



# **Universal Event Monitor**

## Reference Guide

Universal Products

Version 3.2.0



---

# Universal Event Monitor

## Reference Guide

### Universal Products 3.2.0

---

<b>Document Name</b>	<b>Universal Event Monitor 3.2.0 Reference Guide</b>				
<b>Document ID</b>	<b>uem-ref-3204</b>				
<b>Products</b>	<b>z/OS</b>	<b>UNIX</b>	<b>Windows</b>	<b>OS/400</b>	<b>HP NonStop</b>
Universal Event Monitor Manager	√	√	√		
Universal Event Monitor Server		√	√		
UEMLoad		√	√		

---

# Stonebranch Documentation Policy

---

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

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

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

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

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

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

© 2003-2010 by Stonebranch, Inc.

All rights reserved.



---

# Summary of Changes

---

## **Changes for Universal Event Monitor 3.2.0 Reference Guide (uem-ref-3204) September 8, 2009**

- Specified behavior when the [MAX\\_OCCURRENCE\\_COUNT](#) configuration option specifies that no event occurrences are to be monitored (value=0).

### **Universal Event Monitor 3.2.0.2**

- Added the following code pages in Section [7.3 Character Code Pages](#):
  - IBM875
  - IBM4971

## **Changes for Universal Event Monitor 3.2.0 Reference Guide (uem-ref-3203) December 17, 2008**

- Corrected the environment variable name for the Universal Event Monitor Manager [SYSTEM\\_ID](#) option.

## **Changes for Universal Event Monitor 3.2.0 Reference Guide (uem-ref-3202) October 17, 2008**

- Specified the maximum valid value for date and time in:
  - Universal Event Monitor Manager [INACTIVE\\_DATE\\_TIME](#) option
  - UEMLoad [ACTIVE\\_DATE\\_TIME](#) option

---

## Changes for Universal Event Monitor 3.2.0 Reference Guide (uem-ref-3201) September 5, 2008

- Added [Chapter 6 Universal Event Monitor UACL Entries](#).
- Added toll-free telephone number for North America in [Appendix A Customer Support](#).

## Changes for Universal Event Monitor 3.2.0 Reference Guide (uem-ref-320) May 16, 2008

### Universal Event Monitor 3.2.0

- Added the following configuration options in [Chapter 2 UEM Manager Configuration Options](#):
  - [BIF\\_DIRECTORY](#)
  - [CONNECT\\_TIMEOUT](#)
  - [DNS\\_EXPAND](#)
  - [HANDLER\\_TYPE](#)
  - [HOST\\_SELECTION](#)
  - [INSTALLATION\\_DIRECTORY](#)
  - [NLS\\_DIRECTORY](#)
  - [OPTIONS](#)
  - [PLF\\_DIRECTORY](#)
  - [SYSTEM\\_ID](#)
- Modified the [REMOTE\\_HOST](#) configuration option in [Chapter 2 UEM Manager Configuration Options](#).
- Added the [NLS\\_DIRECTORY](#) configuration option in [Chapter 3 UEM Server Configuration Options](#).
- Added the following configuration options in [Chapter 4 UEMLoad Utility Configuration Options](#):
  - [HANDLER\\_TYPE](#)
  - [OPTIONS](#)
- Added [Chapter 5 Universal Event Monitor Component Definition Options](#).
- Deleted the following specification methods for all configuration options in [Chapter 3 UEM Server Configuration Options](#):
  - Command Line, Short Form
  - Command Line, Long Form
  - Environment Variable
- Added Configuration File Keyword as a specification method for Windows configuration options in:
  - [Chapter 2 UEM Manager Configuration Options](#)
  - [Chapter 3 UEM Server Configuration Options](#)

---

# Contents

---

<b>Summary of Changes</b> .....	<b>5</b>
<b>Contents</b> .....	<b>7</b>
<b>List of Tables</b> .....	<b>13</b>
<b>Preface</b> .....	<b>14</b>
Document Structure .....	14
Format .....	14
Conventions .....	15
Vendor References .....	16
Document Organization .....	17
<b>Chapter 1 Overview</b> .....	<b>18</b>
1.1 Overview .....	18
1.2 Usage .....	19
<b>Chapter 2 UEM Manager Configuration Options</b> .....	<b>20</b>
2.1 Overview .....	20
2.2 Configuration Options Information .....	21
2.3 Configuration Options List .....	23
2.4 BIF_DIRECTORY .....	25
2.5 CODE_PAGE .....	26
2.6 COMMAND_FILE_ENCRYPTED .....	27
2.7 COMMAND_FILE_PLAIN .....	28

2.8	COMMAND_ID	29
2.9	CONNECT_TIMEOUT	30
2.10	CTL_SSL_CIPHER_LIST	32
2.11	DNS_EXPAND	33
2.12	ENCRYPTION_KEY	35
2.13	EVENT_ID	36
2.14	EVENT_STATE	37
2.15	EVENT_TYPE	38
2.16	FILE_SPECIFICATION	39
2.17	HANDLER_ID	41
2.18	HANDLER_OPTIONS	42
2.19	HANDLER_TYPE	44
2.20	HELP	46
2.21	HOST_SELECTION	47
2.22	HOSTNAME_RETRY_COUNT	48
2.23	INACTIVE_DATE_TIME	49
2.24	INSTALLATION_DIRECTORY	51
2.25	LOGIN	52
2.26	MAX_OCCURRENCE_COUNT	53
2.27	MAXIMUM_RETURN_CODE	54
2.28	MESSAGE_LANGUAGE	55
2.29	MESSAGE_LEVEL	56
2.30	MINIMUM_FILE_SIZE	58
2.31	NETWORK_DELAY	59
2.32	NLS_DIRECTORY	60
2.33	OPTIONS	61
2.34	OUTBOUND_IP	63
2.35	PLF_DIRECTORY	64
2.36	POLLING_INTERVAL	65
2.37	REMOTE_HOST	66
2.38	REMOTE_PORT	68
2.39	RENAME_FILE	69
2.40	RENAME_FILE_SPECIFICATION	70
2.41	SYSTEM_ID	72



2.42	TRACE_FILE_LINES	73
2.43	TRACE_TABLE	74
2.44	TRACKING_INTERVAL	76
2.45	USER_COMMAND	77
2.46	USER_ID	79
2.47	USER_PASSWORD	80
2.48	USER_SCRIPT	81
2.49	USER_SCRIPT_TYPE	84
2.50	VERSION	85
2.51	WAIT	86
<b>Chapter 3 UEM Server Configuration Options</b>		<b>87</b>
3.1	Overview	87
3.2	Configuration Options Information	88
3.3	Configuration Options List	90
3.4	ASSIGNED_COMPONENT_NAME	92
3.5	CODE_PAGE	93
3.6	EXPIRED_HANDLER_ID	94
3.7	INSTALLATION_DIRECTORY	95
3.8	INTERACT_WITH_DESKTOP	96
3.9	LOGIN	98
3.10	LOGON_METHOD	100
3.11	MAXIMUM_RETURN_CODE	101
3.12	MESSAGE_LEVEL	102
3.13	MINIMUM_FILE_SIZE	103
3.14	NLS_DIRECTORY	104
3.15	POLLING_INTERVAL	105
3.16	REJECTED_HANDLER_ID	106
3.17	RENAME_FILE	107
3.18	RENAME_FILE_SPECIFICATION	108
3.19	SHELL	110
3.20	TRACE_DIRECTORY	111
3.21	TRACE_FILE_LINES	112
3.22	TRACE_TABLE	113

---

3.23 TRACKING_INTERVAL .....	115
3.24 TRIGGERED_HANDLER_ID .....	116
3.25 USER_SCRIPT_TYPE .....	117
3.26 USER_SECURITY .....	118
<b>Chapter 4 UEMLoad Utility Configuration Options .....</b>	<b>120</b>
4.1 Overview .....	120
4.2 Configuration Options Information .....	121
4.3 Configuration Options List .....	123
4.4 ACTION .....	125
4.5 ACTIVE_DATE_TIME .....	127
4.6 ASSIGNED_COMPONENT_NAME .....	128
4.7 BROKER_PORT .....	130
4.8 COMMAND_ID .....	131
4.9 DEFINITION_FILE .....	132
4.10 ENCRYPTION_KEY .....	133
4.11 EVENT_ID .....	134
4.12 EVENT_STATE .....	135
4.13 EVENT_TYPE .....	136
4.14 EXPIRED_HANDLER_ID .....	137
4.15 FILE_SPECIFICATION .....	138
4.16 HANDLER_ID .....	139
4.17 HANDLER_OPTIONS .....	141
4.18 HANDLER_TYPE .....	143
4.19 HELP .....	145
4.20 INACTIVE_DATE_TIME .....	146
4.21 MAXIMUM_RETURN_CODE .....	148
4.22 MESSAGE_LEVEL .....	149
4.23 MINIMUM_FILE_SIZE .....	150
4.24 OPTIONS .....	151
4.25 REJECTED_HANDLER .....	153
4.26 RENAME_FILE .....	154
4.27 RENAME_FILE_SPECIFICATION .....	155
4.28 TRACKING_INTERVAL .....	157

4.29 TRIGGERED_HANDLER_ID .....	158
4.30 USER_COMMAND .....	159
4.31 USER_FILE_ENCRYPTED .....	160
4.32 USER_FILE_PLAIN .....	162
4.33 USER_ID .....	164
4.34 USER_PASSWORD .....	165
4.35 USER_SCRIPT .....	166
4.36 USER_SCRIPT_TYPE .....	168
4.37 VERSION .....	169
<b>Chapter 5 Universal Event Monitor Component Definition Options .....</b>	<b>170</b>
5.1 Overview .....	170
5.2 Component Definition Options Information .....	171
5.3 Component Definition Options .....	173
5.4 AUTOMATICALLY_START .....	174
5.5 COMPONENT_NAME .....	175
5.6 COMPONENT_TYPE .....	176
5.7 CONFIGURATION_FILE .....	177
5.8 RUNNING_MAXIMUM .....	178
5.9 START_COMMAND .....	179
5.10 WORKING_DIRECTORY .....	181
<b>Chapter 6 Universal Event Monitor UACL Entries .....</b>	<b>182</b>
6.1 Overview .....	182
6.2 UACL Entries Information .....	183
6.3 UACL Entries List .....	185
6.4 ACCESS_ACL .....	186
6.5 DATABASE_MAINTENANCE_ACL .....	188
6.6 EVENT_HANDLER_ACL .....	189
<b>Chapter 7 Additional Information for Universal Event Monitor .....</b>	<b>191</b>
7.1 Overview .....	191
7.2 DD Statements used in JCL .....	192
7.3 Character Code Pages .....	193
7.4 UTT Files .....	195

7.5 SSL Cipher Suites .....	196
7.6 Event Definition Parameters - General .....	197
7.7 Event Definition Parameters - Event Type FILE .....	199
7.8 Event Handler Parameters .....	201
<b>Appendix A Customer Support .....</b>	<b>203</b>

---

# List of Tables

---

<b>Chapter 2 UEM Manager Configuration Options</b> .....	<b>20</b>
Table 2.1 UEM Manager Configuration Options .....	24
Table 2.2 UEM Event Types .....	38
<b>Chapter 3 UEM Server Configuration Options</b> .....	<b>87</b>
Table 3.1 UEM Server Configuration Options .....	91
<b>Chapter 4 UEMLoad Utility Configuration Options</b> .....	<b>120</b>
Table 4.1 UEMLoad Utility - Configuration Options .....	124
Table 4.2 UEM Event Types .....	136
<b>Chapter 5 Universal Event Monitor Component Definition Options</b> .....	<b>170</b>
Table 5.1 Universal Event Monitor - Component Definition Options .....	173
<b>Chapter 6 Universal Event Monitor UACL Entries</b> .....	<b>182</b>
Table 6.1 Universal Event Monitor UACL Entries .....	185
<b>Chapter 7 Additional Information for Universal Event Monitor</b> .....	<b>191</b>
Table 7.1 Universal Event Monitor - Additional Information .....	191
Table 7.2 Universal Event Monitor Manager for z/OS - Batch JCL DD Statements .....	192
Table 7.3 Character Code Pages .....	194
Table 7.4 UTT File Locations .....	195
Table 7.5 SSL Cipher Suites .....	196
Table 7.6 Event Definition Parameters - General .....	198
Table 7.7 Event Definition Parameters - Event Type FILE .....	200
Table 7.8 Event Handler Parameters .....	202

---

# Preface

---

## Document Structure

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

## Format

---

Starting with the Universal Products 3.2.0 release, this Universal Event Monitor Reference Guide serves as a companion document to the Universal Event Monitor User Guide.

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

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

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

## Conventions

---

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

### Typeface and Fonts

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

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

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

### Operating System-Specific Text

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

**z/OS**

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

This text resumes the information pertaining to all operating systems.

### Tips from the Stoneman



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

**Stoneman's Tip**

---

## Vendor References

---

References are made throughout this document to a variety of vendor operating systems. Stonebranch, Inc. attempts to use the most current product names when referencing vendor software.

The following names are used within this document:

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

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



# Document Organization

The document is organized into the following sections:

- [Overview](#) (Chapter 1)  
Overview of the information provided in this document.
- [UEM Manager Configuration Options](#) (Chapter 2)  
Detailed information about the configuration options used with the UEM Manager component.
- [UEM Server Configuration Options](#) (Chapter 3)  
Detailed information about the configuration options used with the UEM Server component.
- [UEMLoad Utility Configuration Options](#) (Chapter 4)  
Detailed information about the configuration options used with the UEMLoad utility.
- [Universal Event Monitor Component Definition Options](#) (Chapter 5)  
Detailed information about the UEM component definition options.
- [Universal Event Monitor UACL Entries](#) (Chapter 6)  
Detailed information about the Universal Access Control List (UACL) entries available for use with Universal Event Monitor.
- [Additional Information for Universal Event Monitor](#) (Chapter 7)  
Additional technical information used by or specific to UEM.
- [Customer Support](#) (Appendix A)  
Customer support contact information for Universal Event Monitor.

---

# Chapter 1 Overview

---

## 1.1 Overview

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

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

- Universal Event Monitor Manager options
- Universal Event Monitor Server options
- UEMLoad Utility options
- Universal Event Monitor component definition options
- Universal Event Monitor UACL entries
- Additional information

## 1.2 Usage

The configuration options in this document are organized in the following way:

- Options are listed in separate chapters for separate UEM components:
  - UEM Manager
  - UEM Server
  - UEMLoad Utility
- Options are listed alphabetically in each chapter.
- Each option indicates for which operating systems (one or more) that it can be used.
- Each option identifies any operating-specific data or usage relative to that option

---

# Chapter 2

# UEM Manager

# Configuration Options

---

## 2.1 Overview

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

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

Information on how these options are used is documented in the Universal Event Monitor 3.2 User Guide.

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

## 2.2 Configuration Options Information

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

### Description

---

Describes the configuration option and how it is used.

### Usage

---

Provides a table of the following information:

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	<Format / Value>					
Command Line Option, Long Form	<Format / Value>					
Environment Variable	<Format / Value>					
Configuration File Keyword	<Format / Value>					

### Method

Identifies the different methods used to specify Universal Event Monitor Manager configuration options:

- Command Line Option, Short Form
- Command Line Option, Long Form
- Environment Variable
- Configuration File Keyword

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

---

## Syntax

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

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

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

## (Operating System)

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

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

---

## Values

Identifies all possible values for the specified value type.

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

---

## <Additional Information>

Identifies any additional information specific to the option.

## 2.3 Configuration Options List

Table 2.1, below, identifies all UEM Manager configuration options.

Option	Description	Page
BIF_DIRECTORY	Broker Interface Directory that specifies the location of the Universal Broker interface file	25
CODE_PAGE	Code page used for text translation.	26
COMMAND_FILE_ENCRYPTED	Encrypted file that contains some command options.	27
COMMAND_FILE_PLAIN	Plain text file that contains some command options.	28
COMMAND_ID	Identifier, saved by the Universal Broker, that is used to identify the unit of work represented by the UEM Manager and its associated UEM Server component.	29
CONNECT_TIMEOUT	Amount of time that a UEM Manager will wait for a connection to a remote Universal Broker to complete.	30
CTL_SSL_CIPHER_LIST	SSL cipher list for the control session established between the UEM Manager and Server.	32
DNS_EXPAND	Number of IP addresses returned to UEM Manager following a DNS query issued to resolve a host name.	33
ENCRYPTION_KEY	Optional encryption key used to decrypt the encrypted command file specified by the COMMAND_FILE_ENCRYPTED option.	35
EVENT_ID	ID of a stored event definition record.	36
EVENT_STATE	Event state that, when encountered, will result in the execution of the associated event handler.	37
EVENT_TYPE	Type of event to monitor.	38
FILE_SPECIFICATION	Name or pattern of the file whose creation should be detected and tracked for completion.	39
HANDLER_ID	ID of a stored event handler record.	41
HANDLER_OPTIONS	Options that are passed as command line arguments to any process executed on behalf of an event handler.	42
HANDLER_TYPE	Type of process to execute. Used primarily for integration with other Universal Product applications.	44
HELP	Displays command option help	46
HOST_SELECTION	Host in the REMOTE_HOST list that the UEM Manager will choose to begin its attempts to connect to a remote Universal Broker.	47
HOSTNAME_RETRY_COUNT	Maximum number of attempts that will be made to establish a connection with the remote host.	48
INACTIVE_DATE_TIME	Date and time at which the state of the monitored event should be made <b>inactive</b> .	49
INSTALLATION_DIRECTORY	Base directory in which UEM Server is installed	51
LOGIN	Instructs the UEM Server to establish an execution environment for the user account	52
MAX_OCCURRENCE_COUNT	Maximum number of event occurrences to monitor.	53

Option	Description	Page
MAXIMUM_RETURN_CODE	Highest return code that an event handler may exit with to be considered as having executed successfully.	54
MESSAGE_LANGUAGE	Language of messages written.	55
MESSAGE_LEVEL	Level of messages Written.	56
MINIMUM_FILE_SIZE	Smallest size a file may be in order for it to be considered complete.	58
NETWORK_DELAY	Maximum number of seconds to wait for data communications.	59
NLS_DIRECTORY	Directory location of message catalog and code page tables	60
OPTIONS	Values that are passed as command line arguments to a particular handler specified for a given <a href="#">EVENT_STATE</a> .	61
OUTBOUND_IP	Host or IP address to use for all outgoing IP connections.	63
PLF_DIRECTORY	Program Lock File directory that specifies the location of the UEM Manager program lock file	64
POLLING_INTERVAL	Frequency with which the UEM Server will: <ul style="list-style-type: none"> <li>• Detect any new occurrences of the system event</li> <li>• See if the monitored event should be made <b>inactive</b>.</li> </ul>	65
REMOTE_HOST	List of one or more hosts upon which a command may run.	66
REMOTE_PORT	TCP/IP port number on which the Universal Broker is accepting connections.	68
RENAME_FILE	Flag that indicates whether or not a completed file should be renamed.	69
RENAME_FILE_SPECIFICATION	Name or pattern to use when a file is renamed.	70
SYSTEM_ID	Local Universal Broker with which the Universal Event Monitor Manager must register	72
TRACE_FILE_LINES	Maximum number of lines written to a trace file before it wraps around.	73
TRACE_TABLE	Memory trace table specification.	74
TRACKING_INTERVAL	Frequency, in seconds, with which a tracked event occurrence is tested for completeness.	76
USER_COMMAND	Complete path to an application file or remote script that should be executed on behalf of the event handler.	77
USER_ID	ID of a remote user account that the UEM Server uses to establish the security context in which event monitoring is performed.	79
USER_PASSWORD	Password associated with <a href="#">USER_ID</a> .	80
USER_SCRIPT	Complete path to a local script file or DD statement that contains one or more system commands that should be executed on behalf of the event handler.	81
USER_SCRIPT_TYPE	Type of script interpreter used to evaluate and execute the commands contained in <a href="#">USER_SCRIPT</a> .	84
VERSION	Displays program version	85
WAIT	Forces the UEM Manager to wait for the completion of the UEM Server.	86

Table 2.1 UEM Manager Configuration Options



## 2.4 BIF\_DIRECTORY

### Description

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

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	<code>-bif_directory directory</code>			✓		
Environment Variable	<code>UEMBIFDIRECTORY=directory</code>			✓		
Configuration File Keyword	n/a					

### Values

*directory* is the name of the BIF directory.

