed of many fields and possible extensions. Many of the most popular fields are specified as X.500 DN values.

2.6.2 Sample X.509 Certificate

Figure 2.6, below, illustrates a sample X.509 version 3 certificate for Joe Buck at the Acme corporation.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            01:02:03:04:05:06:07:08
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=US, ST=Florida, O=Acme, Inc., OU=Security, CN=CA
Authority/emailAddress=ca@acme.com
        Validity
            Not Before: Aug 20 12:59:55 2004 GMT
            Not After : Aug 20 12:59:55 2005 GMT
        Subject: C=US, ST=Florida, O=Acme, Inc., OU=Sales, CN=Joe Buck
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:be:5e:6e:f8:2c:c7:8c:07:7e:f0:ab:a5:12:db:
                    fc:5a:1e:27:ba:49:b0:2c:e1:cb:4b:05:f2:23:09:
                    77:13:75:57:08:29:45:29:d0:db:8c:06:4b:c3:10:
                    88:e1:ba:5e:6f:1e:c0:2e:42:82:2b:e4:fa:ba:bc:
                    45:e9:98:f8:e9:00:84:60:53:a6:11:2e:18:39:6e:
                    ad:76:3e:75:8d:1e:b1:b2:1e:07:97:7f:49:31:35:
                    25:55:0a:28:11:20:a6:7d:85:76:f7:9f:c4:66:90:
                    e6:2d:ce:73:45:66:be:56:aa:ee:93:ae:10:f9:ba:
                    24:fe:38:d0:f0:23:d7:a1:3b
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Alternative Name:
                email:joe.buck@acme.com
    Signature Algorithm: md5WithRSAEncryption
        a0:94:ca:f4:d5:4f:2d:da:a8:6d:e3:41:6e:51:83:57:b3:b5:
        31:95:32:b6:ca:7e:d1:4f:fb:01:82:db:23:a0:39:d8:69:71:
        31:9c:0a:3b:ce:f6:c6:e2:5c:af:23:f0:d7:ee:87:3e:8a:7b:
        40:03:39:64:a1:8c:29:7d:5b:99:93:fa:23:19:e1:e4:ac:4d:
        13:0f:de:ad:51:27:e3:4e:4b:9f:40:4c:05:fd:f2:82:09:3e:
        46:05:f0:ad:cc:f7:78:25:3e:11:f8:ca:b6:df:f7:37:57:9b:
        63:00:d0:b5:b5:18:ec:38:73:d2:85:a3:c7:24:21:47:ee:f2:
        8c:0d
```

Figure 2.6 X.509 Version 3 Certificate (sample)

Note: The contents of a certificate file does not look like the information in Figure 2.6, which is produced by a certificate utility that uses the certificate file as input. Certificates can be saved in multiple file formats, so their file contents will look very different.

2.6.3 Certificate Fields

A certificate is composed of many fields.

Figure 2.7, below, describes the main certificate fields.

Field or Section	Description
Version	X.509 certificates come in two versions: 1 and 3.
Serial Number	CA is required to provide each certificate it issues a unique serial number. The serial number is not unique for all certificates, only for the certificates issued by each CA.
Issuer	DN name of the CA that issued the certificate.
Validity	Starting and ending date for which this certificate is valid.
Subject	Identity of the certificate. A certificate may identify a person or a computer. In this case, the certificate identifies Joe Buck in the Sales organization of the Acme company in the state of Florida in the United States.
Public Key	Public key associated with the certificate identity.
X509v3 Extensions	X.509 version 3 introduced this section so that additional certificate fields may be added. In this case, the identity's email address is included as a Subject Alternative Name field. This section is not available in X.509 version 1.
Signature	CA's digital signature of the certificate.

Figure 2.7 Certificate Fields

2.6.4 SSL Peer Authentication

The SSL protocol utilizes X.509 certificates to perform peer authentication. For example, a Universal Command Manager may want to authenticate that it is connected to the correct Universal Broker.

Peer authentication is performed by either one or both of the programs involved in the network session. If a Manager wishes to authenticate the Broker to which it connects, the Broker will send its certificate to the Manager for the Manager to authenticate. Should the Broker wish to authenticate the Manager, the Manager sends its certificate to the Broker.

Certificate authentication is performed in the following steps:

1. Check that the peer certificate is issued by a trusted CA.
2. Check that the certificate has not been revoked by the CA.
3. Check that the certificate identifies the intended peer.

If a step fails, the network session is terminated immediately.

Certificate Verification

The Universal Product must be configured with a list of trusted CA certificates. When a peer certificate is received, the trusted CA certificates are used to verify that the peer certificate is issued by one of the trusted CA's.

The trusted CA certificate list must be properly secured so that only authorized accounts have update access to the list. Should the trusted CA list become compromised, there is a possibility that an untrusted CA certificate was added to the list.

The CA certificate list configuration option is CA_CERTIFICATES. It specifies a PEM formatted file that contains one or more CA certificates used for verification.

Should a peer certificate not be signed by a trusted CA, the session is immediately terminated.

Certificate Revocation

After a certificate is verified to have come from a trusted CA, the next step is to check if the CA has revoked the certificate. Since a certificate is held by the entity for which it identifies, a CA cannot take a certificate back after it is issued. So when a CA needs to revoke a certificate for some reason, it issues a list of revoked certificates referred to as the Certificate Revocation List (CRL). A program that validates certificates needs to have access to the latest CRL issued by the CA.

The CERTIFICATE_REVOCATION_LIST configuration option specifies the PEM formatted file that contains the CRL. This option is available in all Universal Products that utilize certificates.

Certificate Identification

Once a certificate is validated as being issued by a trusted CA, and not revoked by the CA, the next step is to check that it identifies the intended peer.

A Universal Products Manager validates a Broker certificate by the Broker host name or IP address or the certificate serial number. The VERIFY_HOST_NAME configuration option is used to specify the host name or IP address that is identified in the Broker certificate. Each certificate signed by a CA must have a unique serial number for that CA. The VERIFY_SERIAL_NUMBER option is used to specify the serial number in the Broker certificate.

Should certificate identification fail, the session is immediately terminated.

Universal Brokers work differently than the Managers. A Broker maps a peer certificate to a certificate ID. The certificate map definitions are part of the Universal Access Control List (UACL) definitions. At that point, the certificate ID is used by UACL definitions to control access to Broker and Server services.

Certificate Support

Many certificate authority applications, also known as Public Key Infrastructure (PKI) applications, are available. Universal Products should be able to utilize any certificate in a PEM format file. PEM (Privacy Enhanced Mail) is a common text file format used for certificates, private keys, and CA lists.

Universal Products support X.509 version 1 and version 3 certificates.

Although implementing a full featured PKI infrastructure is beyond the scope of Universal Products and this documentation, some assistance is provided using the OpenSSL toolkit (http://www.openssl.org).