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

## 2.5 CODE\_PAGE

### Description

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

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	-t <i>codepage</i>			✓	✓	✓
Command Line Option, Long Form	-codepage <i>codepage</i>			✓	✓	✓
Environment Variable	UEMCODEPAGE= <i>codepage</i>			✓	✓	
Configuration File Keyword	codepage <i>codepage</i>			✓	✓	✓

### Value

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

*codepage* references a Universal Translate Table (UTT) file provided with the product (see Section [7.4 UTT Files](#) for information on UTT files). UTT files are used to translate between Unicode and the local single-byte code page. (All UTT files end with an extension of `.utt`.)

#### [Default

The default is different for different operating systems:

- **ISO8859-1 (8-bit ASCII) ASCII-based operating systems**
- **IBM1047 (EBCDIC) Non-OS/400, EBCDIC-based operating system]**

See Section [7.3 Character Code Pages](#) for a complete list of character code pages provided by Stonebranch Inc. for use with Universal Products.

## 2.6 COMMAND\_FILE\_ENCRYPTED

### Description

The `COMMAND_FILE_ENCRYPTED` option specifies a data set (for z/OS) or file (for Windows and UNIX) that contains encrypted values for command parameters. Storing these parameters and their associated values in an encrypted data set / file can be used in situations where it is not desirable to explicitly specify them on the command line.

The parameters contained in the data set / file must be in their respective command line formats.

The data set referenced by the DD statement identified by *ddname* must be encrypted with the Universal Encrypt utility. (For detailed information on the Universal Encrypt utility, see the Universal Products Utilities documentation.)

It is strongly recommended that any command data set / file containing sensitive data (including those that are encrypted) be further protected from unauthorized access using a native operating system security method, such as RACF.

Note: If the data set / file name is specified in this option, it should not be specified additionally with the `COMMAND_FILE_PLAIN` option. If it is, command options contained in the data set / file name identified by the `COMMAND_FILE_PLAIN` will be used.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	-x <i>ddname</i> or <i>filename</i>			✓	✓	✓
Command Line Option, Long Form	-encryptedfile <i>ddname</i> or <i>filename</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

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

## 2.7 COMMAND\_FILE\_PLAIN

### Description

The `COMMAND_FILE_PLAIN` option specifies a data set (for z/OS) or local file (for UNIX and Windows) that contains command parameters. Storing these parameters and their associated values in a data set / file can be used in situations where it is not desirable to specify them explicitly on the command line.

The parameters contained in the data set / file must be in their respective command line formats.

The data set / file specified by this option is a plain text file. It is strongly recommended that this file be further protected from unauthorized access using a native operating system security method, such as RACF.

**Note:** If the data set / file name is specified in this option, it should not be specified additionally with the `COMMAND_FILE_ENCRYPTED` option. If it is, user information contained in the data set / file name file identified in this `COMMAND_FILE_PLAIN` option will be used.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	<code>-f data set or filename</code>			✓	✓	✓
Command Line Option, Long Form	<code>-file data set or filename</code>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

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

## 2.8 COMMAND\_ID

### Description

The `COMMAND_ID` option specifies an identifier that is saved by the Universal Broker and which is used to identify the unit of work represented by the UEM Manager and its associated UEM Server component.

If this option is not used, UEM provides a default command ID. This default depends on how an event's description is supplied to UEM:

- If an event's description is provided via a stored event definition record, the ID of the record, specified by the `EVENT_ID` option, is used for the command ID.
- If an event's description is provided entirely from command line parameters, the default command ID depends on the type of event monitored.
- For events with an event type of `FILE` (see `EVENT_TYPE`), the name of the file specified by the `FILE_SPECIFICATION` option is used.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	<code>-C id</code>			✓	✓	✓
Command Line Option, Long Form	<code>-cmdid id</code>			✓	✓	✓
Environment Variable	<code>UEMCMDID=id</code>			✓	✓	
Configuration File Keyword	<code>n/a</code>					

### Value

*id* is any value.

If *id* contains spaces, it must be enclosed in double ( " ) quotation marks.

#### UNIX and z/OS

If *id* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

## 2.9 CONNECT\_TIMEOUT

### Description

The `CONNECT_TIMEOUT` option specifies how long a Universal Event Monitor Manager will wait for a connection to a remote Universal Broker to complete.

`CONNECT_TIMEOUT` is particularly helpful when more than one host is specified for the `REMOTE_HOST` option. By default, connection timeouts are controlled by the TCP/IP stack. Depending on this value, it may take several minutes to process a list of hosts before a connection actually succeeds. Setting a `CONNECT_TIMEOUT` value allows connection attempts to unreachable Universal Brokers to fail quickly, decreasing the time required to process a list of one or more hosts.

Note: `CONNECT_TIMEOUT` is most beneficial when set to a value that is less than the TCP/IP stack's default timeout, which is implementation dependent. A relatively small `CONNECT_TIMEOUT` value is recommended, to make sure it — and not the TCP/IP default — is applied.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-connect_timeout <i>seconds</i>			✓	✓	✓
Environment Variable	UEMCONNECTTIMEOUT= <i>seconds</i>			✓	✓	
Configuration File Keyword	connect_timeout <i>seconds</i>			✓	✓	✓

---

## Values

---

*seconds* is the time, in seconds, that the UEM Manager will wait for a connection to a Universal Broker to complete. This value applies to each host contained in the resolved, expanded, and scrubbed [REMOTE\\_HOST](#) list.

Valid values for *seconds* are 0 (zero) to 300.

**[Default is 0.]**

(This means that each connection attempt will use the implementation-defined TCP/IP timeout value. This is the behavior of connection attempts prior to version 3.2.0 of Universal Event Monitor.)

If the time specified by *seconds* elapses before a successful connection to a Universal Broker on the specified system is established, the UEM Manager will select the next host in the list. If no more hosts are available, the application will end with an error to indicate that no connection was made.

Note: It is possible for the total time required to attempt connections to all hosts in the [REMOTE\\_HOST](#) list to exceed the number of seconds specified in this option.

## 2.10 CTL\_SSL\_CIPHER\_LIST

### Description

The CTL\_SSL\_CIPHER\_LIST option specifies the acceptable and preferred SSL cipher suites to use for the control session between UEM components. The SSL protocol uses the cipher suites to specify which encryption and message authentication (or message digest) algorithms to use.

The UEM Manager can request one or more SSL ciphers, listed in order of preference. The list is forwarded to the UEM Server, which compares it to a list of SSL ciphers it is capable of accepting, and the first agreed-upon cipher is chosen.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-ctl_ssl_cipher_list <i>cipherlist</i>			✓	✓	✓
Environment Variable	UEMCTLSSLCPHERLIST= <i>cipherlist</i>			✓	✓	
Configuration File Keyword	ctl_ssl_cipher_list <i>cipherlist</i>			✓	✓	✓

### Values

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

Valid *list* values are:

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

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



## 2.11 DNS\_EXPAND

### Description

The DNS\_EXPAND option specifies how many IP addresses are returned to UEM Manager following a DNS query, which is issued to resolve a host name.

If the UEM Manager is configured to expand the results of the query, all IP addresses defined for a particular host name are returned and expanded (in-place) within the list of hosts specified for the REMOTE\_HOST option. Otherwise, only the first host is returned, and no expansion is performed.

For example, if a host list contains six host names, and the name in the 3rd position resolves to five IP addresses, those addresses will occupy positions 3-7 in the expanded list. Subsequent hosts specified by the user will begin at the 8th position in the expanded list.

That is:

- If the REMOTE\_HOST list contains:  
-host *host1,host2,host3,host4,host5,host6*
- And *host3* resolves to:  
*ip3a, ip3b, ip3c, ip3d, and ip3e*
- Then after the other hosts are resolved, the list will be expanded to:  
-host *ip1,ip2,ip3a,ip3b,ip3c,ip3d,ip3e,ip4,ip5,ip6*

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-dns_expand <i>option</i>			✓	✓	✓
Environment Variable	UEMDNSEXPAND= <i>option</i>			✓	✓	
Configuration File Keyword	dns_expand <i>option</i>			✓	✓	✓

---

## Values

---

*option* specifies whether or not DNS query results are expanded.

Valid values for *option* are:

- **yes**  
All IP addresses returned by a DNS for a given query are returned.
- **no**  
Only the first IP address returned by a DNS for a given query is returned.

**[Default is no.]**

## 2.12 ENCRYPTION\_KEY

### Description

The ENCRYPTION\_KEY option specifies the key that is used to decrypt the data set (for z/OS) or file (for Windows and UNIX) specified by the [COMMAND\\_FILE\\_ENCRYPTED](#) option. The key is required only if a key was specified when the command file was encrypted with Universal Encrypt.

A matching key must be provided to decrypt the file; otherwise, the decryption will fail. If no key is specified, a default key established by the UEM Manager is used.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	-K <i>key</i>			✓	✓	✓
Command Line Option, Long Form	-key <i>key</i>			✓	✓	✓
Environment Variable	UEMKEY= <i>key</i>			✓	✓	
Configuration File Keyword	n/a					

### Value

*key* is the key used to decrypt the data set / file.

## 2.13 EVENT\_ID

### Description

The `EVENT_ID` option specifies the ID of an existing record stored in the event definition database. This record contains all of the parameters that are necessary to describe the event to monitor.

If this option is not used, the event, along with any additional parameters that it requires, must be described in the `EVENT_TYPE` option.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-event_id <i>id</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

*id* is the ID of the existing record in the event definition database.

## 2.14 EVENT\_STATE

### Description

The `EVENT_STATE` option marks the beginning of a set of command line options that describes UEM Server's response to an event that enters the specified state.

When this option is used, a corresponding `HANDLER_ID`, `USER_COMMAND`, or `USER_SCRIPT` option also must be provided.

There are three different event states:

- **Triggered**  
An occurrence of a monitored event was completed.
- **Rejected**  
An occurrence of a monitored event was detected, but failed to complete before the date and time specified by the event's `INACTIVE_DATE_TIME` elapsed.
- **Expired**  
Date and time specified by an event's `INACTIVE_DATE_TIME` elapsed, with no occurrence of the monitored event detected.

The UEM Server may repeat the action it takes in response to Triggered and Rejected events if the Server detects more than one occurrence of a monitored event.

By definition, there are no event occurrences for an Expired event. UEM Server executes actions in response to such events just once.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-triggered   -rejected   -expired			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

(There are no values for these parameters.)

## 2.15 EVENT\_TYPE

### Description

The EVENT\_TYPE option specifies the type of an event to monitor.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-event_type <i>type</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

*type* is the event type to monitor.

Table 2.2 identifies valid UEM event types and UEM Manager configuration options associated with these event types.

Note: Currently, there is only one valid UEM event type: *FILE*.

Event Type	Description	Associated Options
<i>FILE</i>	Detects the creation of a file and monitors it, testing it for completeness.	<ul style="list-style-type: none"> <li>FILE_SPECIFICATION *</li> <li>MINIMUM_FILE_SIZE</li> <li>RENAME_FILE</li> <li>RENAME_FILE_SPECIFICATION</li> </ul>

\* FILE\_SPECIFICATION is required; it is the only option required for the *FILE* event type.

Table 2.2 UEM Event Types


## 2.16 FILE\_SPECIFICATION

### Description

The FILE\_SPECIFICATION option specifies the name of the file whose presence should be detected and monitored by UEM.

Additionally, FILE\_SPECIFICATION also can specify an absolute path or a path relative to the working directory defined in the UEM Server's component definition.

Note: FILE\_SPECIFICATION is valid only for events with an [EVENT\\_TYPE](#) of FILE.



**Stoneman's Tip**

If a Windows UEM Manager has issued a request to monitor an event on a UNIX system, and a complete path to the file is specified file specification, a leading space must be added to the path name and it must be enclosed in double quotes. For example, if the UEM Server is asked to look for a file named `file.exe` in `/home/user`, the value specified for *filespecification* must be `" /home/user / file.exe"`. This requirement is a result of the way that command line arguments are read by a Windows UEM Manager, where a forward slash ( / ) also can be used as a command option prefix. The leading space allows the string beginning with the / to be interpreted as a option value.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-filespec <i>filespecification</i>			√	√	√
Environment Variable	n/a					
Configuration File Keyword	n/a					

---

## Value

---

*filespecification* is the name of the path or file to be monitored by UEM.

If the path or file name contains spaces, it must be enclosed in double ( " ) quotation marks.

### UNIX and z/OS

If the path or file name contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

## Wildcards

*filespecification* also can include the following wildcards:

- \* Match zero, one or more characters.
- ? Match zero or one character.



## 2.17 HANDLER\_ID

### Description

The HANDLER\_ID option specifies the actions that should be taken whenever an event occurrence or an event itself enters the corresponding event state (see [EVENT\\_STATE](#)).

If the desired actions are not defined in an existing event handler record, the [USER\\_COMMAND](#) or [USER\\_SCRIPT](#) options can be used to define an event handler entirely from the command line.

USER\_SCRIPT cannot be used with the [USER\\_COMMAND](#) or [USER\\_SCRIPT](#) options for a specified [EVENT\\_STATE](#).

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-handler_id <i>id</i>			√	√	√
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

*id* is the unique identifier of an existing record in the event handler database.

## 2.18 HANDLER\_OPTIONS

### Description

---

The HANDLER\_OPTIONS option specifies a value that UEM Server passes as a command line argument to the process it executes on behalf of an event handler.

Any value specified for HANDLER\_OPTIONS works in conjunction with — not as a replacement of — any value specified by the [OPTIONS](#) option. UEM Server uses the values of both options to construct a list of one or more command line arguments. Any value specified for HANDLER\_OPTIONS always follows the value specified for [OPTIONS](#).

HANDLER\_OPTIONS allows an event definition to control handler process behavior, regardless of [EVENT\\_STATE](#). That is, UEM Server appends the HANDLER\_OPTIONS value to any command it executes on behalf of a monitored event, even if that event specifies three different handlers for its triggered, rejected, and expired states.

For example, you may find it useful to specify a HANDLER\_OPTIONS value that causes all event handler processes executed on a particular event's behalf to redirect output to a single file.

HANDLER\_OPTIONS also provides the ability to customize a stored event handler's behavior.

For example, if an event handler record contains the following information:

```
HANDLER_ID optsexample
USER_SCRIPT "C:\UEMscripts\somescript.bat"
OPTIONS "-dirname C:\Program Files\Universal"
```

And the UEM Manager specifies the following HANDLER\_OPTIONS value:

```
HANDLER_OPTIONS "-dirname C:\Program Files\Universal\ubroker"
```

If **somescript.bat** uses the last value specified to resolve duplicate command line arguments, then **optsexample** executes with **dirname** set to **C:\Program Files\Universal\ubroker**.

## Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-handler_opts <i>options</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

## Value

*options* is a character string that is appended to the command line constructed by Universal Event Monitor in order to execute the event handler process.

If *options* contains spaces, it must be enclosed in ( " ) quotation marks.

### UNIX and z/OS

If *options* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

If quotation marks are to be passed as part of the value (for example, an argument that contains spaces is to be passed to the handler process and treated as a single argument):

- Enclose *options* in apostrophes.
- Use quotation marks to enclose the quoted value.

For example, specifying `-handler_options 'parm1 "parm2a parm2b" parm3'` causes three arguments to be passed to the process executed on behalf of the event handler. The portion of the string enclosed in quotation marks ("parm2a parm2b") is treated as a single value.

## 2.19 HANDLER\_TYPE

### Description

The HANDLER\_TYPE option specifies the type of process that the UEM Server will execute for an event handler.

A UEM Server can execute either a:

- Command specified by the [USER\\_COMMAND](#) option.
- Script specified by the [USER\\_SCRIPT](#) option.

If this option is not used, UEM Server sets the handler type based on which execution option the UEM Manager specifies (that is, [USER\\_COMMAND](#) or [USER\\_SCRIPT](#)).

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-handler_type <i>type</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

---

## Values

---

*type* is the type of process to be executed.

Valid values for *type* are:

- **cmd**  
Instructs the UEM Server to execute the command specified with the [USER\\_COMMAND](#) option.
- **script**  
Instructs the UEM Server to write the statements specified in the data set or DD name referenced by the [USER\\_SCRIPT](#) option to a temporary file on the UEM Server system, and then execute that file as a script.

**[Default is cmd if a [USER\\_COMMAND](#) is provided.]**

**[Default is script if a [USER\\_SCRIPT](#) is provided.]**

## 2.20 HELP

### Description

---

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

### Usage

---

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	-h, -?			✓	✓	✓
Command Line Option, Long Form	-help			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

---

(There are no values for the HELP option.)

## 2.21 HOST\_SELECTION

### Description

The HOST\_SELECTION option specifies which host in the [REMOTE\\_HOST](#) list the UEM Manager will choose to begin its attempts to connect to a remote Universal Broker.

Regardless of how the first host is selected, UEM Manager processes the list sequentially until either a connection to a remote Universal Broker succeeds or all hosts in the list have been tried.

HOST\_SELECTION is ignored if only one host is specified in the [REMOTE\\_HOST](#) list of hosts.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-host_selection <i>option</i>			√	√	√
Environment Variable	UEMHOSTSELECTION= <i>option</i>			√	√	
Configuration File Keyword	host_selection <i>option</i>			√	√	√

### Values

*option* specifies how UEM Manager will select the first entry within a list of hosts specified by the [REMOTE\\_HOST](#) option.

Valid values for *option* are:

- **sequential**  
UEM Manager will select the first host in the list, and then proceed through the hosts in the order in which they appear within the list.
- **random**  
UEM Manager will select any host in the list, in no particular order.

Note: Attempts to connect to a remote Universal Broker begin with this randomly-selected host, and then proceed in a sequential manner, wrapping around the list (if necessary) until the first host selected is reached again.

**[Default is sequential.]**

## 2.22 HOSTNAME\_RETRY\_COUNT

### Description

The HOSTNAME\_RETRY\_COUNT option specifies the number of times that the UEM Manager will attempt to establish a connection with the specified Universal Broker before it fails.

The UEM Manager will sleep for one second between connection attempts.

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

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	N/A					
Command Line Option, Long Form	-hostname_retry_count <i>count</i>			✓	✓	✓
Environment Variable	UEMHOSTNAMERETRYCOUNT= <i>count</i>			✓	✓	
Configuration File Keyword	hostname_retry_count <i>count</i>			✓	✓	✓

### Value

*count* is the number of times that UEM will attempt to establish a connection.

**[Default is 1.]**



## 2.23 INACTIVE\_DATE\_TIME

### Description

The INACTIVE\_DATE\_TIME option specifies the date and time at which the UEM Server will stop testing for new occurrences of the specified system event. This is also the date and time by which any event occurrences detected by the UEM Server must complete in order for those occurrences to move into a **triggered** state.

If a detected event occurrence does not complete by this date and time, UEM will set that occurrence to a **rejected** state. If the specified date and time elapses, and no occurrence of the specified system event was detected by UEM, the event will be set to an **expired** state.

When the inactive date and time elapses, the UEM Server will wait for any executing handler processes to finish, and then end. When the UEM Server ends, the UEM Manager also will end at this time if it was instructed to wait for the Server by the [WAIT](#) option.

If INACTIVE\_DATE\_TIME is not used, default values are set by the UEM Manager:

- Default inactive date is the current date.
- Default inactive time is **23 : 59**.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-inact_date_time yyyy.mm.dd, hh:mm   +mm			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

---

## Value

---

An *absolute* or *relative* date and time can be specified in this option.

- *absolute* date and time is specified in the format `yyyy.mm.dd, hh:mm`, where:
  - `yyyy.mm.dd` is the date.
  - `hh:mm` is the time in 24-hour format.

Either the date or the time can be omitted, in which case a default value is used.

- If the date is omitted, the comma separator must be provided as a placeholder (that is, `,hh:mm`).
- If the time is omitted, the comma is not required.

Note: The maximum valid value for *absolute* date and time is `2038.01.16, 23:59`.

- *relative* date and time is specified using the format `+mm`, where:
  - `mm` is the number of minutes that the event should remain active, starting from the time that the monitoring request was submitted.

## 2.24 INSTALLATION\_DIRECTORY

### Description

The `INSTALLATION_DIRECTORY` option specifies the location in which UEM Manager is installed.

Note: This option is required and cannot be overridden.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	n/a					
Environment Variable	n/a					
Configuration File Keyword	<code>installation_directory directory</code>			✓	✓	

### Values

*directory* is the location in which UEM Manager is installed.

The full path name is required.

#### Defaults

**UNIX**

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

**Windows**

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

## 2.25 LOGIN

### Description

The LOGIN option instructs the UEM Server to establish an execution environment for the user account (specified by the [USER\\_ID](#) option) that resembles the environment that the user would have if the user were actually to log in to the system on which the UEM Server resides.

The differences between a login environment and a non-login environment depends on the UEM Server platform. For example, when a login environment is established on a UNIX system, the user's preferred shell is invoked as a login shell. This results in the execution of commands specified in the system profile and the user's profile.