Universal Products on most of the supported platforms utilize the OpenSSL toolkit for its SSL and certificate implementation. OpenSSL is delivered on most UNIX distributions and Windows distributions are available on the OpenSSL web site.

Universal Products supports z/OS System SSL on the IBM z/OS operating system as well as OpenSSL. System SSL interfaces directly with the RACF security product for certificate access. All certificates, CA and user certificates, and private keys must be stored in the RACF database to use System SSL.

The Universal Product suite includes an X.509 certificate utility, Universal Certificate, to create certificates for use in the Universal Product suite. See the Universal Certificate chapter in the Universal Products Utilities 3.2.0 User Guide for details.

Chapter 3
Universal Enterprise Controller

3.1 Overview

This chapter describes Universal Enterprise Controller (UEC) started procedure configuration.

It contains the following sections:

3.2 UEC Information

UEC controls two types of information:

- UEC-maintained information
- UEC-monitored information

3.2.1 UEC-Maintained Information

The information that UEC maintains is organized into four categories:

1. Users
2. Agents
3. SAP Systems
4. Groups

This information is maintained via the UEC Administration utility (see Chapter 3 UEC Administration in the Universal Enterprise Controller 3.2.0 Client Applications guide).

Users

Only valid UEC users have access to the UEC client applications (see Section 1.3.3 UEC Client Applications (for Windows). Each UEC user has a user name and password.

Associated with each user is:

- Set of permissions specifying the operations that the user can perform with UEC.
- List of groups containing the agents that the user can interact with via UEC.

UEC maintains its own user list. Each UEC user is assigned a set of permissions and user group membership.

Agents

An agent consists of:

- Agent name.
- Host address.
- Port on which the agent's Universal Broker is listening.

Adding an agent to UEC puts the agent on the UEC polling list. The agent then will be polled each cycle. Information about the status of the agent is sent back to any agent-monitoring utilities connected to UEC.

SAP Systems

An SAP system consists of:

- System name.
- Application Server Host (ASHOST)
- Client Number
- System Number

Adding an SAP system to UEC puts the system on the UEC polling list. The SAP system then will be polled each cycle. Information about the status of the SAP system is sent back to UAM clients connected to UEC.

Groups

Groups provide a simple way of organizing agents and/or SAP systems. Each agent or SAP system can belong to one or more groups.

(All agents are placed automatically in the pre-defined **`All Agents`** group. All SAP systems are placed automatically in the pre-defined **`All SAP Systems`** group.)

Users have access only to the groups assigned to them by their UEC administrator. This means that a user working with the Universal Activity Monitor application can only monitor agents and/or SAP systems in the groups assigned to that user.

3.2.2 UEC-Monitored Information

The information that UEC monitors is organized into four categories:

1. Alerts
2. Jobs
3. Files
4. Systems

This information can be viewed via the Universal Activity Monitor utility (see Chapter 4 Universal Activity Monitor in the Universal Enterprise Controller 3.2.0 Client Applications guide).

Alerts

UEC monitors alerts for all agents and SAP systems assigned to UEC.

Alerts are monitored until the alert condition has resolved.

Jobs

UEC monitors all Universal Command and Universal Data Mover jobs (active, completed, and failed) for all agents assigned to UEC.

Files

UEC monitors all files (active, completed, and failed) transferred by UDM for the Universal Agents being monitored by UEC.

Systems

UEC monitors all Universal Agents and SAP systems that have been assigned to UEC via the UEC Administration utility (see Chapter 3 UEC Administration in the Universal Enterprise Controller 3.2.0 Client Applications guide).

Note: The UEC MONITOR_EVENT_EXPIRATION option defines the length of time that each job and file is monitored (default is 24 hours).

3.3 Polling

3.3.1 Agent Polling

UEC periodically polls each agent in order to retrieve its status information. The polling request is made on the listening port for the agent's Universal Broker (default 7887).

When UEC polls a agent, it determines whether or not a change in status of the agent has occurred since the last poll. If the agent status has changed, UEC sends this information to the Universal Activity Monitor to notify users.

The values specified for the following configuration options affect how polling occurs:

- BKR_QUERIES_PER_THREAD
- BKR_QUERY_THREADS
- BKR_QUERY_TIMEOUT
- POLLING_INTERVAL

These configuration values can be modified, allowing UEC to fit your monitoring needs.

Optimally, UEC attempts to poll every agent in the time interval specified by POLLING_INTERVAL. However, you can define an independent polling interval for a specific agent via the UEC Administration application. For example, if UEC Administration defines a polling interval of 10 seconds for **`agent 123`**, UEC will poll **`agent 123`** every 10 seconds and all other agents at the interval specified by POLLING_INTERVAL.

If, upon a poll, UEC is unable to complete communication with a agent in the number of seconds specified by BKR_QUERY_TIMEOUT, an error is reported which indicates that the agent has timed out.

Use the following equation to calculate the number of agents that UEC can poll at any given time:

Number of agents **=** BKR_QUERIES_PER_THREAD **x** BKR_QUERY_THREADS

Note: UEC can retrieve health and status information only from Universal Broker versions of 1.2.0 and higher. Earlier versions will be reported by UEC as unreachable or not running.

3.3.2 SAP System Polling

UEC periodically polls each SAP system in order to retrieve its status information. The polling request is performed via an RFC connection to the SAP system. When UEC polls a SAP system, it determines if a change in status of the system has occurred since the last poll. If the SAP system status has changed, UEC sends this information to the Universal Activity Monitor to notify users.

In order to prevent the accidental locking of SAP accounts used by UEC, an SAP system will be dropped from the polling cycle if a logon authentication error occurs. This will prevent UEC from exceeding the number of failed logon attempts allowed by the SAP system.

When an SAP system is disabled due to a logon authentication error, a UNV4363T message is printed to the UEC log and an alert is sent to UAM clients monitoring for alerts.

SAP system definitions that have been disabled due to a logon authentication error can be re-enabled by modifying the User ID, Password, or Client field via the UEC Administration client. When an SAP system is re-enabled, a UNV1059T message is printed to the UEC log and the associated alert is removed from UAM clients.

3.4 Universal Enterprise Controller for z/OS

Universal Enterprise Controller (UEC) for z/OS executes as a started task.

3.4.1 Starting UEC

The UEC started task, **UECTLR**, is started with the z/OS START command:

```
S UECTLR
```

3.4.2 Stopping UEC

The UEC started task is stopped with the z/OS MODIFY STOP command:

```
P UECTLR
```

After the STOP command is issued, UEC may take several seconds to shut down.

Note: The **UECTLR** started task should run at a high dispatch priority in order to avoid not being dispatched in a timely enough manner to process the agent polling protocol. If **UECTLR** is not dispatched appropriately, the Broker may be reported as timed out when the Broker itself still is operational.

3.4.3 System MODIFY Command

The UEC started task accepts commands via the system MODIFY command. The MODIFY command's **APPL=** parameter is required, since UEC runs as a USS address space.

In the example below, the procedure name **UECTLR** is assumed.

```
F UECTLR,APPL=DUMP
```

The DUMP command directs UEC to produce a Language Environment dump. The dump is written to the **CEEDUMP** ddname. While the dump is being produced, UEC is paused by LE until the dump completes, after which UEC continues processing.

The DUMP command is used for diagnostic purposes. It should be executed only at the request of Stonebranch, Inc.

3.4.4 JCL Procedure

Figure 3.1, below, illustrates the Universal Enterprise Controller for z/OS JCL procedure (**UECTLR**, located in the **SUNVSAMP** library).

```
//UECTLR   PROC SHLQ=#SHLQ.UNV,
//              PHLQ=#PHLQ.UNV,
//              RGN=100M,
//              UPARM=,
//              LEPARM=,
//              CFG=UECCFG00
//S1       EXEC PGM=UECTLR,REGION=&RGN,
//              PARM='ENVAR(TZ=EST5EDT) &LEPARM/&UPARM'
//STEPLIB  DD  DSN=&SHLQ..SUNVLOAD,
//              DISP=SHR
//UNVCONF  DD  DSN=&PHLQ..UNVCONF(&CFG),
//              DISP=SHR
//UNVNLS   DD  DSN=&SHLQ..SUNVNLS,
//              DISP=SHR
//UNVDB    DD  DSN=&PHLQ..UECDB,
//              DISP=SHR
//UNVMSGS  DD  SYSOUT=*,HOLD=YES
//UNVPRSR  DD  SYSOUT=*,HOLD=YES
//UNVTRACE DD  SYSOUT=*,HOLD=YES
//SYSPRINT DD  SYSOUT=*,HOLD=YES
//SYSOUT   DD  SYSOUT=*,HOLD=YES
//CEEDUMP  DD  SYSOUT=*,HOLD=YES
//SYSIN    DD  DUMMY
```

Figure 3.1 Universal Enterprise Controller for z/OS – JCL Procedure

3.4.5 DD Statements used in JCL Procedure

Table 3.1, below, describes the DD statements used in the Universal Enterprise Controller for z/OS JCL procedure illustrated in Figure 3.1.

ddname	DCB Attributes	Mode	Description
STEPLIB	DSORG=PO, RECFM=U	input	Universal Products load library containing the program being executed.
UNVCONF	DSORG=PS, RECFM=(F, FB, V, VB)	input	UEC configuration member.
UNVNLS	DSORG=PO, RECFM=(F, FB, V, VB)	input	Universal Products national language support library. Contains message catalogs and code page translation tables.
UNVDB	DSNTYPE=HFS	input, output	UEC database.
UNVMSGS	DSORG=PS, RECFM=(F, FB, V, VB)	output	UEC message trace data.
UNVPRSR	DSORG=PS, RECFM=(F, FB, V, VB)	output	UEC parser trace data.
UNVTRACE	DSORG=PO, RECFM=(F, FB, V, VB), LRECL=256 or above.	output	UEC trace output.
SYSPRINT	DSORG=PS, RECFM=(F, FB, V, VB)	output	Standard output file for the UEC program.
SYSOUT	DSORG=PS, RECFM=(F, FB, V, VB)	output	Standard error file for the UEC program.
SYSIN	DSORG=PS, RECFM=(F, FB, V, VB)	input	Standard input file for the UEC program.

Table 3.1 Universal Enterprise Controller for z/OS – DD Statements in JCL Procedure

3.4.6 Configuration Options

This section identifies the configuration options used to execute Universal Enterprise Controller for z/OS.

Option Name	Description
BKR_QUERIES_PER_THREAD	Maximum number of simultaneous Broker queries allowed for each thread.
BKR_QUERY_THREADS	Number of process threads started to initiate Broker queries during a polling cycle.
BKR_QUERY_TIMEOUT	Period of time within which a Broker query must finish before timing out.
CA_CERTIFICATES	UEC started task procedure ddname from which a PEM-formatted list of certificates is read.
CERTIFICATE	UEC started task procedure ddname from which a PEM-formatted certificate is read.
CERTIFICATE_REVOCATION_LIST	File name / ddname of the PEM-formatted CRL.
CODE_PAGE	Code page for text translation of network data.
COMM_SESSIONS_PER_THREAD	Maximum number of UEC client sessions that can occur on each of the communications threads.
COMM_THREADS	Number of threads created to perform communications between UEC and the UEC client applications.
COMMIT_COMPLETE_EXPIRATION	Deletes completed commit configurations, by age.
COMMIT_INCOMPLETE_EXPIRATION	Deletes incomplete commit configurations, by age.
CONVERT	Converts a pre-3.2.0 database into the current database format.
DELETE_EVENTS_ON_BROKER	Specification for whether or not events are deleted on the Universal Broker after they are retrieved and put into the UEC events database.
DNS_CACHE_TIMEOUT	Length of time to retain a resolved host name in memory cache.
DNS_POLLING_INTERVAL	Time interval at which the DNS cache is polled.
HELP	Write options help to SYSPRINT ddname.
HOSTNAME_RETRY_COUNT	Number of times that UEC will attempt to resolve the host name of a specified Universal Broker before it ends with a connect error.
JOB_THREADS	Number of threads created to perform internal tasks in UEC.
KEEP_MONITOR_EVENTS	Specification for whether or not monitor events are written into the UEC temporary database.
LOG_MESSAGES	Specification for whether or not to log all XML message traffic between UEC and any connected applications.
LOGIN_ATTEMPTS	Number of failed login attempts allowed by a user before being disconnected by UEC.
MESSAGE_DESTINATION	Location to which messages are written.
MESSAGE_LANGUAGE	Language used for messages.
MESSAGE_LEVEL	Level of messages written.

Option Name	Description
MONITOR_EVENT_EXPIRATION	Length of time that state data is retained in the UEC database.
MOUNT_POINT	HFS directory in which the HFS database allocated to ddname UNVDB is mounted.
MOUNT_POINT_MODE	HFS access permission mode value with which the mounted database file system's root directory is set.
PERSISTENT_EVENT_EXPIRATION	Deletes event records, by age.
POLLING_INTERVAL	Time interval at which agents are polled.
PRIVATE_KEY	UEC started task procedure ddname from which a PEM-formatted private key is read.
PRIVATE_KEY_PWD	Password for the PRIVATE_KEY.
SAF_KEY_RING	SAF certificate key ring name.
SAF_KEY_RING_LABEL	SAF certificate key ring label.
SAP_POLLING_INTERVAL	Interval (in seconds) at which the SAP systems are polled for their status and job activity.
SERVICE_IP_ADDRESS	IP interface from which to accept connections.
SERVICE_PORT	Port from which to accept connections.
SSL_CIPHER_LIST	SSL cipher suite to be used for network communications.
SSL_IMPLEMENTATION	SSL implementation to be used for network configuration.
TMP_DIRECTORY	HFS directory in which Universal Enterprise Controller creates temporary files.
TRACE_FILE_LINES	Maximum number of lines written to the trace ddname.
TRACE_TABLE	Size of the trace table.
UPDATE_INTERVAL	Time interval at which connected Universal Activity Monitor clients are updated.
USER_AUTHENTICATION_METHOD	Authentication method to be used when authenticating UEC user accounts.
VERSION	Writes the program version and copyright statement.

Table 3.2 Universal Enterprise Controller for z/OS – Configuration Options

3.4.7 Command Line Syntax

Figure 3.2, below, illustrates the command line syntax – using the long form of configuration options – of Universal Enterprise Controller for z/OS.