For more information on the environment established for a given platform, see [Chapter 7 Universal Event Monitor Server](#) in the Universal Event Monitor 3.2.0 User Guide.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	-G <i>option</i>			✓	✓	✓
Command Line Option, Long Form	-login <i>option</i>			✓	✓	✓
Environment Variable	UEMLOGIN= <i>option</i>			✓	✓	
Configuration File Keyword	login <i>option</i>			✓	✓	

### Values

*option* is the instruction for establishing an execution environment.

Valid *option* values are:

- **yes**  
Login environment is established.
- **no**  
Non-login environment is established.

If LOGIN is not used, a default value of **no** (set by the UEM Server configuration) is used.

## 2.26 MAX\_OCCURRENCE\_COUNT

### Description

The MAX\_OCCURRENCE\_COUNT option specifies the maximum number of event occurrences that should be monitored for the current event. Any event occurrences detected by UEM that exceed this number are ignored.

This option is used to force an event that is monitored by a demand-driven UEM Server to become inactive when all occurrences detected for that event are set to a **triggered** state. This makes it possible for an event to become inactive without it having to wait for the date and time, as specified by the [INACTIVE\\_DATE\\_TIME](#) option, to elapse.

If the inactive date and time elapses before UEM detects the specified number of event occurrences, the inactive date and time takes precedence, and the event still will be made inactive. Any tracked occurrences that have not completed by this time will be set to a **rejected** state. If no event occurrences were detected, the event will be set to an **expired** state.

When the event monitored by the demand-driven UEM Server becomes inactive, the UEM Server process will finish when all processes executing on behalf of the specified event handler have finished. The UEM Manager then will wait for the UEM Server to finish (provided it was instructed to do so by the [WAIT](#) option).

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-max_count <i>count</i>			✓	✓	✓
Environment Variable	UEMMAXCNT= <i>count</i>			✓	✓	
Configuration File Keyword	max_count <i>count</i>			✓	✓	✓

### Value

*count* is the number of event occurrences to be monitored.

If *count* = 0, the demand-driven server will remain active until the [INACTIVE\\_DATE\\_TIME](#) option value has been met.

**[Default is 1.]**

## 2.27 MAXIMUM\_RETURN\_CODE

### Description

The MAXIMUM\_RETURN\_CODE option specifies the highest return code value that a process executed on behalf of an event handler can return and still be considered as having executed successfully.

The situation in which any event handler process does not end successfully will be reflected in the return code of a waiting UEM Manager (see the [WAIT](#) option).

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-maxrc <i>returncode</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

*returncode* is the highest return code value that can be returned for a process to be considered successful.

### Default

If this option is not used, a default value of 0 (zero), as set by the UEM Server configuration, is used.

## 2.28 MESSAGE\_LANGUAGE

### Description

The MESSAGE\_LANGUAGE option specifies the message catalog used to format messages issued by the UEM Manager.

#### z/OS

Message catalogs are stored as members in the library referenced by the UNVNLS DD statement in the UEMPRC catalogued procedure. The names of the message catalog members start with UEMMC.

#### UNIX and Windows

Message catalog files are located in the **n1s** subdirectory of the main Universal Products installation directory. All message catalog files end with an extension of **.umc**.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	-L <i>language</i>			✓	✓	✓
Command Line Option, Long Form	-lang <i>language</i>			✓	✓	✓
Environment Variable	UEMLANG= <i>language</i>			✓	✓	
Configuration File Keyword	language <i>language</i>			✓	✓	✓

### Values

*language* is the message catalog to be used.

The first three characters of **language** must match the last three characters of an installed message catalog (minus its extension).

**[Default is ENGLISH, which instructs the UEM Manager to use the UEMMCENG message catalog.]**

## 2.29 MESSAGE\_LEVEL

### Description

The MESSAGE\_LEVEL option controls which messages are issued by a UEM Manager. Optionally, it also allows a date and time stamp to be included with each message.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	-l <i>level</i> [ <i>,time</i> ]			✓	✓	✓
Command Line Option, Long Form	-level <i>level</i> [ <i>,time</i> ]			✓	✓	✓
Environment Variable	UEMLEVEL= <i>level</i> [ <i>,time</i> ]			✓	✓	
Configuration File Keyword	message_level <i>level</i> [ <i>,time</i> ]			✓	✓	✓

### Value

*level* is the type of messages issued.

Valid values for *level* are:

- **trace**  
Issues trace messages to the data set referenced by the UNVTRACE DD statement (for z/OS) or to a file that resides in the UEM Manager working directory named `uem.trc` (for Windows and UNIX).  
Note: **trace** is used for debugging purposes only. It should be used only when requested by Stonebranch Inc. [Customer Support](#).
- **audit**  
Issues audit, informational, warning, and error messages.
- **info**  
Issues informational, warning and error messages.
- **warn**  
Issues warning and error messages.
- **error**  
Issues error messages only.



UNIX and Windows

[Default is warn.]

z/OS

[Default is info.]

*time* specifies whether or not a date and time stamp is to be included with each message.

Valid values for *time* are:

- **time**  
Include a date and time stamp with each message.
- **notime**  
Do not include a date and time stamp with each message.

UNIX and Windows

[Default is notime.]

z/OS

[Default is time.]

## 2.30 MINIMUM\_FILE\_SIZE

### Description

The `MINIMUM_FILE_SIZE` option specifies the smallest allowable size for a file in order for it to be considered complete by UEM Server.

Note: This option is valid only for files being monitored via `EVENT_TYPE` (event type = `FILE`).

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	<code>-min_file_size size[units]</code>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

*size* is the file size. *units* is the unit of storage for that *size*.

The valid values for *units*, and the maximum value that can be specified for *size* for that unit of storage, is:

- **b** bytes (4,294,967,295)
- **k** kilobytes (4,194,304)
- **m** megabytes (4,096)
- **g** gigabytes (4)

If a value for *units* is not specified, the file size is assumed to be in bytes.

### Default

If this option is not used, a default value of 0 (zero), as set by the UEM Server configuration, is used.

## 2.31 NETWORK\_DELAY

### Description

The NETWORK\_DELAY option specifies the maximum acceptable delay in transmitting data between the UEM Manager and UEM Server over the network. If data transmission takes longer than the specified delay, the operation ends with a time-out error.

In this way, NETWORK\_DELAY provides the ability to fine-tune Universal Event Monitor's network protocol.

When a data packet is sent over a TCP/IP network, the time it takes to reach the other end depends on many factors, such as network congestion and bandwidth. If the packet is lost before reaching the other end, the other end may wait indefinitely for the expected data. In order to prevent this situation, NETWORK\_DELAY can be used to tell UEM how long to wait before setting a network time-out condition.

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

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	-d <i>seconds</i>			✓	✓	✓
Command Line Option, Long Form	-delay <i>seconds</i>			✓	✓	✓
Environment Variable	UEMNETWORKDELAY= <i>seconds</i>			✓	✓	
Configuration File Keyword	network_delay <i>seconds</i>			✓	✓	✓

### Value

*seconds* is the number of seconds to wait before setting a network time-out condition.

**[Default is 120.]**

## 2.32 NLS\_DIRECTORY

### Description

The NLS\_DIRECTORY option specifies the name of the directory where the UEM Manager message catalog and code page tables are located.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	n/a					
Environment Variable	n/a					
Configuration File Keyword	nls_directory <i>directory</i>			✓	✓	

### Values

*directory* is the name of the directory where the message catalog and code page tables are located.

Full path names are recommended.

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

### Defaults

#### UNIX

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

#### Windows

[Default is `..\nls.`]

## 2.33 OPTIONS

The OPTIONS option specifies a value that UEM Server uses for command line arguments to an event handler process it executes for a given [EVENT\\_STATE](#).

Any value specified for OPTIONS works in conjunction with — not as a replacement of — any value specified by the [HANDLER\\_OPTIONS](#) option. UEM Server uses the values of both options to construct a list of one or more command line arguments. Any value specified by OPTIONS always precedes the value specified by [HANDLER\\_OPTIONS](#).

OPTIONS allows an event definition to control handler process behavior for a specific [EVENT\\_STATE](#).

For example, given the following UEM Manager command parameters:

```
-event_type file -filespec c:\uemfiles\somefile.txt -inact_date_time +1
-handler_opts ">c:\uemlogs\some\log.txt 2>&1"
-triggered -script c:\uemscripts\somescript.bat
-expired -script c:\uemscripts\somescript.bat -options " -state expired"
```

If one or more occurrences of the event enter a triggered state, UEM Server (on Windows, in this example) executes `c:\uemscripts\somescript.bat`. If the event expires, UEM Server also executes `c:\uemscripts\somescript.bat`, but sets the script's `-state` command line parameter to "expired". UEM Server always redirects the script's stdout and stderr to `c:\uemlogs\some\log.txt`.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	<code>-o string</code>			✓	✓	✓
Command Line Option, Long Form	<code>-options string</code>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

---

## Values

---

*string* is a character string that UEM Server adds to the command line it constructs to execute an event handler process.

If *string* contains spaces, enclose it in double ( " ) quotation marks.

### UNIX and z/OS

If *string* contains spaces, enclose it in single ( ' ) or double ( " ) quotation marks.

To pass quotation marks as part of the value (for example, it is necessary to treat an argument with spaces as a single command argument):

- Enclose options in apostrophes.
- Use quotation marks to enclose the quoted value.

For example, specifying `-options 'parm1 "parm2a parm2b" parm3'` instructs UEM Server to pass three arguments to the process it executes on behalf of the event handler. UEM Server treats the portion of the string enclosed in quotation marks (`"parm2a parm2b"`) as a single value.

## 2.34 OUTBOUND\_IP

### Description

The OUTBOUND\_IP option specifies the host or IP address over which the UEM Manager will establish all outbound network connections.

If this option is not used, the UEM Manager will establish its outbound connections on any interface that is available.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-outboundip <i>host</i>			✓	✓	✓
Environment Variable	UEMOUTBOUNDIP= <i>host</i>			✓	✓	
Configuration File Keyword	outbound_ip <i>host</i>			✓	✓	✓

### Value

*host* is either:

- Host name (for example, **myhost**)
- IP address, in dotted decimal notation (for example, **1 . 2 . 3 . 4**)

**[Default is \*.]**

## 2.35 PLF\_DIRECTORY

### Description

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

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

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-plf_directory <i>directory</i>			✓		
Environment Variable	UEMPLFDIRECTORY= <i>directory</i>			✓		
Configuration File Keyword	n/a					

### Values

*directory* is the name of the PLF directory.

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



## 2.36 POLLING\_INTERVAL

### Description

The POLLING\_INTERVAL option specifies the frequency with which the UEM Server will check to see if an event's inactive date and time has elapsed; if so, the monitored state of the event is set to `inactive`.

If the inactive date and time has not elapsed, and the event is to remain active, UEM also checks for new occurrences of the specified event whenever the polling interval expires.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-polling_int <i>seconds</i>			✓	✓	✓
Environment Variable	UEMPOLLINGINT= <i>seconds</i>				✓	
Configuration File Keyword	polling_int <i>seconds</i>			✓	✓	✓

### Value

*seconds* is the frequency (in seconds) for checking an event's inactive date and time.

### Default

If this option is not used, the UEM Server configuration sets a default value of 30 seconds.

## 2.37 REMOTE\_HOST

### Description

---

The REMOTE\_HOST option specifies a comma-delimited list of one or more hosts upon which a command can run.

Each host in the list can be specified as a host name or an IP address in dotted decimal notation.

**Note:** For purposes of this discussion, even if one host is specified, it is considered a list with a single entry.

If multiple hosts are specified, the UEM Manager will try each one until it successfully connects to a Universal Broker. When a successful connection is established, no more hosts in the list are tried. If the UEM Manager is unable to establish a connection to a Universal Broker on any of the hosts in the list, the UEM Manager will fail.

Each host in the REMOTE\_HOST list also can include a port number or service name that the UEM Manager will use to establish a connection with the Universal Broker on that host. This is useful in situations where the remote Broker is configured to accept incoming connections on a port that is different from the UEM Manager's configured value.

UEM Manager automatically removes any duplicates found in the list of hosts before it makes its first connection attempt. This includes any host names that resolve to the same IP address, or duplicate IP addresses that are added to the list following DNS expansion (see the [DNS\\_EXPAND](#) option).

**Note:** Duplicate IP addresses may appear in the resolved, expanded, scrubbed list of hosts if a different port number is specified for each occurrence of a particular IP address (for example, 192.168.1.1:7887 and 192.168.1.1:7888 are considered distinct entries).

To set configuration options that control selection of the hosts in the REMOTE\_HOST list, see the [HOST\\_SELECTION](#), [CONNECT\\_TIMEOUT](#), and [DNS\\_EXPAND](#) options.

The following text briefly describes each of these options. Each is briefly described below.

#### **HOST\_SELECTION**

UEM Manager uses the [HOST\\_SELECTION](#) option to control which host in the REMOTE\_HOST list is selected first. UEM Manager can be configured to always select the first host or to select with a randomly chosen host as it begins its attempts to connect to a remote Universal Broker. In either case, when a host is selected, UEM Manager processes the list sequentially until either a connection succeeds or all hosts in the list are tried.

(See the [HOST\\_SELECTION](#) option for more information.)

## CONNECT\_TIMEOUT

UEM Manager uses the [CONNECT\\_TIMEOUT](#) option to specify how long it will wait for a connection attempt to succeed before it moves on to the next host in the list.

(See the [CONNECT\\_TIMEOUT](#) option for more information.)

## DNS\_EXPAND

The [DNS\\_EXPAND](#) option controls the number of IP addresses returned when UEM Manager issues a DNS query to resolve a host name. If the UEM Manager is configured to expand the results of the query, all IP addresses defined for a particular host name are returned and expanded (in-place) within the list of hosts. Otherwise, only the first host is returned, and no expansion is performed.

(See the [DNS\\_EXPAND](#) option for more information.)

## Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	-i <i>hostlist</i>			✓	✓	✓
Command Line Option, Long Form	-host <i>hostlist</i>			✓	✓	✓
Environment Variable	UCMDHOST <i>hostlist</i>			✓	✓	
Configuration File Keyword	host <i>hostlist</i>			✓	✓	✓

## Values

*hostlist* is a list of one or more hosts, in the following format:

```
host1[[:port1],host2[:port2],host3[:port3],...hostn[:portn]]
```

In this format:

- *host* is the IP address (in dotted decimal notation) or host name of the system upon which the command may run.
- *port* is an optional port number (or service name), which is necessary only if the remote Universal Broker is accepting incoming connections on a port that is different from the value specified by the [REMOTE\\_PORT](#) option.

Note: The *port* number, when specified, must be separated from the *host* by a colon ( : ).

## 2.38 REMOTE\_PORT

### Description

The REMOTE\_PORT option specifies the TCP port upon which a Universal Broker, on the system specified by the REMOTE\_HOST option, is accepting connections.

The UEM Manager connects to this Universal Broker in order to request the start of a demand-driven UEM Server, which is responsible for monitoring the defined event.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	-p <i>port</i>			✓	✓	✓
Command Line Option, Long Form	-port <i>port</i>			✓	✓	✓
Environment Variable	UEMPORT= <i>port</i>			✓	✓	
Configuration File Keyword	port <i>port</i>			✓	✓	✓

### Value

*port* is the port upon which the Universal Broker is accepting connections.

Valid values for *port* are:

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

**[Default is 7887.]**

## 2.39 RENAME\_FILE

### Description

The RENAME\_FILE option specifies whether or not a file that is being monitored - as specified via an [EVENT\\_TYPE](#) of FILE - should be renamed by UEM when the event occurrence is set to a **triggered** state.

Renaming a file ensures that UEM will not treat a file that it just finished tracking as a new event occurrence.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-rename_file <i>option</i>			√	√	√
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

*option* is the specification for whether or not a file should be renamed.

Valid values for *option* are:

- **yes**  
File is renamed according to the [RENAME\\_FILE\\_SPECIFICATION](#) option.
- **no**  
File is not renamed.

If *option* is **no**, in order to prevent multiple detection of the file by the UEM Server, either:

- Set the value of the [MAX\\_OCCURRENCE\\_COUNT](#) option to **1**.
- Rename the file in a script invoked by an event handler when occurrence is triggered.

### Default

If RENAME\_FILE is not used, a default value of **yes**, as set by the UEM Server configuration, is used.

## 2.40 RENAME\_FILE\_SPECIFICATION

### Description

The RENAME\_FILE\_SPECIFICATION option specifies the file name that the UEM Server should use when both:

- An event occurrence for an event with an **EVENT\_TYPE** of **FILE** is set to a **triggered** state.
- **RENAME\_FILE** option is set to **yes**.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-rename_filespec <i>renamespecification</i>			√	√	√
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

*renamespecification* is the file name to be used by the UEM Server.

Valid values for *renamespecification* can include an absolute path or a path that is relative to the location of the file monitored by UEM. If no path information is provided, the monitored file simply is renamed. The renamed file will reside in the same location as the original file.

If *renamespecification* contains spaces, it must be enclosed in double ( " ) quotation marks.

#### UNIX and z/OS

If *renamespecification* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

## Variables

Variables that UEM will substitute with actual runtime values can be included as part of *renamespecification*. These variables, and the values that UEM will substitute for them, are:

- **\$(compname)**  
Component name
- **\$(compid)**  
Component ID
- **\$(date)**  
Current date, in the format *YYYYMMDD*
- **\$(time)**  
Current time, in the format *HHMMSS*
- **\$(origname)**  
Original base file name, minus its last extension
- **\$(origext)**  
Original file extension
- **\$(seqnum)**  
Sequence number that is initialized to 0 when the demand-driven UEM Server is started, and is then incremented by 1 for each file renamed.

## Default

If `RENAME_FILE_SPECIFICATION` is not used, the following default value, set by the UEM Server configuration, is used:

**\$(compname) . \$(compid) . \$(date) . \$(seqnum)**

## 2.41 SYSTEM\_ID

### Description

The SYSTEM\_ID option identifies the local Universal Broker with which the UEM Manager must register before the Manager performs any request.

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

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-system_id <i>ID</i>					√
Environment Variable	UEMSYSTEMID= <i>ID</i>					√
Configuration File Keyword	n/a					

### Values

*ID* is the system identifier of the local Universal Broker.

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



## 2.42 TRACE\_FILE\_LINES

### Description

The TRACE\_FILE\_LINES option specifies the maximum number of lines to write to a trace file.

A trace file is generated when the MESSAGE\_LEVEL option is set to TRACE. The trace file will wrap around when the number of lines specified by TRACE\_FILE\_LINES has been reached. New trace entries are written at the top of the trace file, just after the trace header lines.

#### z/OS

The trace file is written to the data set referenced by the UNVTRACE DD statement.

Note: This option has no effect if the UNVTRACE DD statement points to a JES SYSOUT file.

#### Windows and UNIX

Trace entries are written to the `uem.trc` file, which resides in the UEM Manager working directory.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-tracefilelines <i>lines</i>			✓	✓	✓
Environment Variable	UEMTRACEFILELINES= <i>lines</i>			✓	✓	
Configuration File Keyword	trace_file_lines <i>lines</i>			✓	✓	✓

### Value

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

When setting this value, allow for an average trace file line size of 50 characters.

**[Default is 50,000.]**

## 2.43 TRACE\_TABLE

### Description

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

Tracing is activated when the MESSAGE\_LEVEL option is set to TRACE.

#### z/OS

The trace table is written to a data set when the program ends under the conditions specified by TRACE\_TABLE.

#### Windows and UNIX

The trace table is written to a file named **uem.trc** when the program ends under the conditions specified by TRACE\_TABLE.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-trace_table size[units],cond			✓	✓	✓
Environment Variable	UEMTRACETABLE=size[units][,cond]			✓	✓	
Configuration File Keyword	trace_table size[units],cond			✓	✓	✓

---

## Values

---

*size* is the size of the trace table.

If *size* is set to 0 (zero), the trace table is not used.

**[Default is 0.]**

*units* is the unit of storage for that *size*.

The valid values for *units*, and the maximum value that can be specified for *size* for that unit of storage, is:

- **b** bytes (2,147,483,647)
- **k** kilobytes (2,097,152)
- **m** megabytes (2,048)
- **g** gigabytes (2)

If a value for *units* is not specified, the file size is assumed to be in bytes.

*cond* specifies the condition under which the trace table is written.

Valid values for *cond* are:

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

**[Default is never.]**

## 2.44 TRACKING\_INTERVAL

### Description

The TRACKING\_INTERVAL option specifies the frequency with which the UEM Server will test for the completion of a tracked event occurrence.

The test for completeness depends upon the type of event that the tracked occurrence represents. For example, an occurrence tracked for an event with an [EVENT\\_TYPE](#) of **FILE** is considered complete when two consecutive checks on a file's size return the same value.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-tracking_int <i>seconds</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

*seconds* is the frequency (in seconds) with which the UEM Server tests for completion of a tracked occurrence.

If TRACKING\_INTERVAL is not used, a default value of 10 seconds, set by the UEM Server configuration, is used.

## 2.45 USER\_COMMAND

### Description

The `USER_COMMAND` option specifies an application or script that the demand-driven UEM Server should execute whenever an event occurrence, or the event itself, enters one of the states specified by the `EVENT_STATE` option.

`USER_COMMAND` cannot be used with the `HANDLER_ID` or `USER_SCRIPT` options for a specified `EVENT_STATE`.

### Examples

The following example is invalid:

```
-triggered -cmd dir -handler_id id001
```

In this example, it is unclear whether the UEM Server should execute the `dir` command or the actions contained in the stored event handler `id001` when an event occurrence is set to a `triggered` state.

The following example is valid:

```
-triggered -handler_id id001 -expired -cmd dir
```

In this example, the `-handler_id` and `-cmd` options are specified for different event states.

- If an event occurrence is set to a `triggered` state, the UEM Server will execute the actions specified in the stored event handler `id001`.
- If the monitored event is set to an `expired` state, the UEM Server will execute the `dir` command.



#### Stoneman's Tip

If a Windows UEM Manager has issued a request to monitor an event on a UNIX system, and the value specified by command contains a complete path, a leading space must be added to the path name, and the path must be enclosed in quotes.

For example, if the UEM Server is to execute an application named `someapp` that resides in `/opt`, the value specified for command must be `" /opt /someapp"`.

This requirement is a result of the way that command line arguments are read by a Windows UEM Manager, where a forward slash ( / ) also can be used as a command option prefix. The leading space allows the string beginning with the / to be interpreted as a option value.

## Usage

---

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-cmd <i>command</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

## Value

---

*command* is the application or script to be executed.

If the application or script is not in the UEM Server home directory, or if it resides outside of the system path, the complete path to the file must be provided.

If *command* contains spaces, it must be enclosed in double ( " ) quotation marks.

### UNIX and z/OS

If *command* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

## 2.46 USER\_ID

### Description

The USER\_ID option specifies the user account in whose security context the demand-driven UEM Server will perform all event monitoring and execute all event handler processes.

A user ID is required only if security is enabled in the UEM Server's configuration.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	-u <i>user</i>			✓	✓	✓
Command Line Option, Long Form	-userid <i>user</i>			✓	✓	✓
Environment Variable	UEMUSERID= <i>user</i>			✓	✓	
Configuration File Keyword	userid <i>user</i>			✓	✓	✓

### Value

*user* is the user account.

Valid values for *user* are valid user accounts that are known to the target system (that is, the system identified by the [REMOTE\\_HOST](#)).

## 2.47 USER\_PASSWORD

### Description

The USER\_PASSWORD option specifies a password for the user account specified by the USER\_ID option.

A password is required only if the UEM Server has enabled its USER\_SECURITY option.

Note: If the UEM Server is executing on a UNIX system, an entry can be added to the Universal Event Monitor Access ACL to not require authentication of certain user accounts. For those user accounts, a user password is not required. However, if the UEM Server is executing on a Windows system, this option is not available, as all user account authentication done by Windows requires a password.

For more information on Universal ACLs, see the section on Universal Access Control List in the Universal Event Monitor 3.2.0 User Guide.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	-w password			✓	✓	✓
Command Line Option, Long Form	-pwd password			✓	✓	✓
Environment Variable	UEMPWD=password			✓	✓	
Configuration File Keyword	password password			✓	✓	✓

### Value

*password* is the password for the user account.



## 2.48 USER\_SCRIPT

### Description

The USER\_SCRIPT option specifies the location of one or more system commands that are executed as a script file by the UEM Server whenever an event occurrence, or the event itself, enters one of the states specified by the [EVENT\\_STATE](#) option.

The system commands (that is, script statements) are read from:

- DD statement referenced by *ddname* (z/OS)
- Local file specified by *filename* (Windows and UNIX)

The statements are sent to the demand-driven UEM Server and stored in a temporary script file. It is this script file that is executed by UEM on behalf of the event handler.



To execute a script that resides on the remote system where an event is being monitored by a demand-driven UEM Server, use the [USER\\_COMMAND](#) option and give the script file as the value for command.

#### Stoneman's Tip

The script's contents can include any statement that is valid for the system on which they will execute (that is, the system identified by the [REMOTE\\_HOST](#) option). The maximum size of a script file is approximately 64,000 characters.

USER\_SCRIPT cannot be used with the [HANDLER\\_ID](#) or [USER\\_COMMAND](#) options for a specified [EVENT\\_STATE](#).

## Examples

### z/OS

The following example is invalid:

```
-triggered -script myscript -handler_id id001
```

In this example, it is unclear whether the contents of the script contained in the data set referenced by the DD name `myscript`, or the actions contained in the stored event handler `id001`, should be executed whenever an event occurrence is set to a `triggered` state.

The following example is valid:

```
-triggered -handler_id id001 -expired -script myscript
```

In this example, the `-handler_id` and `-script` options are specified for different event states.

- If an event occurrence is set to a `triggered` state, the UEM Server will execute the actions specified in the stored event handler `id001`.
- If the monitored event is set to an `expired` state, the UEM Server will execute the contents of the script referenced by the DD statement `myscript`.

### UNIX and Windows

The following example is invalid:

```
-triggered -script c:\myscript.txt -handler_id id001
```

In this example, it is unclear whether the contents of the script contained in the local file `c:\myscript.txt` or the actions contained in the stored event handler `id001` should be executed whenever an event occurrence is set to a `triggered` state.

The following example is valid:

```
-triggered -handler_id id001 -expired -script c:\myscript.txt
```

In this example, the `-handler_id` and `-script` options are specified for different event states.

- If an event occurrence is set to a `triggered` state, the demand-driven UEM Server will execute the actions specified in the stored event handler `id001`.
- If the monitored event is set to an `expired` state, the UEM Server will execute the contents of the script contained in the file `c:\myscript.txt`.

## Usage

---

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-script <i>ddname</i> or <i>filename</i>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

## Value

---

*ddname* (for z/OS) and *filename* (for Windows and UNIX) is the location of the commands to be executed as a script file.

## 2.49 USER\_SCRIPT\_TYPE

### Description

The `USER_SCRIPT_TYPE` option describes the type of statements contained within the script specified by the `USER_SCRIPT` option.

Note: Currently, `USER_SCRIPT_TYPE` applies only to scripts executed by UEM Servers running on a Windows system.

When the UEM Server writes the script statements to a temporary script file, the value specified by `USER_SCRIPT_TYPE` is used as that file's extension. On the target Windows system, an association between files with that file extension and an application responsible for opening files with that extension must exist prior to any attempt by a UEM Server to execute the script. Otherwise, execution of the script will fail.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	<code>-script_type type</code>			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

*type* is the type of statement in the script specified by `USER_SCRIPT`.

If this option is not used, a default value of `bat`, set by the UEM Server configuration, is used.

## 2.50 VERSION

### Description

---

The VERSION option instructs UEM Manager to display program version and copyright information.

### Usage

---

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	-v			✓	✓	✓
Command Line Option, Long Form	-version			✓	✓	✓
Environment Variable	n/a					
Configuration File Keyword	n/a					

### Value

---

(There are no values for this option.)

## 2.51 WAIT

### Description

The WAIT option instructs UEM Manager to execute until it receives notification that the demand-driven UEM Server has ended.

A demand-driven Server ends when both of the following occur:

- Event that it is monitoring is set to an inactive state.
- Any handler processes executed by the Server have completed.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Command Line Option, Short Form	n/a					
Command Line Option, Long Form	-wait <i>option</i>			✓	✓	✓
Environment Variable	UEMWAIT= <i>option</i>			✓	✓	
Configuration File Keyword	wait <i>option</i>			✓	✓	✓

### Value

*option* is the instruction for whether or not to wait for the UEM Server to finish executing.

Valid values for *option* are:

- **yes**  
UEM Manager will wait for the demand-driven UEM Server to finish executing.
- **no**  
UEM Manager will end as soon as it receives notification from the demand-driven UEM Server that the event parameters are valid and monitoring of the event has begun. All additional event monitoring activity will be reported only in the Universal Broker log.

**[Default is yes.]**

---

# Chapter 3

## UEM Server

# Configuration Options

---

### 3.1 Overview

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

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

Information on how these options are used is documented in the Universal Event Monitor 3.2 User Guide.

Section [3.2 Configuration Options Information](#) provides a guideline for understanding the information presented on each option.

## 3.2 Configuration Options Information

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

### Description

---

Describes the option and how it is used.

### Usage

---

Provides a table of the following information:

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<Format / Value>					
UEM Load Override	<Format / Value>					
UEM Manager Override	<Format / Value>					

### Method

Identifies the different methods used to specify Universal Event Monitor Server configuration options:

- Configuration File Keyword
- UEMLoad Override
- UEM Manager Override

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



---

## Syntax

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

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

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

## (Operating System)

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

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

---

## Values

Identifies all possible values for the specified value type.

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

---

## <Additional Information>

Identifies any additional information specific to the option.

## 3.3 Configuration Options List

Table 3.1 UEM Server Configuration Options identifies all UEM Server configuration options.

Option	Description	Page
ASSIGNED_COMPONENT_NAME	UEM Server component to which a stored event definition is assigned.	92
CODE_PAGE	Code page used for text translation	93
EXPIRED_HANDLER_ID	ID of a stored event handler record that should be used whenever an event is set to an expired state.	94
INSTALLATION_DIRECTORY	Base directory in which UEM Server is installed	95
INTERACT_WITH_DESKTOP	Specifies whether or not the desktop of the current interactive logon session is accessible to the handler process.	96
LOGIN	Specifies whether or not to load a user's profile and environment when a handler process is executed.	98
LOGON_METHOD	Specifies how a user account is logged on when the environment for execution of a handler process is established.	100
MAXIMUM_RETURN_CODE	Highest return code than an event handler may exit with to still be considered as having executed successfully.	101
MESSAGE_LEVEL	Level of messages printed.	102
MINIMUM_FILE_SIZE	Smallest size a file may be in order for it to be considered complete.	103
NLS_DIRECTORY	Directory location of message catalog and code page tables	104
POLLING_INTERVAL	Frequency with which a UEM Server will detect new occurrences of a system event. The UEM Server will also check at this time see if a monitored event should be made inactive.	105
REJECTED_HANDLER_ID	ID of a stored event handler record that should be used whenever an event occurrence is set to a rejected state.	106
RENAME_FILE	Flag that indicates whether or not a completed file should be renamed.	107
RENAME_FILE_SPECIFICATION	Name or pattern to use when a file is renamed.	108
SHELL	Default shell interpreter for event handler processes executed as scripts.	110
TRACE_DIRECTORY	Specifies the directory in which trace files are written.	111
TRACE_FILE_LINES	Maximum number of lines written to a trace file before it wraps around.	112
TRACE_TABLE	Memory trace table specification.	113
TRACKING_INTERVAL	Frequency, in seconds, with which a tracked event occurrence is tested for completeness.	115
TRIGGERED_HANDLER_ID	ID of a stored event handler record that should be used whenever an event occurrence is set to a triggered state.	116

Option	Description	Page
USER_SCRIPT_TYPE	Type of script interpreter used to evaluate and execute a script on behalf of an event handler.	<a href="#">117</a>
USER_SECURITY	Specifies whether or not user account authorization is required to monitor an event and/or execute a handler process.	<a href="#">118</a>

Table 3.1 UEM Server Configuration Options

## 3.4 ASSIGNED\_COMPONENT\_NAME

### Description

---

The ASSIGNED\_COMPONENT\_NAME option specifies the event-driven UEM Server component to which new event definitions will be assigned if no such value is provided when the record is added with the UEMLoad utility.

### Usage

---

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	comp_name <i>compname</i>				√	
UEM Load Override	-comp_name <i>compname</i>				√	
UEM Manager Override	n/a					

### Value

---

*compname* is the name of the event-driven UEM Server.

**[Default is *uems*.]**

## 3.5 CODE\_PAGE

### Description

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

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	codepage <i>codepage</i>			✓	✓	
UEM Load Override	n/a					
UEM Manager Override	n/a					

### Value

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

*codepage* references a Universal Translate Table (UTT) file provided with the product (see Section [7.4 UTT Files](#) for information on UTT files). UTT files are used to translate between Unicode and the local single-byte code page. (All UTT files end with an extension of `.utt`.)

#### [Default

The default code page is different for different operating systems:

- **ISO8859-1 (8-bit ASCII)** ASCII-based operating systems
- **IBM1047 (EBCDIC)** Non-OS/400, EBCDIC-based operating system]

See Section [7.3 Character Code Pages](#) for a complete list of character code pages provided by Stonebranch Inc. for use with Universal Products.

## 3.6 EXPIRED\_HANDLER\_ID

### Description

The EXPIRED\_HANDLER\_ID option specifies the ID of a default stored event handler record that should be used when an event expires.

The UEM Server stores this ID in records that are added to the event definition database if it is not overridden by the specified UEMLoad utility command option.

This ID also is used when both of the following occur:

1. Event monitored by a demand-driven Server expires.
2. Specified UEM Manager command option was omitted.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	expired_id <i>id</i>			✓	✓	
UEM Load Override	-expired_id <i>id</i>			✓	✓	
UEM Manager Override	-expired -handler_id <i>id</i>			✓	✓	

### Value

*id* is the ID to be used.

**[There are no defaults.]**

## 3.7 INSTALLATION\_DIRECTORY

### Description

The INSTALLATION\_DIRECTORY option specifies the full path name of the directory in which the Universal Event Monitor Server is installed.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	installation_directory <i>directory</i>			✓	✓	
UEM Load Override	n/a					
UEM Manager Override	n/a					

### Value

*directory* is the full path name of the directory.

#### Defaults

##### UNIX

[Default is /opt/universal/uemsrv.]

##### Windows

[Default is c:\Program Files\Universal\uemsrv.]

## 3.8 INTERACT\_WITH\_DESKTOP

### Description

The `INTERACT_WITH_DESKTOP` option specifies whether or not event handler processes are allowed to interact with the current console logon session.

If event handler processes are allowed to interact, they are run in a context that permits the current interactive console session to interact with them. This interaction can go both ways, as the process may invoke system functions that access desktop elements (for example, Windows, menus, and buttons) associated with the session. This is considered a security risk, in that it creates an opportunity for malicious code to hijack the desktop. If the security context of the interactive session is higher than that of the process, the process could invoke code using an elevated security context.

If event handler processes are not allowed to interact, they execute in a context that is isolated from the current interactive logon session. Unless event handler processes requiring user interaction are executed, this is the recommended value.



#### Stoneman's Tip

This option applies only when the `LOGON_METHOD` option is set to `INTERACTIVE`. If user accounts are authenticated using the `BATCH` logon method, the established security context already disallows all interaction with the desktop.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<code>interact_with_desktop option</code>				√	
UEM Load Override	<code>n/a</code>					
UEM Manager Override	<code>n/a</code>					



---

## Value

---

*option* is the specification for whether or not the event handler processes are allowed to interact:

Valid values for *option* are:

- **yes**  
Allow event handler processes to interact with the current console logon session.
- **no**  
Do not allow event handler processes to interact with the current console logon session.

**[Default is no.]**

## 3.9 LOGIN

### Description

---

The LOGIN option instructs the UEM Server whether or not to establish an environment, when executing a handler process, that resembles the environment that the user account (in whose security context the process executes) would have were the user actually to log in to the system on which the UEM Server resides.

#### Windows

If the environment is established, the user's environment block is loaded and the user's registry hive is mapped to HKEY\_CURRENT\_USER. This makes the user's profile available to the handler process.

#### UNIX

If the environment is established, the SHELL used to execute an event handler process is invoked as a login shell, which reads and executes commands from the system and user profiles. The profiles that are read and executed depends on the type of shell invoked.

#### AIX

The AIX platform provides a `/etc/environment` that allows global environmental variables to be exported for all users of a given machine. UEM Server adds the environment variables defined in this file to the user's login environment.

The order in which it is processed by UEM Server is slightly different than the order used by the AIX login process. The AIX login process reads and processes the following files in the order listed:

1. `/etc/profile`
2. `/etc/environment`
3. `$HOME/.profile`
4. `$HOME/.env`

The LOGIN option directs a UEM Server to use the login shell to execute the event handler process. The AIX login shell does not process the `/etc/environment` file, so UEM Server processes the files in the following order:

1. `/etc/environment`
2. `/etc/profile`
3. `$HOME/.profile`
4. `$HOME/.env`

Note: If the UEM Server `USER_SECURITY` option is set to `none`, the LOGIN option is ignored.

## Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	login <i>option</i>			√	√	
UEM Load Override	n/a					
UEM Manager Override	-login <i>option</i>			√	√	

## Values

*option* is the indication for whether or not to establish the environment.

### Windows

Valid value for *option* are:

- **yes**  
Login environment is established.
- **no**  
Non-login environment is established.

[Default is no.]

### UNIX

Valid values for *option* are:

- **yes**  
Invoke the shell as a login shell.
- **no**  
Do not invoke the shell as a login shell.

[Default is no.]

## 3.10 LOGON\_METHOD

### Description

The LOGON\_METHOD option specifies the logon method used when a user is logged on by UEM Server prior to execution of an event handler process.

Note: If the UEM Server's **USER\_SECURITY** option is set to *none*, the LOGON\_METHOD option is ignored.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	logon <i>method</i>				√	
UEM Load Override	n/a					
UEM Manager Override	n/a					

### Value

*method* is the logon method used.

Valid values for *method* are:

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

**[Default is interactive.]**

## 3.11 MAXIMUM\_RETURN\_CODE

### Description

---

The MAXIMUM\_RETURN\_CODE option specifies the highest return code value that a process, executed on behalf of an event handler, can return still to be considered as having executed successfully.

### Usage

---

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<code>maxrc returncode</code>			✓	✓	
UEM Load Override	<code>-maxrc returncode</code>			✓	✓	
UEM Manager Override	<code>-maxrc returncode</code>			✓	✓	

### Value

---

*returncode* is the return code value.

**[Default is 0 (zero).]**

## 3.12 MESSAGE\_LEVEL

### Description

The MESSAGE\_LEVEL option controls which messages are issued by a UEM Server.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	message_level <i>level</i>			✓	✓	
UEM Load Override	n/a					
UEM Manager Override	n/a					

### Value

*level* is the level of messages to be issued by the UEM Server.

Valid values for *level* are:

- **trace**  
Issues trace messages to a file named `uemsrv-nnnnnnnnnn-1.trc`, where `nnnnnnnnnn` is the 10-digit component ID.

#### Windows

The trace file is located in the `.\Universal\UEMSrv` directory.

#### UNIX

The trace file is located in the `./Universal/trace` directory.

Note: Use **trace** only for debugging, and only when requested by Stonebranch Inc. [Customer Support](#).

- **audit**  
Issues audit, informational, warning, and error messages.
- **info**  
Issues informational, warning and error messages.
- **warn**  
Issues warning and error messages.
- **error**  
Issues error messages only.

**[Default is info.]**

## 3.13 MINIMUM\_FILE\_SIZE

### Description

The MINIMUM\_FILE\_SIZE option specifies the smallest size that a file - tracked for an event with an event type of FILE - can be in order to be considered complete.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	min_file_size size[units]					
UEM Load Override	-min_file_size size[units]			✓	✓	
UEM Manager Override	-min_file_size size[units]			✓	✓	

### Value

*size* is the file size. *units* is the unit of storage for that *size*.

The valid values for *units*, and the maximum value that can be specified for *size* for that unit of storage, is:

- **b** bytes (4,294,967,295)
- **k** kilobytes (4,194,304)
- **m** megabytes (4,096)
- **g** gigabytes (4)

If a value for *units* is not specified, the file size is assumed to be in bytes.

**[Default is 0 (zero).]**

## 3.14 NLS\_DIRECTORY

### Description

The NLS\_DIRECTORY option specifies the name of the directory where the UEM Server message catalog and code page tables are located.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	nls_directory <i>directory</i>			✓	✓	
UEM Load Override	n/a					
UEM Manager Override	n/a					

### Values

*directory* is the name of the directory where the message catalog and code page tables are located.

Full path names are recommended.

Relative path names are relative to the `universal1` installation directory.

#### Defaults

##### UNIX

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

##### Windows

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



## 3.15 POLLING\_INTERVAL

### Description

---

The POLLING\_INTERVAL option specifies the frequency with which the Universal Event Monitor Server will check to see if an event's inactive date and time has elapsed.

If the inactive date and time has not elapsed, and the monitored event is to remain active, the UEM Server then will check for new occurrences of the event.