```
uec
[-ca_certs ddname]
[-cert ddname [-private_key ddname [-private_key_pwd pwd ] ] ]
[-crl ddname]
[-codepage codepage]
[-convert]
[-hostname_retry_count count]
[-keep_monitor_events option]
[-dest destination]
[-lang language]
[-level {trace|audit|info|warn|error}]
[-mount_point directory]
[-mount_point_mode mode]
[-saf_key_ring name]
[-saf_key_ring_label label]
[-svcipaddr ipaddress]
[-svcport port]
[-ssl_cipher_list cipherlist]
[-ssl_implementation {openssl|system}]
[-tracefilelines lines]

uec
{-help | -version}
```

Figure 3.2 Universal Enterprise Controller for z/OS – Command Line Syntax

For a description of the options, see Chapter 2 Universal Enterprise Controller Configuration Options.

3.5 Universal Enterprise Controller for Windows

Universal Enterprise Controller for Windows executes as a service.

Changes to UEC configuration requires service be stopped and restarted by the Windows Service Control Manager.

3.5.1 Starting UEC

By default, service is set to start automatically whenever Windows is booted.

Changes to UEC configuration requires service be stopped and restarted by the Windows Service Control Manager.

To access the Service Control Manager:

1. Click the **Control Panel** on the Windows Start menu.
2. Double-click the **Administrative Tools** icon on the Control Panel window.
3. Double-click the **Services** icon on the Administrative Tools window.
4. On the Services window:
   a. Select Universal Enterprise Controller in the list of services.
   b. Click **Start** in the Action menu.

3.5.2 Stopping UEC

Changes to UEC configuration requires service be stopped and restarted by the Windows Service Control Manager.

To access the Service Control Manager:

1. Click the **Control Panel** icon on the Windows Start menu.
2. Double-click the **Administrative Tools** icon on the Control Panel window.
3. Double-click the **Services** icon on the Administrative Tools window.
4. On the Services window:
   a. Select Universal Enterprise Controller in the list of services.
   b. Click **Stop** in the Action menu.

The service is set to start automatically whenever Windows is booted.

3.5.3 Configuration Options

This section identifies the configuration options used to execute Universal Enterprise Controller for Windows.

Option Name	Description
BKR_QUERIES_PER_THREAD	Maximum number of simultaneous Broker queries allowed for each thread.
BKR_QUERY_THREADS	Number of process threads started to initiate Broker queries during a polling cycle.
BKR_QUERY_TIMEOUT	Period of time within which a Broker query must finish before timing out.
CA_CERTIFICATES	UEC started task procedure ddname from which a PEM-formatted list of certificates is read.
CERTIFICATE	UEC started task procedure ddname from which a PEM-formatted certificate is read.
CERTIFICATE_REVOCATION_LIST	File name / ddname of the PEM-formatted CRL
CODE_PAGE	Code page for text translation of network data.
COMM_SESSIONS_PER_THREAD	Maximum number of UEC client sessions that can occur on each of the communications threads.
COMM_THREADS	Number of threads created to perform communications between UEC and the UEC client applications.
COMMIT_COMPLETE_EXPIRATION	Deletes completed commit configurations, by age.
COMMIT_INCOMPLETE_EXPIRATION	Deletes incomplete commit configurations, by age.
DELETE_EVENTS_ON_BROKER	Specification for whether or not events are deleted on the Universal Broker after they are retrieved and put into the UEC events database.
DNS_CACHE_TIMEOUT	Length of time to retain a resolved host name in memory cache.
DNS_POLLING_INTERVAL	Time interval at which the DNS cache is polled.
HOSTNAME_RETRY_COUNT	Number of times that UEC will attempt to resolve the host name of a specified Universal Broker before it ends with a connect error.
JOB_THREADS	Number of threads created to perform internal tasks in UEC.
KEEP_MONITOR_EVENTS	Specification for whether or not monitor events are written into the UEC temporary database.
LOG_MESSAGES	Specification for whether or not to log all XML message traffic between UEC and any connected applications.
LOG_MESSAGES_DIRECTORY	Directory used for UEC log messages.
LOGIN_ATTEMPTS	Number of failed login attempts allowed by a user before being disconnected by UEC.
MESSAGE_DESTINATION	Location to which messages are written.
MESSAGE_LANGUAGE	Language used for messages.
MESSAGE_LEVEL	Level of messages written.
MONITOR_EVENT_EXPIRATION	Length of time that state data is retained in the UEC database.

Option Name	Description
PERSISTENT_EVENT_EXPIRATION	Deletes event records, by age.
POLLING_INTERVAL	Time interval at which agents are polled.
PRIVATE_KEY	UEC started task procedure ddname from which a PEM-formatted private key is read.
PRIVATE_KEY_PWD	Password for the PRIVATE_KEY.
SAP_POLLING_INTERVAL	Interval (in seconds) that the SAP systems are polled for their status and job activity.
SERVICE_IP_ADDRESS	IP interface from which to accept connections.
SERVICE_PORT	Port from which to accept connections.
SSL_CIPHER_LIST	SSL cipher suite to be used for network communications.
TRACE_DIRECTORY	Directory used for UEC trace files.
TRACE_FILE_LINES	Maximum number of lines written to the trace ddname.
TRACE_TABLE	Size of the trace table.
UPDATE_INTERVAL	Time interval at which connected Universal Activity Monitor clients are updated.
USER_AUTHENTICATION_METHOD	Authentication method to be used when authenticating UEC user accounts.

Table 3.3 Universal Enterprise Controller for Windows – Configuration Options

Chapter 4