### Usage

---

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	polling_int secs			✓	✓	
UEM Load Override	n/a					
UEM Manager Override	-polling_int secs			✓	✓	

### Value

---

secs is the frequency (number of seconds) with which the UEM Server checks.

**[Default is 30.]**

## 3.16 REJECTED\_HANDLER\_ID

### Description

The REJECTED\_HANDLER\_ID option specifies the ID of a default stored event handler record that should be used when an event occurrence is rejected.

The UEM Server stores this value in records added to the event definition database if it is not overridden by the specified UEMLoad utility command option. This ID also is used for event occurrences that are rejected by a demand-driven Server if the specified UEM Manager command option was omitted.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	rejected_id <i>id</i>			✓	✓	
UEM Load Override	-rejected_id <i>id</i>			✓	✓	
UEM Manager Override	-rejected_id <i>id</i>			✓	✓	

### Value

*id* is the ID to be used.

**[There is no default.]**

## 3.17 RENAME\_FILE

### Description

The RENAME\_FILE option specifies whether or not a file tracked for an event with an event type of **FILE** should be renamed by the UEM Server when the event occurrence is set to a **triggered** state.

Renaming a file ensures that UEM won't treat a file that it just finished tracking as a new event occurrence.

Note: This option is used only for events with an event type of **FILE**.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<code>rename_file option</code>			✓	✓	
UEM Load Override	<code>-rename_file option</code>			✓	✓	
UEM Manager Override	<code>-rename_file option</code>			✓	✓	

### Value

*option* is the specification for whether or not the file should be renamed.

Valid values for *option* are:

- **yes**  
File is renamed according to the [RENAME\\_FILE\\_SPECIFICATION](#) option.
- **no**  
File is not renamed.

If *option* is **no**, you also must either:

- Set the value of the UEM Manager [MAX\\_OCCURRENCE\\_COUNT](#) option to 1.
- Rename the file in a script invoked by an event handler when the occurrence is triggered. This will prevent multiple detection of the file by the UEM Server.

**[Default is yes.]**

# 3.18 RENAME\_FILE\_SPECIFICATION

## Description

The RENAME\_FILE\_SPECIFICATION option specifies the file name that a Universal Event Monitor Server should use when both of the following occur:

- Event occurrence for an event with an event type of FILE is set to a triggered state.
- [RENAME\\_FILE](#) option is set to **yes**.

## Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<code>rename_filespec <i>renamespecification</i></code>			✓	✓	
UEM Load Override	<code>-rename_filespec <i>renamespecification</i></code>			✓	✓	
UEM Manager Override	<code>-rename_filespec <i>renamespecification</i></code>			✓	✓	

## Values

---

*renamespecification* is the file name to be used. It can include an absolute path or a path that is relative to the location of the file monitored by UEM. If no path information is provided, the monitored file is simply renamed, and the renamed file will reside in the same location as the original file.

Variables that UEM will substitute with actual runtime values also can be included as part of *renamespecification*. These variables, and the values that UEM will substitute for them, are listed below.

- **\$(compname)**  
Component name
- **\$(compid)**  
Component ID
- **\$(date)**  
Current date, in the format *YYYYMMDD*
- **\$(time)**  
Current time, in the format *HHMMSS*
- **\$(origname)**  
Original base file name, minus its last extension
- **\$(origext)**  
Original file extension
- **\$(seqnum)**  
A sequence number that is initialized to 0 when a UEM Server is started, and is then incremented by 1 for each file renamed.

If *renamespecification* contains spaces, it must be enclosed in double ( " ) quotation marks.

### UNIX

If *renamespecification* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

**[Default is *\$(compname).\$(compid).\$(date).\$(seqnum).*]**

## 3.19 SHELL

### Description

---

The SHELL option specifies the UNIX command shell that should be used to execute an event handler process.

This command shell must be specified with an argument that directs it to execute the given command and then exit.

### Usage

---

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	shell <i>shell</i>			✓		
UEM Load Override	n/a					
UEM Manager Override	n/a					

### Value

---

*shell* is the name of the UNIX command shell.

**[Default is /bin/sh -c.]**

## 3.20 TRACE\_DIRECTORY

### Description

The TRACE\_DIRECTORY option specifies the location where trace files generated by UEM Server are stored.

Trace files are generated when the MESSAGE\_LEVEL option is set to TRACE.

A complete path, or a path relative to the UEM Server installation directory (for example, C:\Program Files\Universal\UEMSrv) can be specified.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	trace_directory <i>directory</i>			✓	✓	
UEM Load Override	n/a					
UEM Manager Override	n/a					

### Value

*directory* is the location where trace files are stored.

#### Windows

[Default is the UEM Server installation directory.]

#### UNIX

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

## 3.21 TRACE\_FILE\_LINES

### Description

The TRACE\_FILE\_LINES option specifies the maximum number of lines to write to a trace file.

A trace file is generated when the MESSAGE\_LEVEL option is set to TRACE. In this situation, the trace file will wrap around when the number of lines specified by this TRACE\_FILE\_LINES option has been reached.

New trace entries are written at the top of the trace file, just after the trace header lines. The trace entries are written to a file named `uemsrv-nnnnnnnnnn-1.trc` file, where *nnnnnnnnnn* is the 10-digit component ID assigned to the UEM Server process.

#### Windows

The trace file is located in the `.\Universal\UEMSrv` directory.

#### UNIX

The trace file is located in the `./Universal/trace` directory.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<code>trace_file_lines <i>lines</i></code>			✓	✓	
UEM Load Override	<code>n/a</code>					
UEM Manager Override	<code>n/a</code>					

### Value

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

When setting this value, allow for an average trace file line size of 50 characters.

**[Default is 50,000.]**



## 3.22 TRACE\_TABLE

### Description

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

Tracing is activated when the MESSAGE\_LEVEL option is set to TRACE. The trace table then is printed when the program ends under the conditions specified by this TRACE\_TABLE option.

The trace file is named `uemsrv-nnnnnnnnnn-1.trc`, where `nnnnnnnnnn` represents the 10-digit component ID assigned to the UEM Server process when it is started by the Universal Broker.

#### Windows

The trace file is located in the `.\Universal\UEMSrv` directory.

#### UNIX

The trace file is located in the `./Universal/trace` directory.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<code>trace_table size[units][,cond]</code>			✓	✓	
UEM Load Override	n/a					
UEM Manager Override	n/a					

---

## Value

---

*size* is the size of the trace table.

If *size* is set to 0 (zero), the trace table is not used.

*units* is the unit of storage for that *size*.

The valid values for *units*, and the maximum value that can be specified for *size* for that unit of storage, is:

- **b** bytes (2,147,483,647)
- **k** kilobytes (2,097,152)
- **m** megabytes (2,048)
- **g** gigabytes (2)

If a value for *units* is not specified, the file size is assumed to be in bytes.

**[Default is 0.]**

*cond* specifies the condition under which the trace table is written.

Valid values for *cond* are:

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

**[Default is never.]**

## 3.23 TRACKING\_INTERVAL

### Description

---

The TRACKING\_INTERVAL option specifies the frequency with which a Universal Event Monitor Server tests for the completion of a tracked event occurrence.

The test for completeness depends upon the type of event that the tracked occurrence represents. For example, an occurrence tracked for an event with an event type of **FILE** is considered complete when two consecutive checks on a file's size return the same value.

### Usage

---

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	tracking_int secs			✓	✓	
UEM Load Override	-tracking_int secs			✓	✓	
UEM Manager Override	-tracking_int secs			✓	✓	

### Value

---

secs is the frequency (number of seconds) with which the UEM Server tests for completion of an event occurrence.

**[Default is 10.]**

## 3.24 TRIGGERED\_HANDLER\_ID

### Description

The TRIGGERED\_HANDLER\_ID option specifies the ID of a default stored event handler record that should be used when an event occurrence is triggered.

The UEM Server stores this ID in records that are added to the event definition database if it is not overridden by the specified UEMLoad utility command option.

The ID also is used for event occurrences triggered by a demand-driven UEM Server if the specified UEM Manager command option was omitted.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	triggered_id <i>id</i>			✓	✓	
UEM Load Override	-triggered_id <i>id</i>			✓	✓	
UEM Manager Override	-triggered_id <i>id</i>			✓	✓	

### Value

*id* is the ID to be used.

**[There is no default.]**

## 3.25 USER\_SCRIPT\_TYPE

### Description

The USER\_SCRIPT\_TYPE option specifies the command processor that UEM Server uses to execute an event handler process on Windows.

The value specified for USER\_SCRIPT\_TYPE is a file extension. To execute the event handler process, UEM Server invokes the default program or command interpreter assigned to open files with that extension.

For example:

- To have UEM Server execute a script containing Windows command statements, set USER\_SCRIPT\_TYPE to **bat**.
- To have UEM Server execute a script with Perl statements, set USER\_SCRIPT\_TYPE to an extension (for example, **p1**) that, by default, invokes a Perl interpreter when opened.

If no association exists between a USER\_SCRIPT\_TYPE's corresponding file extension and a default program, the script fails.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	<code>script_type type</code>				√	
UEM Load Override	<code>-script_type type</code>				√	
UEM Manager Override	<code>-script_type type</code>			√	√	√

### Value

*type* is a file extension that describes the script's type.

**[Default is *bat*.]**

## 3.26 USER\_SECURITY

### Description

The USER\_SECURITY option specifies user security usage for the UEM Server.

The way in which this option is used depends on whether a UEM Server is running in demand-driven or event-driven mode.

For a demand-driven UEM Server, enabling this option means an authenticated local user account is required to start the Server and monitor an event. All event monitoring activity, including execution of event handler processes, is done in the security context of this user.

For an event-driven UEM Server, enabling this option means that before an event handler process can be executed, a local user account and password must be stored in the event handler record (remember that an event-driven Server relies completely upon stored event definition and event handler records for its input). If the stored account information is valid, any processes executed on behalf of the event handler will be done so in the security context of that account.

#### Windows

This option should be set only via the Universal Configuration Manager; check the box labelled “Require user ID and password to start process” on the Server Options page of the Universal Event Monitor Server configuration control. (See the Stored Configuration Options section in the UEM 3.20 User Guide for more information on the Universal Configuration Manager.)

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Configuration File Keyword	security <i>method</i>			√	√	
UEM Load Override	n/a					
UEM Manager Override	n/a					

## Values

---

*method* is the user authentication method to be used.

### Windows

Valid values for *method* are:

- **default**  
Windows authenticates provided account information, requiring a user ID and password to do so.
- **none**  
Disable user security.

### UNIX

Valid values for *method* are:

- **default**  
Default UNIX user authentication method, which relies upon the `/etc/passwd` file.
- **pam**  
Pluggable Authentication Module interface
- **trusted**  
HP Trusted Security authentication
- **none**  
Disable user security.

---

# Chapter 4

## UEMLoad Utility

# Configuration Options

---

## 4.1 Overview

This chapter provides detailed information on the configuration options available for use with the Universal Event Monitor Load Utility.

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

Information on how these options are used is documented in the Universal Event Monitor 3.2 User Guide.

Section [4.2 Configuration Options Information](#) provides a guideline for understanding the information presented on each option.



## 4.2 Configuration Options Information

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

### Description

---

Describes the option and how it is used.

### Usage

---

Provides a table of the following information:

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<Format / Value>					
Command Line, Long Form	<Format / Value>					
Definition File Keyword	<Format / Value>					

#### Specification Method

List of all possible methods for specifying an option:

- Command Line, Short Form
- Command Line, Long Form
- Definition File Keyword

#### Parameter / Value

Syntax for specifying the option:

- Parameter           Parameter syntax for the corresponding Specification Method.
- Value               Specify alphanumeric value for that parameter

#### (Operating System)

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

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

---

## Values

---

Identifies all possible values for the option.

### Default

Identifies default information regarding the option or its values.

Defaults for option values are identified in **[bracketed bold type]**.

## <Additional Information>

---

Identifies any additional information specific to that option.

## 4.3 Configuration Options List

Table 4.1 UEMLoad Utility - Configuration Options identifies all UEMLoad Utility configuration options.

Option	Description	Page
ACTION	Requested database operation.	125
ACTIVE_DATE_TIME	Date and time at which UEM will begin monitoring an event definition.	127
ASSIGNED_COMPONENT_NAME	Event-driven UEM Server responsible for monitoring the event.	128
BROKER_PORT	Port on which a local Universal Broker is accepting incoming connections.	130
COMMAND_ID	Unique command ID associated with the database request.	131
DEFINITION_FILE	Name of a file that contains event definition and/or event handler parameters.	132
ENCRYPTION_KEY	Key that was used with Universal Encrypt to encrypt the encrypted user file.	133
EVENT_ID	Identifier that uniquely identifies an event definition record.	134
EVENT_STATE	Event definitions that should be processed or ignored by UEM.	135
EVENT_TYPE	Type of system event represented by the event definition record.	136
EXPIRED_HANDLER_ID	ID of an event handler record that UEM will execute when an event expires.	137
FILE_SPECIFICATION	Name of a file to monitor.	138
HANDLER_ID	Identifier that uniquely identifies an event handler record.	139
HANDLER_OPTIONS	String that will be passed as command line arguments to the event handler executed by UEM.	141
HANDLER_TYPE	Type of process executed for the event handler, based on the contents of the USER_COMMAND and USER_SCRIPT parameters	143
HELP	Displays command line help.	145
INACTIVE_DATE_TIME	Date and time at which UEM will stop monitoring an event definition.	146
MAXIMUM_RETURN_CODE	Highest value with which a handler may exit to still be considered as having executed successfully.	148
MESSAGE_LEVEL	Sets the level of messages reported by UEMLoad.	149
MINIMUM_FILE_SIZE	Size a file must be in order to be considered complete by UEM.	150
OPTIONS	String value that is added to the command line UEM Server builds in order to execute an event handler process. The event definition's HANDLER_OPTIONS field is used for the same purpose, but is added to the command string after this field, in order to customize process behavior for that event.	151

Option	Description	Page
REJECTED_HANDLER_ID	ID of an event handler record that UEM will execute when an event occurrence is rejected.	153
RENAME_FILE	Specifies whether or not UEM should rename a monitored file when an event occurrence is triggered.	154
RENAME_FILE_SPECIFICATION	Specifies how a file should be renamed when an event occurrence is triggered.	155
TRACKING_INTERVAL	Frequency with which UEM will test for the completion of an event occurrence.	157
TRIGGERED_HANDLER_ID	ID of an event handler record that UEM will execute when an event occurrence is triggered.	158
USER_COMMAND	Command to execute on behalf of the event handler.	159
USER_FILE_ENCRYPTED	Complete path to a file encrypted with Universal Encrypt. If this file contains a user ID and password, the values for each are stored in the USER_ID and USER_PASSWORD fields, respectively. A UEM Server will re-read this file as it prepares an event handler process for execution, in order to obtain any changes to the user ID and/or password values contained in the file.	160
USER_FILE_PLAIN	Name of a text file that contains user ID and/or password information.	162
USER_ID	ID of a user account in whose security context the handler process will be executed.	164
USER_PASSWORD	Password for the user account specified by the USER_ID parameter.	165
USER_SCRIPT	Text file that contains statements that are executed collectively as a script by UEM.	166
USER_SCRIPT_TYPE	Type of script statements contained in the action field of the event handler record.	168
VERSION	Displays version information.	169

Table 4.1 UEMLoad Utility - Configuration Options

## 4.4 ACTION

### Description

The ACTION option specifies the requested database operation. It is a required option. Only one database operation can be specified for each invocation of the UEMLoad utility. The database operations supported by the UEMLoad utility are listed below, in Section [Values](#).

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	<code>-action {add   update   delete   list   export}</code>			✓	✓	
Definition File Keyword	n/a					

### Values

There are no values for this option. Instead, the desired action is specified by the parameter:

- **add**  
Writes one or more new event definition and/or event handler records to the appropriate database.
- **update**  
Changes one or more existing event definition and/or event handler records.
- **delete**  
Removes the specified event definition and/or event handler records from the appropriate database.
- **list**  
Displays the complete contents of the specified event definition and/or event handler records.
- **export**  
Dumps the contents of the specified event definition and/or event handler records to a text file that can be used as input to a subsequent run of the UEMLoad utility.

## add

If **add** is specified, any optional event definition or event handler parameters not specified will be provided with default values as specified in the UEM Server configuration. If an attempt is made to add a record that already exists, the operation will fail.

## update

If **update** is specified, only those parameters being changed are required. Values for other parameters are obtained from the existing event definition or event handler record. If an attempt is made to update a record that does not exist, the operation will fail.

## list

If **list** is specified, the only other recognized configuration options are:

- [EVENT\\_ID](#)
- [ASSIGNED\\_COMPONENT\\_NAME](#)
- [HANDLER\\_ID](#)

## export

If **export** is specified, the only other recognized configuration options are:

- [DEFINITION\\_FILE](#)
- [EVENT\\_ID](#)
- [ASSIGNED\\_COMPONENT\\_NAME](#)
- [HANDLER\\_ID](#)



### Stoneman's Tip

To list or export all event definition and event handler records, no configuration options are required; simply specify the *list* or *export* action, respectively.

To export all records to a text file, simply specify a definition load file using the [DEFINITION\\_FILE](#) option.

In both cases, if no event ID, assigned component name, or handler ID is specified, all records in the event definition and event handler databases will be retrieved.

## 4.5 ACTIVE\_DATE\_TIME

### Description

The ACTIVE\_DATE\_TIME option specifies the date and time when UEM Server will begin checking for the occurrence of the system event represented by the event definition.

If ACTIVE\_DATE\_TIME option is not used, default values are set by the UEM Server. When an event definition is added, the date portion of the date and time value is set to the current date. If no time value is provided, the current time is used.

When an event definition is updated and a value for either the date or the time is omitted, the existing value is preserved.

ACTIVE\_DATE\_TIME is ignored for all other requests.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-act_date_time yyyy.mm.dd, hh:mm			✓	✓	
Definition File Keyword	act_date_time yyyy.mm.dd, hh:mm			✓	✓	

### Values

*yyyy.mm.dd, hh:mm* is the date and time when the UEM Server will begun checking.

Either the date or the time can be omitted.

- If the date is omitted, the comma separator must be provided to serve as a placeholder (for example: *,hh:mm*).
- If the time is omitted, the comma is not required.

Note: The maximum valid value for date and time is **2038.01.16, 23:59**.

## 4.6 ASSIGNED\_COMPONENT\_NAME

### Description

---

The `ASSIGNED_COMPONENT_OPTION` specifies the name of an event-driven UEM Server component that is responsible for monitoring the event represented by the event definition record.

To ensure proper assignment, this name should match the name of an installed UEM Server component.

#### Windows

A component's name matches the name of its component definition file. A component definition file is a text file that contains parameters used by the Universal Broker when it starts a component.

Component definition files are installed in the `%ALLUSERSPROFILE%\Application Data\Universal\comp` directory, where `%ALLUSERSPROFILE%` is an environment variable that expands to the All Users folder; typically `C:\Documents and Settings\All Users`.

A list of installed components can be viewed from the Universal Configuration Manager by selecting the Component Definitions page of the Universal Broker configuration control. (For additional information on the Universal Configuration Manager, see the section on Stored Configuration Options in the Universal Event Monitor 3.2.0 User Guide.)

#### UNIX

A component's name matches the name of its component definition file. A component definition file is a text file that contains parameters used by the Universal Broker when it starts a component.

Component definition files are provided by the Universal Products installation, and are installed in the `etc/universal/comp` directory.

When an event-driven UEM Server starts, it obtains all event definition records that have been assigned to it, based on the value specified by this option. An event-driven UEM Server is any UEM Server component whose component definition has the value `uems` specified for its `component_type` parameter.

If a demand-driven UEM Server is asked to monitor an event using a stored event definition record (that is, a UEM Manager was started using the `EVENT_ID` option), the value stored in this field is ignored.

By default, new event definition records are assigned to the UEM Server component defined by the `uems` component definition.





### Stoneman's Tip

The assignment of an event definition to a component is provided primarily for future support of concurrent event-driven UEM Servers. As of the current release, support for only a single instance of an event-driven Server is provided. Therefore, for any event definitions that you want to be monitored by an event-driven Server, it is strongly recommended that you use the default assignment of **uems**.

To store event definitions that do not need to be monitored by an event-driven Server, you can assign them to any component that does not match **uems**. For example, to create an event definition that is intended to only be referenced by a UEM Manager, use a value of **uemd** for the component name parameter.

## Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-comp_name <i>compname</i>			✓	✓	
Definition File Keyword	comp_name <i>compname</i>			✓	✓	

## Value

*compname* is the name of the event-driven UEM Server.