Universal Event Subsystem

4.1 Overview

The Universal Event Subsystem (UES) is a subsystem of Universal Enterprise Controller.

UES records, routes, and manages event messages generated by Universal Product components.

The event messages are generated whenever a Universal Product component performs an action that impacts the computing environment on which it executes.

4.2 Event Messages

An event message contains information that identifies:

- Source of the event
- Data relating to the event itself

Event messages are collected by Universal Brokers from components that run local to the Brokers. Universal Enterprise Controller (UEC), in turn, collects the event messages from the Brokers. UEC stores the collected event messages into a database for long-term management and access.

4.2.1 Examples

Examples of event messages include:

- Universal Command Server starts a user job, which may be a command, script, or other form of work.
- Universal Broker denies access to a client due to a Universal Access Control List denial.
- Universal Data Mover Manager transfers a file from one server to another.

4.2.2 Universal Broker Event Message Processing

Universal Products components generate event messages and route them to a Universal Broker running on the same system; that is, the local Universal Broker. The Broker receives the event messages and records them into a local UES database.

Event messages are recorded in the order in which they are received by the Broker. This order is maintained throughout the subsystem.

Note: This order is based on the time that the Broker records the event, not the time that the component generates the event.

The Broker UES database maintains the event messages generated by local Universal Product components. The Broker can be stopped and restarted with no loss of event messages. The event messages remain in the database until the Broker deletes them.

4.3 UES Activation

The Universal Event Subsystem is not activated by default.

In order to generate and capture event messages, each Universal Products component that is able to generate event messages has an EVENT_GENERATION option. This option controls which event message types to generate.

By default, EVENT_GENERATION is set so that no event message types are generated. The value must be set so that event messages of interest are generated by the component.

4.3.1 Broker UES Database Cleanup

The UES database continues to accumulate event messages until the Broker deletes them.

Event messages are deleted based upon two criteria:

1. Event message expires.
2. Event message is delivered to a Universal Enterprise Controller that requested delete access to event messages.

Event message expiration is controlled with the EVENT_EXPIRATION option. This option specifies the number of seconds that an event message should remain in the UES database before it is eligible for deletion. Each event message contains the time that it was recorded in the database. The Broker considers an event message expired if the difference between the current time and the recorded time is greater than the EVENT_EXPIRATION value.

The consequences of this using this method for determining whether or nor an event message is expired is that if the value of EVENT_EXPIRATION is increased or decreased, the life of all recorded event messages is increased or decreased as well.

4.3.2 Broker UES Database Access

A Broker provides UES database access to Universal Enterprise Controller (UEC). UEC sends a request to a Broker asking for the latest event messages. The Broker responds with event messages that satisfy the UEC request.

The Universal Access Control List (UACL) entries EVENT_READ and EVENT_DELETE control read and delete access, respectively, to the UES database.

The default EVENT_READ rule allows read access. The default EVENT_DELETE rule denies access. These UACL defaults allow any UEC read access to event messages while denying all UEC's delete access to event messages.

An event message becomes eligible for deletion from the Broker UES database once it has been delivered to a UEC that requested delete access. There should be one UEC designated as the production UEC responsible for maintaining the central UES database for all Brokers. This one production UEC should be given delete access on each Broker.

Chapter 5
UECLoad Utility

5.1 Overview

This chapter provides information on the UECLoad utility specific to the z/OS and Windows operating systems.

UECLoad provides the user with a command line interface to add, delete, view, and export data from the Universal Enterprise Controller database tables.

5.2 Usage

UECLoad executes as a command line application.

Through the use of UECLoad, the user can:

- Add, delete, list, or export individual Universal Agent definitions.
- Provide a Universal Agent definition file to add, delete, list, or export multiple Universal Agents.
- Delete, list, or export the currently defined Universal Agents in the UEC database.
- Export Universal Event Subsystem events, with the option to delete them from UEC.

This section describes the configuration, configuration options, and command line syntax of UECLoad.

Section 5.3 Examples of UECLoad provides examples demonstrating the flexibility of UECLoad.

5.2.1 UECLoad for z/OS

This section identifies the following information for UECLoad for z/OS:

- JCL
- DD Statements used in JCL

JCL

Figure 5.1, below, illustrates the JCL required to execute UECLoad for z/OS.

```
//STEP1    EXEC PGM=UECLOAD,PARM='ENVAR(TZ=EST5EDT)/'
//STEPLIB  DD  DISP=SHR,DSN=#SHLQ.UNV.SUNVLOAD
//*
//UNVCONF  DD  DISP=SHR,DSN=#PHLQ.UNV.UNVCONF(UECCFG00)
//*
//UNVTRACE DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
//*
//LOAD     DD  *
<BROKERDEF>
  broker_name   unxprod
  broker_host   prd-unix
  broker_port   7887
</BROKERDEF>
<BROKERDEF>
  broker_name   unxtest
  broker_host   tst-unix
  broker_port   7887
</BROKERDEF>
<BROKERDEF>
  broker_name   unxdev
  broker_host   dev-unix
  broker_port   7887
</BROKERDEF>
/*
//*
//USER     DD  *
-u admin -w admin
/*
//*
//SYSIN    DD  *
 -add -deffile load -f user
```

Figure 5.1 Universal UECLoad for z/OS – JCL

DD Statements used in JCL

Table 5.1, below, describes the DD statements used in the UECLoad for z/OS JCL illustrated in Figure 5.1.

ddname	DCB Attributes	Mode	Description
STEPLIB	DSORG=PO, RECFM=U	input	Universal Products load library containing the program being executed.
UNVCONF	DSORG=PS, RECFM=(F, FB, V, VB)	input	UEC configuration member.
UNVTRACE	DSORG=PO, RECFM=(F, FB, V, VB), LRECL=256 or above.	output	UECLoad trace output.
SYSPRINT	DSORG=PS, RECFM=(F, FB, V, VB)	output	Standard output file for the UECLoad program.
SYSOUT	DSORG=PS, RECFM=(F, FB, V, VB)	output	Standard error file for the UECLoad program.
SYSIN	DSORG=PS, RECFM=(F, FB, V, VB)	input	Standard input file for the UECLoad program.

Table 5.1 UECLoad for z/OS – DD Statements in JCL

5.2.2 Configuration

Configuration consists of:

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

Configuration options are read from the following sources:

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

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

5.2.3 Configuration Options

This section describes the configuration options used to execute UECLoad.

Configuration Options Categories

Table 5.2, below, categorizes the configuration options into logical areas of application.

Category	Description
Action	Action being taken on the specified Universal Agent definition.
Broker Definition	Definition of the Universal Agent being modified in the UEC database.
Events	Options available when using the action –export EVENTS
Host	UEC connection options.
Miscellaneous	Options used to display command help and program versions.
Options	Alternative methods to specify command options.
User	User account that UECLoad executes with in UEC.

Table 5.2 UECLoad Utility - Configuration Option Categories

The UECLoad options for each category are summarized in the following tables. Each **Option Name** is a link to detailed information about that option in Chapter 3 UECLoad Configuration Options.

Action Category Options

Option Name	Description
ADD	Specification to add agent definitions to the UEC.
DELETE	Specification to delete agent definitions from the UEC.
EXPORT	Specification to output the described agent definitions in a format to be used by an agent definition file.
LIST	Specification to output the described agent definitions in a user-friendly format.

Broker Definition Category Options

Option Name	Description
BROKER_DESCRIPTION	Description of the defined Universal Broker.
BROKER_HOST	TCP/IP host name of the defined Universal Broker.
BROKER_NAME	Unique name of the defined Universal Broker.
BROKER_PORT	TCP/IP port number of the defined Universal Broker.

Events Category Options

Option Name	Description
ARCFILE	Archived file to retrieve for export.
END_TIME	End time of exported data.
EXPORT_DELETE	Delete records in Events database.
FORMAT	Output format of event report (formats supported are CSV, XML, and ARC).
START_TIME	Start time of exported data.

Host Category Options

Option Name	Description
UEC_PORT	TCP/IP port number of UEC.

Miscellaneous Category Options

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

Options Category Options

Option Name	Description
BROKER_DEFFILE	File containing multiple broker definitions to be added or deleted in the UEC database.
CODE_PAGE	Code page used for text translation.
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.
MESSAGE_LEVEL	Level of messages written.

User Category Options

Option Name	Description
USER_ID	UEC user ID or account with which brokers will be modified.
USER_PASSWORD	Password associated with USER_ID.

5.2.4 Command Line Syntax

Figure 5.2, below, illustrates the syntax – using the long form of command line options – of the UECLoad utility.

```
uecload
{-add | -delete | -list | -export [EVENTS] }
[-userid user [-pwd pwd] ]
[-port port]
[-broker_desc description]
[-broker_host address]
[-broker_name name]
[-broker_port port]
[-arcfile filename]
[-codepage codepage]
[-level {trace|audit|info|warn|error} ]
[-deffile filename]
[-file ddname / filename | -encryptedfile ddname / filename  [-key key] ]
[-format [XML|CVS|ARC] ]
[-export_delete]
[-stime startdate [,starttime] ]
[-etime enddate [,endtime] ]

uecload
{-help | -version}
```

Figure 5.2 UECLoad Utility - Command Line Syntax

For a description of the UECLoad configuration options, see Chapter 3 UECLoad Configuration Options.

5.3 Examples of UECLoad

This section contains examples demonstrating the use of the UECLoad utility.

The following list provides a link to each example.

- List All Defined Brokers
- Export a Specific Defined Broker
- Export Events
- Retrieve Archived File and Export
- Delete a Specific Defined Broker
- Add Specific Defined Broker via deffile
- Export Events into ARC Format (z/OS)
- Retrieve Archived File and Export into XML (z/OS)
- Export Events into ARC Format (Windows)
- Retrieve Archive File and Export into CSV (Windows)

Additional z/OS examples are located in **`#HLQ.UNV.SUNVSAMP(UECLSAM1-UECLSAM4)`**.

5.3.1 List All Defined Brokers

Figure 5.3, below, illustrates the output of a user-friendly format of the Brokers defined in the UEC database.

```
uecload –port 8778 –userid joe –pwd akkSdig -list –broker_name “*”
```

Figure 5.3 UECLoad - List All Defined Brokers

5.3.2 Export a Specific Defined Broker

Figure 5.4, below, illustrates the output of a Broker defined in the UEC database in a format suitable for use within a broker definition file.

Note: Although this command is illustrated on two lines, it should be entered as one line at the command prompt.

```
uecload –port 8778 –userid joe –pwd akkSdig –level audit
            –export –broker_name mybroker1
```

Figure 5.4 UECLoad - Export a Specific Defined Broker

5.3.3 Export Events

Figure 5.5, below, illustrates the export of an events file into CSV format.

Note: Although this command is illustrated on two lines, it should be entered as one line at the command prompt.

```
uecload -port 8778 -userid joe -pwd akkSdig -level audit -export EVENTS
-stime *-5 -etime * -format CSV -deffile events.csv
```

Figure 5.5 UECLoad - Export Events

5.3.4 Retrieve Archived File and Export

Figure 5.6, below, illustrates the retrieval of an archived events file and its export into CSV format.

Note: Although this command is illustrated on two lines, it should be entered as one line at the command prompt.