**[Default is *uems*.]**

## 4.7 BROKER\_PORT

### Description

The BROKER\_PORT option specifies the TCP port upon which a local Universal Broker is accepting incoming connections.

The UEMLoad utility establishes a connection to the Broker over this port to request the startup of a UEM Server component that will handle processing of the UEMLoad request.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-p <i>port</i>			✓	✓	
Command Line, Long Form	-port <i>port</i>			✓	✓	
Definition File Keyword	n/a					

### Values

*port* is the TCP port upon which the local Universal Broker is accepting connections.

The format of *port* is either:

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

**[Default is 7887.]**

## 4.8 COMMAND\_ID

### Description

The COMMAND\_ID option specifies an identifier that is saved by the Universal Broker and which is used to identify the unit of work represented by an instance of the UEMLoad utility and its associated UEM Server component.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-C <i>id</i>			✓	✓	
Command Line, Long Form	-cmdid <i>id</i>			✓	✓	
Definition File Keyword	n/a					

### Value

*id* is any value.

If *id* contains spaces, it must be enclosed in double ( " ) quotation marks.

#### UNIX

If *id* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

### Defaults

If the **ACTION** option is set to *add*, *update*, or *delete*, the default *id* is *UEMLoad – Database Update*.

If the **ACTION** option is set to *list*, the default *id* is *UEMLoad – Database List*.

For database *exports*, the default *id* is *UEMLoad – Database Export*.

## 4.9 DEFINITION\_FILE

### Description

The DEFINITION\_FILE option specifies a file that is either read from or written to, depending on the value of the ACTION option:

- If ACTION specifies *-add*, *-update*, or *-delete*, it specifies the path to a text file that contains the parameters for one or more event definition and/or event handler records.
- If ACTION specifies *-export*, it specifies the path to a text file to which the contents of the event definition and event handler records (requested using the EVENT\_ID and HANDLER\_ID options, respectively) are written.

Note: A database export is the only situation in which DEFINITION\_FILE and the EVENT\_ID and/or HANDLER\_ID options can be specified together. DEFINITION\_FILE is ignored if ACTION specifies *-list*.

When parameters for more than one event definition or event handler record are specified in a definition file, certain keywords are required to mark the beginning and ending of each record. These keywords, along with other rules specific to definition files, are described in the Definition File Format sections ( [Chapter 11 UEMLoad Utility for Windows](#) and [Chapter 12 UEMLoad Utility for UNIX](#)) of the UEM 3.2.0 User Guide.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-deffile <i>filename</i>			✓	✓	
Definition File Keyword	n/a					

### Value

*filename* is the file to be written to or read from.

Valid values contain either:

- Complete path
- Path that is relative to the directory from which the UEMLoad utility is invoked

# 4.10 ENCRYPTION\_KEY

## Description

The ENCRYPTION\_KEY option specifies the key that is used to decrypt the file specified by the [USER\\_FILE\\_ENCRYPTED](#) option.

This key is required only if a key was used to encrypt the user file with Universal Encrypt. (For complete information on the Universal Encrypt utility, see the Universal Products Utilities 3.2.0 User Guide.)

If this option is not used, a default key established by the UEMLoad utility is used.

## Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-K <i>key</i>			✓	✓	
Command Line, Long Form	-key <i>key</i>			✓	✓	
Definition File Keyword	key <i>key</i>			✓	✓	

## Value

*key* is the key used to decrypt the data set / file.

## 4.11 EVENT\_ID

### Description

The EVENT\_ID option specifies a value that uniquely identifies an event definition record.

If an attempt is made to *add* an event definition record with an event ID that matches this value, the request will fail. All other operations will fail if the value does NOT match the ID of an existing event definition record.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-event_id <i>id</i>			✓	✓	
Definition File Keyword	event_id <i>id</i>			✓	✓	

### Value

*id* is the ID of an event definition record.

The length of *id* must not exceed 32 characters.

*id* is case-insensitive. When an event definition is added, *id* is stored within the record exactly as specified. However, when that record later is referenced by event ID, case is ignored. For example, if a record is added with an *id* of Event001, the mixed case of the ID is preserved within the record. To access the record after it's been added, an event ID of any case (for example: event001, eVENT001, or EvenT001) can be specified.

### Wildcards

If a *-list*, *-export*, or *-delete* operation is being used for UEMLoad, wildcards can be used in *id* to select multiple event definition records:

- Asterisk ( \* ) will match 0 or more characters.
- Question mark ( ? ) can be used to match any single character.

## 4.12 EVENT\_STATE

### Description

The EVENT\_STATE option specifies whether or not an event is processed by the event-driven UEM Server to which it is assigned (via the event definition record's [ASSIGNED\\_COMPONENT\\_NAME](#) field).

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-state <i>eventstate</i>			✓	✓	
Definition File Keyword	state <i>eventstate</i>			✓	✓	

### Value

*eventstate* is the specification for whether or not the event is processed.

Valid values for *eventstate* are:

- **enable**  
Event definition is processed by UEM.
- **disable**  
Event definition is ignored by UEM.

An event definition whose *eventstate* is set to **disable** still is added to the list of assigned event definitions obtained by an event-driven UEM Server during startup processing. However, the UEM Server will not process that event until:

1. *eventstate* is set to **enable**.
2. UEM Server refreshes its list of assigned event definitions.

**[Default is enable.]**

## 4.13 EVENT\_TYPE

### Description

The EVENT\_TYPE option specifies the type of system event represented by the event definition record.

EVENT\_TYPE is required for new event definition records.

For updates to existing event definition records, EVENT\_TYPE is required only if a change is being made to one of the event definition options that are specific to event types (see [Table 4.2](#), below).

EVENT\_TYPE is ignored for all other requests.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-event_type <i>type</i>			✓	✓	
Definition File Keyword	event_type <i>type</i>			✓	✓	

### Value

*type* is the event type represented by the event definition record.

It must match one of the known event types supported by UEM ([Table 4.2](#)).

Note: Currently, there is only one valid UEM event type: FILE.

Event Type	Description	Associated Options
<i>FILE</i>	Detects the creation of a file and monitors it, testing it for completeness.	<ul style="list-style-type: none"> <li>• <a href="#">FILE_SPECIFICATION</a> *</li> <li>• <a href="#">MINIMUM_FILE_SIZE</a></li> <li>• <a href="#">RENAME_FILE</a></li> <li>• <a href="#">RENAME_FILE_SPECIFICATION</a></li> </ul>

Table 4.2 UEM Event Types



## 4.14 EXPIRED\_HANDLER\_ID

### Description

The EXPIRED\_HANDLER\_ID option specifies the event handler that UEM Server will execute for the event whenever it is set to an `expired` state.

To remove the assignment of an event handler for a particular event definition:

1. Use the `-update` operation (see [ACTION](#)).
2. Specify a value of `none` for this option.

If EXPIRED\_HANDLER\_ID is not used when an event definition record is added, the following default value is set by the UEM Server:

- Windows    Empty string
- UNIX        `NONE`

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	<code>-expired_id handlerid</code>			✓	✓	
Definition File Keyword	<code>expired_id handlerid</code>			✓	✓	

### Value

*handlerid* is the ID of the event handler that UEM will execute.

It must match the ID of an existing record in the event handler database. Execution of the event handler will fail if the UEM Server cannot find the specified handler record.

If *handlerid* contains spaces, it must be enclosed in double ( " ) quotation marks.

#### UNIX

If *handlerid* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

## 4.15 FILE\_SPECIFICATION

### Description

The FILE\_SPECIFICATION option specifies the name of a file whose presence should be detected and monitored by UEM Server.

An absolute path or a path relative to the UEM Server working directory, as defined in the component definition, also can be specified by this option.

Note: This option is valid only for events with an [EVENT\\_TYPE](#) of FILE.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-filespec <i>filename</i>			✓	✓	
Definition File Keyword	filespec <i>filename</i>			✓	✓	

### Values

*filename* is the name of the file to be monitored by UEM.

If *filename* contains spaces, it must be enclosed in double ( " ) quotation marks.

#### UNIX

If *filename* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

### Wildcards

The following wildcards can be specified as part of *filename*.

- \* Match zero, one or more characters.
- ? Match zero or one character.

## 4.16 HANDLER\_ID

### Description

The HANDLER\_ID option specifies a value that uniquely identifies an event handler record.

If an attempt is made to *add* an event handler record with a HANDLER\_ID value that matches the ID of an existing event handler record, the request will fail. All other operations will fail if the HANDLER\_ID value does NOT match the ID of an existing event handler record.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-handler_id <i>id</i>			✓	✓	
Definition File Keyword	handler_id <i>id</i>			✓	✓	

### Value

*id* is the ID of an event handler record.

The length of *id* must not exceed 32 characters.

*id* is case-insensitive. When an event handler is added, *id* is stored within the record exactly as specified. However, when that record later is referenced by handler ID, case is ignored. For example, if a record is added with a handler ID of Handler001, the mixed case of the ID is preserved within the record. To access the record after it's been added, a handler ID of any case (for example: handler001, hANDLEr001, or HANDLer001) can be specified.

If *id* contains spaces, it must be enclosed in double ( " ) quotation marks.

#### UNIX

If *id* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

## Wildcards

If a *-list*, *-export*, or *-delete* operation is being used for UEMLoad, wildcards can be used in *id* to select multiple event handler records:

- Asterisk ( \* ) will match 0 or more characters.
- Question mark ( ? ) can be used to match any single character.

## 4.17 HANDLER\_OPTIONS

### Description

The HANDLER\_OPTIONS option specifies a value that is passed as a command line argument to the process that is executed on behalf of an event handler.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-handler_opts <i>string</i>			✓	✓	
Definition File Keyword	handler_opts <i>string</i>			✓	✓	

### Value

*string* is a character string that is appended to the command line constructed by UEM in order to execute the event handler process.

A value of *none* can be used to remove the handler options string from an existing event definition record.

If *string* contains spaces, it must be enclosed in double ( " ) quotation marks.

#### UNIX

If *string* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

### Windows

If quotes are to be passed as part of the parameter's value (for example, an argument that contains spaces is to be passed to the handler process and treated as a single argument), the required syntax depends on whether the parameter is specified from the command line or from a definition file.

From the command line, use a backslash ( \ ) to escape the quotes that need to be preserved (for example: `-handler_options "parm1 \"parm2a parm2b\" parm3"`). From a definition file, use an extra double quote to escape the quotes (for example, `handler_opts "parm1 ""parm2a parm2b"" parm3"`).

In both cases, three arguments will be passed to the event handler process. The portion of the string enclosed in double quotes (that is, `"parm2a parm2b"`) is treated as a single argument.

### UNIX

If quotes are to be passed as part of the parameter's value (for example, an argument that contains spaces is to be passed to the handler process and treated as a single argument), enclose *options* in single quotes. Use a set of double quotes to enclose the quoted value.

For example, specifying `-handler_options 'parm1 "parm2a parm2b" parm3'` will cause three arguments to be passed to the process executed on behalf of the event handler. The portion of the string enclosed in double quotes (that is, `"parm2a parm2b"`) is treated as a single value.

The HANDLER\_OPTIONS value is very similar to the [OPTIONS](#) value stored in an event handler record. UEM adds both as command line parameters to the command string that it builds to execute an event handler process. The difference is that the value stored in HANDLER\_OPTIONS is used by every event handler referenced by an event definition.

For example, if an event definition contains a value for [TRIGGERED\\_HANDLER\\_ID](#) and [EXPIRED\\_HANDLER\\_ID](#), the command line arguments stored in HANDLER\_OPTIONS are passed to both handler processes.

On the other hand, the value stored in [OPTIONS](#) is used every time the event handler's process is executed. Keep in mind that a single event handler record can be referenced by many event definition records. Because of this one-to-many relationship, UEM adds the value stored in HANDLER\_OPTIONS to the command string after the value stored in the event handler's [OPTIONS](#). This effectively allows an event definition to change the behavior of an event handler by overriding command line parameters (assuming that the last duplicate parameter specified is the one used by the process).

## Default

There is no default value for *options* for new event definition records.

## 4.18 HANDLER\_TYPE

### Description

---

The HANDLER\_TYPE option provides UEM with information that it needs when building a command string in order to execute a process for the event handler.

### Usage

---

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-handler_type <i>type</i>			√	√	
Definition File Keyword	handler_type <i>type</i>			√	√	

---

## Values

---

*type* is the type of information provided to UEM.

Valid values for *type* are:

- **cmd**  
Instructs UEM to execute the command stored in the [USER\\_COMMAND](#) field of the event handler record.
- **script**  
Instructs UEM to write a set of stored script statements to a temporary file, and then execute that file as a script. These script statements can be provided instream from a definition load file using a **begin\_/end\_script** block. The statements also can be imported from a local file, using the [USER\\_SCRIPT](#) option.

## Default

If a *type* is not specified, the default is:

- **cmd**, if a [USER\\_COMMAND](#) is specified.
- **script**, if a [USER\\_SCRIPT](#) is specified.



## 4.19 HELP

### Description

---

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

### Usage

---

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-h -?			✓	✓	
Command Line, Long Form	-help			✓	✓	
Definition File Keyword	n/a					

### Value

---

(There are no values for the HELP option.)

## 4.20 INACTIVE\_DATE\_TIME

### Description

The INACTIVE\_DATE\_TIME option specifies the date and time at which UEM Server will stop checking for the occurrence of the system event represented by the event definition.

If this specified date and time elapses, and no occurrence of the event has been detected, the event is set to an **expired** state.

This option also specifies the date and time by which a tracked event occurrence must complete. Any event occurrences being tracked when this date and time elapse are set to a **rejected** state.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-inact_date_time <i>yyyy.mm.dd,hh:mm</i>			✓	✓	
Definition File Keyword	inact_date_time <i>yyyy.mm.dd,hh:mm</i>			✓	✓	

---

## Values

---

*yyyy.mm.dd,hh:mm* is the date and time - separated by a comma ( , ) - when the UEM Server will stop checking for the occurrence of the system event.

Either the date or time can be omitted:

- If the date is omitted, the comma must be specified to serve as a placeholder (that is: *,hh:mm*).
- If the time is omitted, a comma is not required.

### Default

If INACTIVE\_DATE\_TIME is not used, default values are set by the UEM Server.

- When an event definition is added, the date defaults to *2038.01.16*.
- If time only is omitted, a default of *23:59* is used.

When an event definition is updated, and a value for date and/or time is omitted, the existing value is preserved.

INACTIVE\_DATE\_TIME is ignored for all other requests.

## 4.21 MAXIMUM\_RETURN\_CODE

### Description

The MAXIMUM\_RETURN\_CODE option specifies the highest value that a process executed on behalf of an event handler can return to be considered as having executed successfully.

UEM Server reports an execution error if the value returned by the handler process is greater than the value specified by MAXIMUM\_RETURN\_CODE.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-maxrc <i>returncode</i>			✓	✓	
Definition File Keyword	maxrc <i>returncode</i>			✓	✓	

### Value

*returncode* is the highest value that a process can return to be considered as having been successful.

If MAXIMUM\_RETURN\_CODE is not used, a default value of 0 (zero) is set by the UEM Server.

## 4.22 MESSAGE\_LEVEL

### Description

The MESSAGE\_LEVEL option specifies the level of messages that will be issued.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-l <i>level</i>			✓	✓	
Command Line, Long Form	-level <i>level</i>			✓	✓	
Definition File Keyword	n/a					

### Value

*level* is the level of messages to be issued.

Valid values for *level* are:

- **trace**  
Generates a text file that contains detailed program execution information.  
Note: **trace** is used for debugging purposes only. It should be used only when requested by Stonebranch Inc. [Customer Support](#).
- **audit**  
Issues audit, informational, warning and error messages.
- **info**  
Issues informational, warning and error messages.
- **warn**  
Issues warning and error messages.
- **error**  
Issues error messages only.

**[Default is info.]**

## 4.23 MINIMUM\_FILE\_SIZE

### Description

The MINIMUM\_FILE\_SIZE option specifies the smallest allowable size for a file in order for it to be considered complete by UEM Server.

Note: This option is valid only for files being monitored via [EVENT\\_TYPE](#) (event type = **FILE**).

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-min_file_size <i>size</i> [ <i>units</i> ]			✓	✓	
Definition File Keyword	min_file_size <i>size</i> [ <i>units</i> ]			✓	✓	

### Value

*size* is the file size. *units* is the unit of storage for that *size*.

The valid values for *units*, and the maximum value that can be specified for *size* for that unit of storage, is:

- **b** bytes (4,294,967,295)
- **k** kilobytes (4,194,304)
- **m** megabytes (4,096)
- **g** gigabytes (4)

If a value for *units* is not specified, the file size is assumed to be in bytes.

If this option is not used, a default value of 0 (zero), as set by the UEM Server configuration, is used.

A value of 0 (zero) also can be used to indicate that the file can be any size.

# 4.24 OPTIONS

## Description

The OPTIONS option specifies a value that is passed as a command line argument to the process executed for an event handler.

UEM adds this value to the command string that it builds as it prepares the event handler process for execution.

OPTIONS is very similar to the [HANDLER\\_OPTIONS](#) option stored in the event definition. UEM adds the parameters for both as command line parameters to the command string built in order to execute an event handler process. The difference is that the value stored in [HANDLER\\_OPTIONS](#) is used for every event handler referenced by an event definition. For example, if an event definition contains a value for the [TRIGGERED\\_HANDLER\\_ID](#) and [EXPIRED\\_HANDLER\\_ID](#), then the command line arguments stored in [HANDLER\\_OPTIONS](#) are passed to both handler processes.

On the other hand, the value stored in OPTIONS is used every time that the event handler's process is executed. Keep in mind that a single event handler record can be referenced by many event definition records. Because of this one-to-many relationship, UEM adds the value stored [HANDLER\\_OPTIONS](#) to the command string after the value stored in the event handler's OPTIONS. This effectively allows an event definition to change the behavior of an event handler by overriding command line parameters (assuming that the last duplicate parameter specified is the one used by the process).

## Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-options <i>string</i>			✓	✓	
Definition File Keyword	options <i>string</i>			✓	✓	

---

## Value

---

*string* is the value passed as a command line argument.

A value of *none* can be used to remove the options string from an existing event handler record.

If *string* contains spaces, it must be enclosed in double ( " ) quotation marks.

### UNIX

If *string* contains spaces, it must be enclosed either in single ( ' ) or double ( " ) quotation marks.

### Windows

If quotes are to be passed as part of the parameter's value (for example, an argument that contains spaces is to be passed to the handler process and treated as a single argument), the required syntax depends on whether the parameter is specified from the command line or from a definition file.

From the command line, use a backslash ( \ ) to escape the quotes that need to be preserved (for example, **-options "parm1 \"parm2a parm2b\" parm3"**).

From a definition file, use an extra double quote to escape the quotes (for example, **options "parm1 \"\"parm2a parm2b\"\" parm3"**).

In both cases, three arguments will be passed to the event handler process. The portion of the string enclosed in double quotes (that is, **"parm2a parm2b"**) is treated as a single argument.

### UNIX

If quotes are to be passed as part of the parameter's value (for example, an argument that contains spaces is to be passed to the handler process and treated as a single argument), enclose *string* in single quotes, and use a set of double quotes to enclose the quoted value.

For example, specifying **-options 'parm1 "parm2a parm2b" parm3'** will cause three arguments to be passed to the event handler process. The portion of the string enclosed in double quotes (that is, **"parm2a parm2b"**) is treated as a single value.

## Default

There is no default value for this option for new event handler records.



## 4.25 REJECTED\_HANDLER

### Description

The REJECTED\_HANDLER option specifies the event handler that UEM Server will execute for a tracked event occurrence that it is set to a **rejected** state.

The specified event handler must match the ID of an existing record in the event handler database. Execution of the event handler will fail if the UEM Server cannot find the specified handler record.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-rejected_id <i>handlerid</i>			✓	✓	
Definition File Keyword	rejected_id <i>handlerid</i>			✓	✓	

### Value

*handlerid* is the event handler to be executed.

If *handlerid* contains spaces, it must be enclosed in double ( " ) quotation marks.

#### UNIX

If *handlerid* contains spaces, it must be enclosed either in single ( ' ) or double ( " ) quotation marks.

To remove the assignment of an event handler for a particular event definition, use the *-update* command operation (see [ACTION](#)) and specify a value of *none* for *handlerid*.

### Default

If REJECTED\_HANDLER is not used when an event definition record is added, the following default values are used:

- Windows empty string
- UNIX *none*

## 4.26 RENAME\_FILE

### Description

The RENAME\_FILE option specifies whether or not a file that is being monitored - as specified via an [EVENT\\_TYPE](#) of `FILE` - should be renamed by UEM when the event occurrence is set to a `triggered` state.

Renaming a file ensures that UEM will not treat a file that it just finished tracking as a new event occurrence.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-rename_file <i>option</i>			✓	✓	
Definition File Keyword	rename_file <i>option</i>			✓	✓	

### Value

*option* is the specification for whether or not a file should be renamed.

Valid values for *option* are:

- **yes** File is renamed according to the [RENAME\\_FILE\\_SPECIFICATION](#) option.
- **no** File is not renamed.

If *option* is set to **no**, in order to prevent multiple detection of the file by the UEM Server, rename the file in a script invoked by an event handler when the occurrence is triggered.

If RENAME\_FILE is not used, a default value of **yes**, as set by the UEM Server configuration, is used.

# 4.27 RENAME\_FILE\_SPECIFICATION

## Description

The RENAME\_FILE\_SPECIFICATION option specifies the file name that the UEM Server should use when both:

- An event occurrence for an event with an **EVENT\_TYPE** of **FILE** is set to a **triggered** state.
- **RENAME\_FILE** option is set to **yes**.

## Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-rename_filespec <i>renamefile</i>			✓	✓	
Definition File Keyword	rename_filespec <i>renamefile</i>			✓	✓	

## Value

*renamefile* is the file name to be used by the UEM Server.

Valid values for *renamefile* can include an absolute path or a path that is relative to the location of the file monitored by UEM. If no path information is provided, the monitored file simply is renamed. The renamed file will reside in the same location as the original file.

If *renamefile* contains spaces, it must be enclosed in double ( " ) quotation marks.

### UNIX

If *renamefile* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

## Variables

Variables that UEM will substitute with actual runtime values can be included as part of *renamefile*. These variables, and the values that UEM will substitute for them, are:

- **\$(compname)**  
Component name
- **\$(compid)**  
Component ID
- **\$(date)**  
Current date, in the format *YYYYMMDD*
- **\$(time)**  
Current time, in the format *HHMMSS*
- **\$(origname)**  
Original base file name, minus its last extension
- **\$(origext)**  
Original file extension
- **\$(seqnum)**  
Sequence number that is initialized to 0 when the demand-driven UEM Server is started, and is then incremented by 1 for each file renamed.

If this option is not used, the following default value, set by the UEM Server configuration, is used:

**\$(compname) . \$(compid) . \$(date) . \$(seqnum)**

## 4.28 TRACKING\_INTERVAL

### Description

The TRACKING\_INTERVAL option specifies the frequency with which the UEM Server will test for the completion of a tracked event occurrence.

The test for completeness depends upon the type of event that the tracked occurrence represents. For example, an occurrence tracked for an event with an [EVENT\\_TYPE](#) of **FILE** is considered complete when two consecutive checks on a file's size return the same value.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-tracking_int <i>seconds</i>			✓	✓	
Definition File Keyword	tracking_int <i>seconds</i>			✓	✓	

### Value

*seconds* is the frequency (in seconds) with which the UEM Server tests for completion of a tracked occurrence.

If TRACKING\_INTERVAL is not used, a default value of 10 seconds, set by the UEM Server configuration, is used.

## 4.29 TRIGGERED\_HANDLER\_ID

### Description

The TRIGGERED\_HANDLER\_ID option specifies the event handler that the UEM Server will execute for a tracked event occurrence that is set to a **triggered** state.

This event handler must match the ID of an existing record in the event handler database. Execution of the event handler will fail if the UEM Server cannot find the specified handler record.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-triggered_id <i>handlerid</i>			✓	✓	
Definition File Keyword	triggered_id <i>handlerid</i>			✓	✓	

### Value

*handlerid* is the event handler to be executed.

If *handlerid* contains spaces, it must be enclosed in double ( " ) quotation marks.

#### UNIX

If *handlerid* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

To remove the assignment of an event handler for a particular event definition, use the *-update* command operation (see [ACTION](#)) and specify a value of *NONE* for *handlerid*.

### Default

If TRIGGERED\_HANDLER\_ID is not used when an event definition record is added, the following default values are set by the UEM Server:

- Windows empty string
- UNIX *NONE*

## 4.30 USER\_COMMAND

### Description

The USER\_COMMAND option specifies a system command that is stored in an event handler record with a handler type of **CMD**. It identifies the handler process, which is executed by UEM Server on behalf of the stored event handler.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-cmd <i>command</i>			✓	✓	
Definition File Keyword	cmd <i>command</i>			✓	✓	

### Value

*command* is the system command stored in an event handler record. If *command* contains spaces, it must be enclosed in double ( " ) quotation marks.

#### Windows

If quotes are to be saved as part of *command*, the required syntax depends on whether the parameter is specified from the command line or from a definition file.

From the command line, use a backslash ( \ ) to escape the quotes that need to be preserved (for example, **-cmd "c:\someapp.exe \"quoted string\"**).

From a definition file, use an extra double quote to escape the quotes (for example, **cmd "c:\someapp.exe \"quoted string\""**).

In both cases, the parameter **"quoted string"** will be passed to **c:\someapp.exe** and treated as a single command line argument.

#### UNIX

If *command* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

If quotes are to be saved as part of *command*:

- Enclose *command* in single quotes.
- Use a set of double quotes to enclose the quoted value.

For example, specifying **-cmd '/usr/someapp "quoted string"'** will save everything between the single quotes, including the spaces and double quotes, in the event handler record.

## 4.31 USER\_FILE\_ENCRYPTED

### Description

---

The `USER_FILE_ENCRYPTED` option specifies a file that contains the parameters and encrypted values for the `USER_ID` and/or `USER_PASSWORD` options.

Storing these parameters and values in a file can be used in situations where it is not desirable to specify them explicitly on the command line. These parameters must be specified in their respective command line formats.

Any file specified by this `USER_FILE_ENCRYPTED` option must be encrypted with the Universal Encrypt utility. (For more information on Universal Encrypt, see the Universal Products Utilities 3.2.0 User Guide.)

**Note:** This file should not be specified additionally with the `USER_FILE_PLAIN` option. If both `USER_FILE_ENCRYPTED` and `USER_FILE_PLAIN` specify this file, the file specified by `USER_FILE_PLAIN` will be used.

It is strongly recommended that any user file (including those that are encrypted) which contains sensitive data should be further protected from unauthorized access using file level security.

UEM Server uses this information to execute an event handler process in the security context of the specified user account. An event-driven UEM Server requires this information to be stored in an event handler record if the `USER_SECURITY` option is enabled in the UEM Server configuration. This is the only means by which an event-driven UEM Server can access this information. Because an event-driven UEM Server is started without user interaction, it cannot prompt for user account information if it is required but not provided.

To simplify database administration, the file specified in `USER_FILE_ENCRYPTED` is stored in the event handler record. If this file contains a user ID and password, the UEMLoad utility saves those values in the event handler's `USER_ID` and `USER_PASSWORD` fields, respectively.

However, UEM does not rely strictly on the values stored in those fields. UEM re-reads the encrypted file stored in `USER_FILE_ENCRYPTED`, just before executing the event handler process, in order to obtain any changes made to the account information stored in the file. This makes it possible to update account information for several event handler processes without having to change the event handler records themselves.



## Usage

---

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-x <i>filename</i>					
Command Line, Long Form	-encryptedfile <i>filename</i>			√	√	
Definition File Keyword	encryptedfile <i>filename</i>			√	√	

## Value

---

*filename* is file containing the encrypted values.

*filename* can contain either:

- Complete path
- Path that is relative to the directory from which the UEMLoad utility is invoked

If *filename* contains spaces, it must be enclosed in double ( " ) quotation marks.

### UNIX

If *filename* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

## 4.32 USER\_FILE\_PLAIN

### Description

The USER\_FILE\_PLAIN option specifies a plain text file that contains the parameters and their values for the [USER\\_ID](#) and/or [USER\\_PASSWORD](#) options.

Storing these parameters and values in a file can be used in situations where it is not desirable to specify them explicitly on the command line. These parameters must be specified in their respective command line formats.

Note: This file should not be specified additionally with the [USER\\_FILE\\_ENCRYPTED](#) option. If both [USER\\_FILE\\_ENCRYPTED](#) and USER\_FILE\_PLAIN specify this file, the file specified by USER\_FILE\_PLAIN will be used.

It is strongly recommended that any user file which contains sensitive data should be further protected from unauthorized access using file level security.

UEM Server uses this information to execute an event handler process in the security context of the specified user account. An event-driven UEM Server requires this information to be stored in an event handler record if the [USER\\_SECURITY](#) option is enabled in the UEM Server configuration. This is the only means by which an event-driven UEM Server can access this information. Because an event-driven UEM Server is started without user interaction, it cannot prompt for user account information if it is required but not provided.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-f <i>filename</i>			✓	✓	
Command Line, Long Form	-file <i>filename</i>			✓	✓	
Definition File Keyword	file <i>filename</i>			✓	✓	

---

## Value

---

*filename* is file containing the values.

*filename* can contain either:

- Complete path
- Path that is relative to the directory from which the UEMLoad utility is invoked

If *filename* contains spaces, it must be enclosed in double ( " ) quotation marks.

### UNIX

If *filename* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

## 4.33 USER\_ID

### Description

The `USER_ID` option specifies the ID of a user account directly from the command line or definition file.

This is an alternative to specifying the ID from a file.

UEM Server uses this information to execute an event handler process in the security context of the specified user account. An event-driven UEM Server requires this information to be stored in an event handler record if the `USER_SECURITY` option is enabled in the UEM Server configuration. This is the only means by which an event-driven UEM Server can access this information. Because an event-driven UEM Server is started without user interaction, it cannot prompt for user account information if it is required but not provided.

Note: If UEMLoad specifies an encrypted file (via `USER_FILE_ENCRYPTED`), it extracts the `-userid` value from that file and stores it in the `USER_ID` field. However, UEMLoad also saves a reference to the encrypted file itself.

When UEM Server executes a process on behalf of the event handler, it re-extracts the `-userid` option from the file. This means that if the contents of the encrypted file change after the UEMLoad, the `USER_ID` value reported in a list or export may not match the actual `USER_ID` that UEM Server uses when it executes a secured process on behalf of that event handler.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	<code>-u <i>userid</i></code>			✓	✓	
Command Line, Long Form	<code>-userid <i>userid</i></code>			✓	✓	
Definition File Keyword	<code>userid <i>userid</i></code>			✓	✓	

### Value

*userid* is the ID of a user account.

Valid values for *userid* are valid user accounts that are known to the target system.

## 4.34 USER\_PASSWORD

### Description

The USER\_PASSWORD option specifies the password for the user account identified by the USER\_ID option.

This is an alternative to specifying the password from a plain text or encrypted user file.

UEM Server uses this information to execute an event handler process in the security context of the specified user account. An event-driven UEM Server requires this information be stored in an event handler record if the USER\_SECURITY option is enabled in the UEM Server configuration. This is the only way that an event-driven UEM Server can access this information. Because an event-driven UEM Server is started without user interaction, it cannot prompt for user account information if it is required but not provided.

Note: If UEMLoad specifies an encrypted file (via USER\_FILE\_ENCRYPTED), it extracts the -pwd value from that file and stores it in the USER\_PASSWORD field. However, UEMLoad also saves a reference to the encrypted file itself.

When UEM Server executes a process on behalf of the event handler, it re-extracts the -pwd option from the file. This means that if the contents of the encrypted file change after the UEMLoad, the stored USER\_PASSWORD may not match the actual USER\_PASSWORD that UEM Server uses when it executes a secured process on behalf of that event handler.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-w password			✓	✓	
Command Line, Long Form	-pwd password			✓	✓	
Definition File Keyword	pwd password OR password password			✓	✓	

### Value

*password* is the password for the user account.

## 4.35 USER\_SCRIPT

### Description

The USER\_SCRIPT option specifies a path to a text file that contains one or more script statements that UEM Server will execute on behalf of the event handler.

UEMLoad will store each of these statements in the action field of the event handler record. As UEM Server prepares to execute the handler, it will write each of these statements to a temporary script file and construct a command line to execute that file.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-script <i>filename</i>			✓	✓	
Definition File Keyword	script <i>filename</i>			✓	✓	

---

## Value

---

*filename* is the script file containing the statements to be executed.

*filename* can contain either:

- Complete path
- Path that is relative to the directory from which the UEMLoad utility is invoked

If *filename* contains spaces, it must be enclosed in double ( " ) quotation marks.

### UNIX

If *filename* contains spaces, it must be enclosed in either single ( ' ) or double ( " ) quotation marks.

When specified from the command line, *filename* must be for that of an existing file. If a [DEFINITION\\_FILE](#) option is used to store the event handler record, script statements can be specified in-line.

(For information on formatting the contents of a definition file, see the Definition File Format sections in the Universal Event Monitor 3.2.0 User Guide.)

## 4.36 USER\_SCRIPT\_TYPE

### Description

The USER\_SCRIPT\_TYPE option describes the type of statements contained in the script specified by the [USER\\_SCRIPT](#) option.

### Usage

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	n/a					
Command Line, Long Form	-script_type <i>type</i>			✓	✓	
Definition File Keyword	script_type <i>type</i>			✓	✓	

### Value

*type* is the type of statements in the script specified by [USER\\_SCRIPT](#).

#### Windows

When UEM Server executes an event handler with a [HANDLER\\_TYPE](#) of **script**, it writes the statements contained in the Handler Actions field of the handler record to a temporary file. The USER\_SCRIPT\_TYPE **type** value is used as that file's extension.

On the target system, a file association between *type* and an application responsible for opening files with that extension must be defined. While this association is not required, when the event handler record is added with UEMLoad, it must be defined prior to any attempt by UEM Server to execute the handler. Otherwise, execution of the script will fail.

#### UNIX

While this option can be specified from the command line, its value will be ignored when UEM Server executes the script.

### Default

If this option is not used, a default value of **bat**, set by the UEM Server configuration, is used.



## 4.37 VERSION

### Description

---

The VERSION option instructs the UEMLoad utility to display version and copyright information.

### Usage

---

Specification Method	Parameter / Value	OS/400	NonStop	UNIX	Windows	z/OS
Command Line, Short Form	-v			✓	✓	
Command Line, Long Form	-version			✓	✓	
Definition File Keyword	n/a					

### Value

---

(There are no values for this option.)

---

# Chapter 5

# Universal Event Monitor

# Component Definition Options

---

## 5.1 Overview

This chapter provides detailed information about the options that comprise Universal Event Monitor (UEM) component definitions.

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

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

Section [5.2 Component Definition Options Information](#) provides a guideline for understanding the information presented for each component definition option.

## 5.2 Component Definition Options Information

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

### Description

---

Describes the option and how it is used.

### Usage

---

Provides a table of the following information:

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

#### Method

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

- Component Definition Keyword

#### Syntax

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

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

#### (Operating System)

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

- **O** OS/400
- **T** HP NonStop
- **U** UNIX
- **W** Windows
- **Z** z/OS

---

## Values

---

Identifies all possible values for the specified value type.

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

## 5.3 Component Definition Options

Table 5.1 identifies all Universal Event Monitor component definitions.

Component	Description	Page
AUTOMATICALLY_START	Specification for whether or nor the UEM Server starts automatically when the Universal Broker is started	174
COMPONENT_NAME	Name by which clients know the UEM Server	175
COMPONENT_TYPE	Type of UEM Server (demand-driven or event-driven)	176
CONFIGURATION_FILE *	Full path name of the UEM Server configuration file	177
RUNNING_MAXIMUM	Maximum number of UEM Servers that can run simultaneously	178
START_COMMAND *	Full path name of the UEM Server executable file	179
WORKING_DIRECTORY *	Full path name of the UEM Server working directory	181
* These options are required in all component definitions.		

Table 5.1 Universal Event Monitor - Component Definition Options

## 5.4 AUTOMATICALLY\_START

### Description

The AUTOMATICALLY\_START option indicates whether or not UEM Server is started automatically when the Universal Broker is started.

Note: AUTOMATICALLY\_START is optional in a component definition.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Component Definition Keyword	auto_start <i>option</i>			✓	✓	

### Values

*option* is the specification for how the UEM Server is started.

Valid values for *option* are:

- **yes**  
UEM Server is started automatically when the Universal Broker is started.  
Note: Select **yes** for an event-driven UEM Server. The attempt to start a UEM Server automatically will succeed only if the values for the [COMPONENT\\_TYPE](#) and [START\\_COMMAND](#) component definition options are compatible with an event-driven UEM Server.
- **no**  
UEM Server is not started automatically when the Universal Broker is started.  
Note: Select **no** for a demand-driven UEM Server and for any event-driven UEM Server that should not be started automatically.

**[Default is yes.]**

## 5.5 COMPONENT\_NAME

### Description

---

The COMPONENT\_NAME option specifies the name of the UEM Server.

Component start requests refer to the UEM Server by this name.

Note: COMPONENT\_NAME is optional in a component definition. If it is not specified, the file name is used as the component name.

### Usage

---

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Component Definition Keyword	component_name <i>name</i>			✓	✓	

### Values

---

*name* is the name by which the clients know the UEM Server:

- Name of the demand-driven UEM Server is **uemd**.
- Name of the default event-driven UEM Server is **uems**.

## 5.6 COMPONENT\_TYPE

### Description

The COMPONENT\_TYPE option identifies whether the component definition is for a demand-driven or event-driven UEM Server.

Note: COMPONENT\_TYPE is optional in a component definition. If it is not specified, the component name is used.

### Usage

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

### Values

*type* is the type of component.

- For demand-driven UEM Servers, *type* is **uemd**.
- For event-driven UEM servers, *type* is **uems**.

Note: This value is used for internal validation by the Universal Broker and should not be changed.



## 5.7 CONFIGURATION\_FILE

### Description

The CONFIGURATION\_FILE option specifies the full path name of the UEM Server configuration file.

Note: CONFIGURATION\_FILE is required in a component definition.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Component Definition Keyword	configuration_file <i>filename</i>			✓	✓	

### Values

*filename* is the name of the configuration file.

*filename* can be any valid file name.

#### Defaults

##### UNIX

Installation default is `/etc/universal/uems.conf`.

##### Windows

Installation default is `C:\Documents and Settings\All Users\Universal\conf\uems.conf`.

## 5.8 RUNNING\_MAXIMUM

### Description

---

The RUNNING\_MAXIMUM option specifies the maximum number of UEM Servers that can run simultaneously.

If this maximum number is reached, any command received to start a UEM Server is rejected.

Note: RUNNING\_MAXIMUM is optional in a component definition.

### Usage

---

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Component Definition Keyword	running_max <i>max</i>			✓	✓	

### Values

---

*max* is the maximum number of UEM Servers that can run simultaneously.

**[Default is 100.]**

## 5.9 START\_COMMAND

### Description

The START\_COMMAND option specifies the full path name to the UEM Server program.

START\_COMMAND also specifies whether the UEM Server is to run in demand-driven mode or event-driven mode.

- For a demand-driven UEM Server (that is, a UEM Server whose [COMPONENT\\_TYPE](#) is set to `uemd`), the following parameter and value are required as part of this option: `-demand yes`.
- For an event-driven UEM Server, (that is, a UEM Server whose [COMPONENT\\_TYPE](#) is set to `uems`), the `-demand` parameter must be omitted from this option.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
Component Definition Keyword	<code>start_command name</code> <code>-demand option</code>			✓	✓	

### Values

*name* is the full path name of the UEM Server program.

*option* is the specification for whether the UEM Server is to be run in demand-driven mode or event-driven mode.

## Defaults

### UNIX

- For a demand-driven UEM Server: **INSTALLDIR/universal/uemsv/bin/uemsv -demand yes**, where **INSTALLDIR** is the primary installation directory (for example, **/opt**).
- For an event-driven UEM Server: **INSTALLDIR/universal/uemsv/bin/uemsv**, where **INSTALLDIR** is the primary installation directory (for example, **/opt**).

### Windows

- For the demand-driven UEM Server: **INSTALLDIR\Universal\UEMSrv\bin\uemsv.exe -demand yes**, where **INSTALLDIR** is the primary installation directory (for example, **C:\Program Files**).
- For the event-driven UEM Server: **INSTALLDIR\Universal\UEMSrv\bin\uemsv.exe**, where **INSTALLDIR** is the primary installation directory (for example, **C:\Program Files**).

Note: These values are set during product installation and should not be changed.

## 5.10 WORKING\_DIRECTORY

### Description

The WORKING\_DIRECTORY option specifies the full path name used as the working directory of UEM Server.

Note: WORKING\_DIRECTORY is required in a component definition.

### Usage

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

### Values

*directory* is the full path name of the working directory.

#### UNIX

By default, the value of this option is **INSTALLDIR/uemsrv**, where **INSTALLDIR** is the primary installation directory (for example, **/opt/universal**).

When a demand-driven UEM Server is run with the **USER\_SECURITY** configuration option set to **yes**, the working directory is the home directory of the user account specified by the UEM Manager's command options.