```
uecload -arcfile c:\test.arc -export EVENTS -stime 2006/10/07 -etime
2008/01/01 -level audit  -format CSV -deffile c:\test.csv
```

Figure 5.6 UECLoad - Retrieve Archived File and Export

5.3.5 Delete a Specific Defined Broker

Figure 5.7, below, illustrates the deletion of a Broker defined in the UEC database. Specifically, Broker **mybroker1** is deleted from use of UEC.

Note: Although this command is illustrated on two lines, it should be entered as one line at the command prompt.

```
uecload –port 8778 –userid joe –pwd akkSdig –level audit
           –delete –broker_name mybroker1
```

Figure 5.7 UECLoad - Delete a Specific Defined Broker

5.3.6 Add Specific Defined Broker via deffile

Figure 5.8, below, illustrates the addition of a group of Broker definitions specified within a definition file in the UEC database. The name **sample_deffile** represents the name of the created file.

Note: Although this command is illustrated on two lines, it should be entered as one line at the command prompt.

```
uecload -port 8778 -userid joe -pwd akkSdig -level audit
            -add -deffile sample_deffile
```

Figure 5.8 UECLoad - Add Specific Defined Broker via a Definition File

Figure 5.9, below, is the definition file to be used for this example.

```
<BROKERDEF>
broker_name mybroker1
broker_host localhost
broker_port 7887
broker_desc "This is a description of broker1."
</BROKERDEF>
<BROKERDEF>
broker_name mybroker2
broker_host 127.0.0.1
broker_port 7887
broker_desc "This is a description of broker2."
</BROKERDEF>
<BROKERDEF>
broker_name mybroker3
broker_host 10.20.30.40
broker_port 7887
broker_desc "This is a description of broker3."
</BROKERDEF>
```

Figure 5.9 UECLoad - Definition File used for Adding Specific Defined Broker

5.3.7 Export Events into ARC Format (z/OS)

Figure 5.10, below, illustrates the export of events into an ARC format file on z/OS.

```
//STEP1      EXEC   PGM=UECLOAD,PARM='ENVAR(TZ=EST5EDT)/'
//STEPLIB    DD     DISP=SHR,DSN=#HLQ.UNV.SUNVLOAD
//*
//UNVCONF    DD     DISP=SHR,DSN=#HLQ.UNV.UNVCONF(UECCFG00)
//*
//UNVTRACE   DD     SYSOUT=*
//ARCFILE    DD     DSN=APP.UEC.ARCH,
//                  DISP=(,CATLG),UNIT=3390,VOL=SER=STG001,
//                  SPACE=(CYL,(5,5)),
//                  DCB=(RECFM=FB,LRECL=200,BLKSIZE=8000)
//SYSPRINT   DD     SYSOUT=*
//SYSOUT     DD     SYSOUT=*
//CEEDUMP    DD     SYSOUT=*
//SYSIN      DD     *
 -export EVENTS -port 8778 -u joe -w akkSdig -level audit
 -stime 2008/04/29,10:00:00 -etime 2008/04/30,10:00:00
 -format ARC -deffile ARCFILE
```

Figure 5.10 UECLoad for z/OS - Export Events into ARC Format

5.3.8 Retrieve Archived File and Export into XML (z/OS)

Figure 5.11, below, illustrates the retrieval of an archived file and its export into XML on z/OS.

```
//STEP1      EXEC   PGM=UECLOAD,PARM='ENVAR(TZ=EST5EDT)/'
//STEPLIB    DD     DISP=SHR,DSN=#HLQ.UNV.SUNVLOAD
//*
//UNVCONF    DD     DISP=SHR,DSN=#HLQ.UNV.UNVCONF(UECCFG00)
//OUTPUT     DD     SYSOUT=*
//UNVTRACE   DD     SYSOUT=*
//ARCFILE    DD     DSN=APP.UEC.ARCH,DISP=SHR
//DEFFILE    DD     DSN=APP.UEC.DEFFILE,DISP=SHR
//SYSOUT     DD     SYSOUT=*
//CEEDUMP    DD     SYSOUT=*
//SYSIN      DD     *
-export EVENTS -arcfile ARCFILE -level audit
-format XML -deffile DEFFILE
```

Figure 5.11 UECLoad for z/OS- Retrieve Archived File and Export into XML

5.3.9 Export Events into ARC Format (Windows)

Figure 5.13, below, illustrates the export of events into an ARC format file on Windows.

```
uecload -export EVENTS -u admin -pwd admin -format ARC -stime 2008/07/24
-etime 2008/07/24 -deffile c:\test.xml -arcfile c:\test.arc
```

Figure 5.12 UECLoad for Windows - Export Events into ARC Format

5.3.10 Retrieve Archive File and Export into CSV (Windows)

Figure 5.13, below, illustrates the retrieval of an archived file and its export into CSV on Windows.

```
uecload -arcfile c:\test.arc -export EVENTS -stime 2006/10/07 -etime
2008/01/01 -level audit -format CSV -deffile c:\test.csv
```

Figure 5.13 UECLoad for Windows - Retrieve Archived File and Export into CSV

Note: **`-port`**, **`-userid`**, and **`-pwd`** are not used, since no connection is made to UEC for this operation.

Chapter 6
Troubleshooting

6.1 Overview

This chapter provides information on troubleshooting Universal Enterprises Controller (UEC).

6.2 Java Under Windows

6.2.1 Java Compatibility

The UEC client applications have been tested and verified with Sun Java Runtime versions 1.5.

6.2.2 Known Problems

Java Upgrade Problems

There have been various problems reported, when installing one version of Sun's Java over another, that will cause some Java applications to work incorrectly. Un-install the original version of the JVM and install the new version. A fresh install will usually resolve these issues.

6.3 Java Under Linux

6.3.1 Java Compatibility

The UEC client applications have been tested and verified with Sun Java Runtime versions 1.5.

6.3.2 Known Problems

Wrong Window/Dialog Sizes Under KDE

The main window and dialogs may display at the incorrect sizes when using Java version 1.3.1 from Sun and the KDE window manager. Upgrading to Java version 1.4.1 or using another window manager (such as Gnome) will solve this problem.

6.4 Java Under Mac OS X

6.4.1 Java Compatibility

UEC has been tested and verified with the release 1.5 versions of Apple's JVM.

6.5 UEC Problems

6.5.1 UEC Incorrectly Reports a Universal Broker as Unreachable

UEC uses the Universal Query protocol to poll the Universal Brokers in its list.

Universal Broker versions earlier than 2.1 (or 1.2 with PTF 5 on the AS/400) do not support this protocol; they will appear to be unreachable by UEC.

If a Universal Broker being reported – incorrectly – as unreachable is of the proper version, ensure that:

- Address and port have been entered correctly
- TCP connection can be made from the machine running UEC to the machine with the incorrectly reported Broker.

Universal Query can be used to verify the connection. If you can query the Broker using Universal Query from the machine on which UEC is running, UEC should be able to poll the Broker.

Chapter 7
UEC Database Administration

7.1 Overview

Universal Enterprise Controller (UEC) uses databases to maintain agent, user, configuration, and event data. If a database becomes corrupted, it will prevent UEC from running.

7.1.1 Database Files

The UEC databases reside in three files:

1. **`uec.db`** contains the definitions of agents, groups, users, SAP systems, and a record of updates to distributed components' configurations in a managed environment.
2. **`uec_evm.db`** contains the UES persistent events.
3. **`uec_tmp.db`** contains UES events and component information that is temporary to support UAM. This file is deleted and created upon restart of UEC.

7.1.2 Database Management

Automated Database Cleanup

Two routines are run to clean up records that meet their expiration criteria from their UEC database.

1. Routine for monitor events used for Universal Activity Monitor.
2. Routine for persistent events stored for the Universal Event Subsystem.

Both routines execute at UEC start-up. Thereafter, they are scheduled to execute one hour after the previous execution's completion. At the time of execution, all records that meet the expiration criteria are removed from their UEC database.

The following UEC configuration options control database record retention:

- COMMIT_COMPLETE_EXPIRATION
- COMMIT_INCOMPLETE_EXPIRATION
- MONITOR_EVENT_EXPIRATION
- PERSISTENT_EVENT_EXPIRATION

Memory Management

Berkeley DB uses a temporary cache in memory to manage its databases. If this cache becomes sufficiently large, it must be written to disk.

Berkeley DB has a default location for storing temporary cache files, but if UEC cannot access that location, or there is no space to write these files in the default location, the following error can occur in UEC, and UEC shuts down:

```
UNV4301D Database error: 'temporary: write failed for page XXXXX'
```

To work around this issue, the following steps will write the temporary cache files to the UEC database directory:

1. For z/OS installations, mount the **UECDB** HFS dataset.
2. Inside the UEC database directory (or, on z/OS, the mount point), create a text file named **DB_CONFIG**.
3. Inside the **DB_CONFIG** file, add the following string:

   set_tmp_dir *dbpath*

   Where **dbpath** is the path to the location in which the database files reside.
4. Start / restart UEC.

7.1.3 Database Recovery

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

Databases can potentially become corrupt due to system and address spaces ending abnormally. Abnormal methods of termination include:

- z/OS CANCEL or FORCE command.
- Windows process termination through the Task Manager.

If UEC terminates abnormally, it creates the file **`uec.hf`** in the database directory, which prompts UEC to initiate database verification upon restart.

Upon start-up, if UEC determines that an abnormal termination occurred, a verification process is performed on the database files. Verification tests the integrity of the files and determines if they are suitable for opening. If errors are detected and the integrity of the file is compromised, UEC reports the errors to the console and UEC immediately shuts down.

The Universal Database Dump (**`UDBDUMP`**) utility and the Universal Database Load (**`UDBLOAD`**) utility enable recovery from a corrupted Berkeley database. (For detailed information on these utilities, see the Universal Products Utilities 3.2.0 User Guide.)

Database recovery procedures depend partly on the operating system on which UEC is executing: z/OS or Windows. The following sections describe the procedures for each operating system.

z/OS

The UEC started task must be down to perform database recovery. A backup of either the database file being recovered or the entire HFS data set should be created before recovery is attempted.

A sample database recovery job is provided in member **`UECDBREC`** in the **`SUNVSAMP`** library. The job uses the Universal Database Utilities to dump and reload a database file.

All databases are located in the HFS product data set **`#HLQ.UNV.UECDB`**. The HFS data set is allocated to the **`UNVDB`** ddname in both the dump and load steps. The HFS data set must be mounted prior to running **`UECDBREC`**. Refer to the Universal Products 3.2.0 Installation Guide for additional information on mounting the HFS data set.

The user ID with which the recovery job runs requires appropriate permissions to the root directory of the HFS data set and to the database file. Write access is required to the directory and read and write access is required to the database file.

Customize **`UECDBREC`** to meet local JCL and installation requirements. All UEC databases are recovered by the job. When all modifications are complete, submit the job. All steps should end with return code 0.

Windows

The UEC service must be stopped to perform database recovery. A backup of either the database file being recovered or the entire directory should be created before recovery is attempted.

A sample database recovery batch file is provided in file uecdbrec.bat in the **"\Program Files\Universal\UECtlr\bin"** directory. The batch file uses the Universal Database Utilities to dump and reload a database file.

The default location of all UEC databases is **"\Program Files\Universal\UECtlr"**.

Note: Stonebranch has identified an issue with upgrades *from* releases earlier than UEC 3.2.0.0 (such as 3.1.0.x or 3.1.1.x) *to* releases 3.2.0.0 and later. Following the upgrade, UEC databases reside in the location specified by the user's currently configured working_directory location. This location defaults to "**\Program Files\Universal\UECtlr\bin**".

If the current UEC install was not an upgrade, it may be necessary to pass the path to the **uec_evm.db** file as a command line argument to the script. You can provide an absolute path or a path relative to the **uecdbrec.bat** script's location.

The user ID with which the recovery script runs requires appropriate permissions to the database directory and to the database file. Write access is required to the directory and read and write access is required to the database file.

The uecdbrec.bat batch file accepts an optional argument-the database file name to recover. If no database file name is specified, the **uec_evm.db** database is recovered. The batch file ends with exit code 0 if successful and a non-zero exit code if it failed.

7.1.4 Database Backups

Database recovery is not a replacement for database backups. If the data maintained by the product in the database has long term value, the databases must be periodically backed up.

Appendix A
Customer Support

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

TELEPHONE

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

North America

(+1) 678 366-7887, extension 6

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

Europe

+49 (0) 700 5566 7887

E-MAIL

All Locations

support@stonebranch.com

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

www.stonebranch.com

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