The working directory for all event-driven UEM Servers and demand-driven UEM Servers run when the security configuration option is set to **no** is the one specified by this parameter.

#### Windows

By default, the value of this parameter is **INSTALLDIR\Universal\UEMHome**, where **INSTALLDIR** is the primary installation directory (for example, **C:\Program Files**). If this directory does not exist, it is created during product installation.

When a demand-driven UEM Server is run with the **USER\_SECURITY** option set to **yes**, the actual working directory is a subdirectory of the directory specified by this WORKING\_DIRECTORY option. This subdirectory's name matches the value of the user ID specified by the UEM Manager.

The working directory for all event-driven UEM Servers, and demand-driven UEM Servers run when the **USER\_SECURITY** option is set to **no**, is the one specified by this option.

---

# Chapter 6

# Universal Event Monitor

# UACL Entries

---

## 6.1 Overview

This chapter provides detailed information on the Universal Access Control List (UACL) entries available for use with Universal Event Monitor.

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

Information on how these UACL entries are used is documented in the Universal Event Monitor 3.2.0 User Guide.

Section [6.2 UACL Entries Information](#) provides a guideline for understanding the information presented for each UACL entry.

## 6.2 UACL Entries Information

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

### Description

---

Describes the UACL entry and how it is used.

### Usage

---

Provides a table of the following information:

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

#### Method

Identifies the method used for specifying a UACL entry:

- UACL File Keyword

#### Syntax

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

- Type      Universal Products component to which the rule applies.
- Rule      Client's identity, request to which the entry pertains, and security attributes that the entry enforces.

#### (Operating System)

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

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

---

## Values

---

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

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



## 6.3 UACL Entries List

Table 6.1 identifies all Universal Event Monitor UACL entries.

UACL Entry	Description	Page
ACCESS_ACL	Allows or denies access to a demand-driven Universal Event Monitor Server.	186
DATABASE_MAINTENANCE_ACL	Allows or denies a user account access to the event definition and event handler databases.	188
EVENT_HANDLER_ACL	Allows or denies a user account the authority to execute an event handler process.	189

Table 6.1 Universal Event Monitor UACL Entries

## 6.4 ACCESS\_ACL

### Description

The ACCESS\_ACL option is used to allow or deny access to a demand-driven Universal Event Monitor Server. These rules are applied anytime a request to start a demand-driven Server is made, regardless of which Universal Event Monitor client application (that is, the UEMLoad utility or UEM Manager) issued the request.

Access is allowed or denied based on the following attributes:

- IP address or host name of the system on which the UEM client application is executing (*host*)
- ID of the user account with which a UEM Manager is executing (*remote\_user*)
- ID of a user account, specified via the USER\_ID parameter of the UEM Manager, that is defined to the system on which the UEM Server is executing (*local\_user*)

#### UNIX

An Access ACL entry also uses the value specified by **auth** to determine whether the local user account must be authenticated in order for the UEM Server to continue.

For a detailed discussion on Universal ACLs, see Section [2.7 Universal Access Control List](#) in the Universal Event Monitor 3.2.0 User Guide.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
UACL File Keyword	<code>uem_access host,remote_user,local_user,access, auth</code>			✓	✓	

---

## Value

---

*access* is the specification for whether or not access is permitted.

Valid values for *access* are:

- **allow**  
Access is permitted.
- **deny**  
Access is not permitted. A message is returned to the client application (either the UEMLoad utility or UEM Manager), and the connection between the client and the Server is closed.

### UNIX

*auth* is the specification for whether or not the local user account must be authenticated.

Valid values for *auth* are:

- **auth**  
Local user account must be authenticated. The UEM Manager must provide a proper password for the account.
- **noauth**  
Local user account does not require user authentication. The UEM Manager still must supply a password to satisfy command syntax rules, but it will not be verified; any password value will suffice.

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

## 6.5 DATABASE\_MAINTENANCE\_ACL

### Description

The DATABASE\_MAINTENANCE\_ACL option allows or denies a user account access to the event definition and event handler databases.

Access is granted or denied based on the following attributes:

- ID of the user account executing the Universal Event Monitor Load utility (`local_user`)
- Type of database access requested (`add,update,delete,list`)

For a detailed discussion on Universal ACLs, see Section [2.7 Universal Access Control List](#) in the UEM 3.2.0 User Guide.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
UACL File Keyword	<code>uem_maintenance</code> <code>local_user,add,update,delete,list</code>			✓	✓	

### Value

Valid values for *add*, *update*, *delete*, and *list* are:

- **allow**  
Access is permitted for the specified operation for the user account identified by `local_user`.
- **deny**  
Access is not permitted for the specified operation. If this operation is attempted by the user account identified by `local_user`, a message is returned to the UEMLoad utility and the connection between the client and the UEM Server is closed.

## 6.6 EVENT\_HANDLER\_ACL

### Description

The EVENT\_HANDLER\_ACL option allows or denies a user account the authority to execute an event handler process.

Note: For a detailed discussion on Universal ACLs, see Section [2.7 Universal Access Control List](#) in the UEM 3.2.0 User Guide.

### Usage

Method	Syntax	OS/400	NonStop	UNIX	Windows	z/OS
UACL File Keyword	uem_handler <i>user,access,auth</i>			✓	✓	

### Values

*user* is the name of the user account.

#### Windows

For a demand-driven Server, *user* is the one specified from the Universal Event Monitor Manager's command options. For an event-driven Server, the *user* checked is the one stored in the event handler record.

For either type of Server, *user* is checked only if the [USER\\_SECURITY](#) configuration option is not set to **none**.

*access* specifies the security context of the event handler processes.

Valid values for *access* are:

- **allow**  
Event handler processes can be run in the security context of the user account specified by *user*.
- **deny**  
Event handler processes cannot be run in the security context of the user account specified by *user*.

**UNIX**

An Event Handler ACL entry also uses the value specified by *auth* to determine whether the user account must be authenticated with a password before the handler process can be executed.

Valid values for *auth* are:

- **auth**  
User account must be authenticated.
  - For a demand-driven UEM Server, a proper password must be provided by the UEM Manager for the account.
  - For an event-driven UEM Server, a valid user ID and password must be available in the stored event handler record.

- **noauth**  
The user account does not require user authentication. A password still may be necessary to satisfy command syntax rules, but it will not be verified. Any password value will suffice.

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

---

# Chapter 7

## Additional Information for Universal Event Monitor

---

### 7.1 Overview

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

[Table 7.1](#), below, identifies the type of information provided in this chapter and provides a link to each section.

Information	Description	Page
DD Statements	DD statements used in the Universal Command Manager for z/OS Batch JCL.	<a href="#">192</a>
Character Code Pages	Character code pages for use with Universal Event Monitor.	<a href="#">193</a>
UTT Files	Universal Translate Table (UTT) files are used to translate between Unicode and the local single-byte code page.	<a href="#">195</a>
SSL Cipher Suites	SSL cipher suites for use with Universal Event Monitor.	<a href="#">196</a>
Event Definition Parameters - General	General parameters for event definitions.	<a href="#">197</a>
Event Definition Parameters - Event Type FILE	Parameters that are available in event definitions for an Event Type of FILE.	<a href="#">199</a>
Event Handler Parameters	Parameters for event handlers.	<a href="#">201</a>

Table 7.1 Universal Event Monitor - Additional Information

## 7.2 DD Statements used in JCL

Table 7.2, below, describes the DD statements used in the Universal Event Monitor Manager for z/OS batch JCL.

DD name	DCB Attributes *	Mode	Description
STEPLIB	DSORG=PO, RECFM=U	input	Universal Products load library which contains the program to execute.
UNVCONF	DSORG=PS, RECFM=(F, FB, V, VB)	input	Universal Event Monitor Manager 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.
SYSPRINT	DSORG=PS, RECFM=(F, FB, V, VB)	output	Standard output file for the UEM program. UEM does not write any messages to SYSPRINT.
SYSOUT	DSORG=PS, RECFM=(F, FB, V, VB)	output	Standard error file for the UEM program. UEM writes its messages to SYSOUT.
SYSIN	DSORG=PS, RECFM=(F, FB, V, VB)	input	Standard input file for the UEM program. UEM reads its command options from SYSIN.
* The C runtime library determines the default DCB attributes. Refer to the IBM manual <i>OS/390 C/C++ Programming Guide</i> for details on default DCB attributes for stream I/O			

Table 7.2 Universal Event Monitor Manager for z/OS - Batch JCL DD Statements



## 7.3 Character Code Pages

Table 7.3 identifies the character code pages provided by Stonebranch Inc. for use with Universal Products on each supported operating system.

Code Page	CCSID	z/OS	UNIX	Windows	OS/400		HP NonStop
					HFS	LIB	
IBM037	037	✓			✓	✓	
IBM273	273	✓			✓	✓	
IBM277	277	✓			✓	✓	
IBM278	278	✓			✓	✓	
IBM280	280	✓			✓	✓	
IBM284	284	✓			✓	✓	
IBM500	500	✓			✓	✓	
IBM875	875	✓					
IBM1047							
IBM1140	1140	✓			✓	✓	
IBM1141	1141	✓			✓	✓	
IBM1142	1142	✓			✓	✓	
IBM1143	1143	✓			✓	✓	
IBM1144	1144	✓			✓	✓	
IBM1145	1145	✓			✓	✓	
IBM1146	1146	✓			✓	✓	
IBM1147	1147	✓			✓	✓	
IBM1148	1148	✓			✓	✓	
IBM4971	4971	✓					
ISO8859-1	819		✓	✓	✓		✓
ISO8859-2	912		✓	✓	✓		✓
ISO8859-3	913		✓	✓	✓		✓
ISO8859-4	914		✓	✓	✓		✓
ISO8859-5	915		✓	✓	✓		✓
ISO8859-6	1089		✓	✓	✓		✓
ISO8859-7	813		✓	✓	✓		✓
ISO8859-8	916		✓	✓	✓		✓
ISO8859-9	920		✓	✓	✓		✓
ISO8859-10			✓	✓	✓		✓
ISO8859-13	921		✓	✓	✓		✓
ISO8859-14			✓	✓	✓		✓
ISO8859-15	923		✓	✓	✓		✓
PC437	437			✓	✓		

Code Page	CCSID	z/OS	UNIX	Windows	OS/400		HP NonStop
					HFS	LIB	
PC737	737			✓	✓		
PC775	775			✓	✓		
PC850	850			✓	✓		
PC852	852			✓	✓		
PC855	855			✓	✓		
PC857	857			✓	✓		
PC860	860			✓	✓		
PC861	861			✓	✓		
PC862	862			✓	✓		
PC863	863			✓	✓		
PC864	864			✓	✓		
PC865	865			✓	✓		
PC866	866			✓	✓		
PC869	869			✓	✓		
PC874	874			✓	✓		
WIN1250	1250			✓	✓		
WIN1251	1251			✓	✓		
WIN1252	1252			✓	✓		
WIN1253	1253			✓	✓		
WIN1254	1254			✓	✓		
WIN1255	1255			✓	✓		
WIN1256	1256			✓	✓		
WIN1257	1257			✓	✓		
WIN1258	1258			✓	✓		

Table 7.3 Character Code Pages

## 7.4 UTT Files

Table 7.4 identifies the Universal Translate Table (UTT) files that are used to translate between Unicode and the local single-byte code page.

Operating System	UTT File Location
z/OS	UTT files are located in the library allocated to the <b>UNVNLS</b> ddname. <i>codepage</i> is the member name of the UTT file.
UNIX	UTT files are located in the <b>nls</b> subdirectory of the installation directory. <i>codepage</i> is the base file name of the UTT file. All UTT files end with an extension of <b>.utt</b> .
Windows	UTT files are located in the <b>NLS</b> subdirectory of the installation directory. <i>codepage</i> is the base file name of the UTT file. All UTT files end with an extension of <b>.utt</b> .

Table 7.4 UTT File Locations

## 7.5 SSL Cipher Suites

Table 7.5, below, identifies all of SSL cipher suites provided by Stonebranch Inc. for use with Universal Enterprise Monitor.

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

Table 7.5 SSL Cipher Suites

## 7.6 Event Definition Parameters - General

Table 7.6, below, describes the general event definition parameters.

The parameters that make up the record's key are underlined.

Parameter	Description	Remarks
<u>Event ID</u>	Unique identifier for the event definition.	Length of the handler ID cannot be greater than 32 characters. This value is case-sensitive.
<u>Assigned UEM Component</u>	Name of an event-driven Universal Event Monitor Server component responsible for monitoring the event.	
Enabled Flag	A <i>true / false</i> value that determines whether an event-driven UEM Server processes the event definition.	If <i>true</i> , the event is checked periodically by its assigned event-driven UEM Server to see if it should be made active or inactive. If <i>false</i> , the event, while still included in a UEM Server's list of assigned components, is not checked to see if it should be made active. [Default is <i>true</i> .]
Active Flag	A <i>yes / no</i> value that indicates whether UEM Server is monitoring the event.	If <i>yes</i> , the current date/time is greater than the defined <b>Activation Date/Time</b> , but less than the defined <b>Inactivation Date/Time</b> . If <i>no</i> , the current date/time is greater than the defined <b>Inactivation Date/Time</b> . This parameter is set only through UEM Server. It cannot be set using the UEMLoad utility. [Default is <i>no</i> .]
Event Type	System event represented by the event definition.	Value corresponds to one of the supported event types. (Currently, <b>FILE</b> is the only support event type.)
Activation Date	Date on which UEM will begin checking for the occurrence of the system event represented by the event definition.	Specified using the format <i>YYYY.MM.DD</i> . [Default is current date.]
Activation Time	Time on the <b>Activation Date</b> at which UEM will begin checking for the occurrence of the system event represented by the event definition.	Specified using the format <i>HH:MM</i> . [Default is current time.]
Inactivation Date	Date on which UEM will stop checking for the occurrence of the system event represented by the event definition.	Specified using the format <i>YYYY.MM.DD</i> . [Default is <i>2038.01.16</i> for event definitions stored with the UEMLoad utility.]
Inactivation Time	Time on the <b>Inactivation Date</b> at which UEM will stop checking for the occurrence of the system event represented by the event definition.	Specified using the format <i>HH:MM</i> . [Default is <i>23:59</i> .]

Parameter	Description	Remarks
Tracking Interval	Frequency with which UEM tests for the completion of an occurrence of the system event represented by the event definition.	Value is expressed in seconds. Value depends on the event type. Some system events will be considered complete as soon as their occurrence is detected by UEM. For these events, this parameter should be 0 (zero). [Default is 10.]
Triggered Event Handler ID	ID of an event handler database record whose specified actions should be taken when an event occurrence satisfies its defined completion criteria.	Value is case sensitive. Value can be blank, in which case the default value specified in the UEM Server configuration (empty string for Windows, <i>NONE</i> for UNIX) will be used.
Rejected Event Handler ID	ID of an event handler database record whose specified actions should be taken if one or more tracked event occurrences fail to complete before the event is made inactive by UEM.	Value is case sensitive. Value can be blank, in which case the default value specified in the UEM Server configuration (empty string for Windows, <i>NONE</i> for UNIX) will be used.
Expired Event Handler ID	ID of an event handler database record whose specified actions should be taken if an event is made inactive with no occurrence of the system event represented by the event definition being detected by UEM.	Value is case sensitive. Value can be blank, in which case the default value specified in the UEM Server configuration (empty string for Windows, <i>NONE</i> for UNIX) will be used.
Event Handler Options	String that contains one or more parameters that UEM adds to the command line it constructs to execute an event handler's specified command or script.	Added to the command line for every process executed for the event handlers referenced by this event definition. This includes those event handlers referenced by the Triggered Event Handler ID, Rejected Event Handler ID, and Expired Event Handler ID fields.  If a value is stored in the Options field of any event handler that this event definition references, the value stored in this field is placed after the value of that field when UEM constructs the command line.
Last Update Date/Time	Date and time the event definition was created or last modified.	
Last Update User	ID of the user account that created or most recently updated the event definition.	

Table 7.6 Event Definition Parameters - General

## 7.7 Event Definition Parameters - Event Type FILE

Table 7.7, below, describes the parameters that are available in event definitions for an Event Type of FILE.

Parameter	Description	Remarks
File Specification	Complete path of the file to watch.	File specification can contain wildcards: <ul style="list-style-type: none"> <li>• ? is available to match up to 1 character.</li> <li>• * can be used to match 0 or more characters.</li> </ul>
Minimum File Size	Smallest size that a file can be in order to be considered complete.	Value of 0 indicates that a file of any size is acceptable. To specify a storage unit for this value, add either of the following after the file size: <ul style="list-style-type: none"> <li>• (b)ytes</li> <li>• (k)ilobytes</li> <li>• (m)egabytes</li> <li>• (g)igabtyes</li> </ul> If a storage unit is not specified, UEM assumes the file size is given in bytes.
Rename File	Indicates whether or not UEM Server will rename the file prior to executing the triggered event handler. This will prevent multiple detections of the same file.	Acceptable values are <i>yes</i> or <i>no</i> . [Default is <i>yes</i> .]

Parameter	Description	Remarks
Rename File Specification	Format used by UEM to rename a file.	<p>Complete file name or a file mask that accepts specific well-known variables that will be substituted by UEM Server at run time. When specified, variables must be in the format <b>\$(var)</b>.</p> <p>The following variables are available:</p> <ul style="list-style-type: none"> <li>• <b>\$(compname)</b> – component name</li> <li>• <b>\$(compid)</b> - component ID</li> <li>• <b>\$(date)</b> - current date in the format YYYYMMDD</li> <li>• <b>\$(time)</b> - current time, n the format HHMMSS</li> <li>• <b>\$(origname)</b> - original base file name, minus its last extension</li> <li>• <b>\$(origext)</b> - original file extension</li> <li>• <b>\$(seqnum)</b> - sequence number that starts at 0 when a UEM Server component is started, and is then incremented by 1 for each file renamed.</li> </ul> <p>If no path is specified, the file is simply renamed. Otherwise, the file can be renamed, moved from its original location, and placed in the path specified by this parameter.</p> <p>If no value is specified, a default value of <b>\$(compname) . \$(compid) . \$(date) . \$(seqnum)</b>, specified in the UEM Server configuration, will be used.</p>

Table 7.7 Event Definition Parameters - Event Type FILE



## 7.8 Event Handler Parameters

Table 7.8, below, describes the event handler parameters.

The parameters that make up the record's key are underlined.

Parameter	Description	Remarks
Encrypted File	Complete path to a file encrypted with Universal Encrypt. UEM reads the contents of this file to obtain the User ID and Password of a user account, which is then used to establish a security context in which the event handler process is executed.	A Password may not be necessary on UNIX, depending on the existing EVENT_HANDLER_ACL entries. Storing the path to an encrypted file allows the file to be shared between Universal Product applications. This makes it unnecessary to update individual event handler records whenever account information is updated or security requirements change.
Handler Actions	Command or script executed by UEM Server for this handler.	For records with a Handler Type of <i>cmd</i> , this parameter contains the syntax required to execute a system command or other command line application. For records with a Handler Type of <i>script</i> , this parameter contains one or more system commands that are executed collectively as a batch script.
<u>Handler ID</u>	Unique identifier for the event handler.	Length of the handler ID cannot be greater than 32 characters. Value is case-sensitive.
Handler Type	Type of action that will be executed.	If the Handler Actions field contains a system command or other command line application, this field has a value of <i>cmd</i> . If the Handler Actions field contains a series of system commands that should be executed as a script file, this field has a value of <i>script</i> .
Last Update Date/Time	Date and time that the event handler was created or last modified.	
Last Update User	ID of the user account that created or most recently updated the event handler.	
Maximum Acceptable Return Code	Highest value that the handler process can return to still be considered as having executed successfully.	If the value returned by the action is less than or equal to the value specified by this parameter, UEM will report a successful completion of the handler process. If the value returned by the action is greater than the value specified by this parameter, UEM will report a failure of the handler process.

Parameter	Description	Remarks
Options	String with one or more parameters that UEM adds to the command line that it constructs in order to execute a specified command or script.	<p>This parameter serves the same purpose as the event definition's Handler Options parameter, but differs in scope.</p> <p>An event definition's Handler Options are applied to every handler process executed on its behalf. An event definition's Handler Options may specify global handler execution options (for example, output redirection) or may customize event handler behavior for that specific event.</p> <p>An event handler's Options value, on the other hand, is applied every time the handler is executed, regardless of which event definition causes it to execute.</p> <p>When command line options are specified in the event definition (via Handler Options) and in the event handler (via Options), both are used. However, the event handler's Options value is added first.</p>
Password	Password of the user account specified in the User ID parameter.	<p>If the UEM Server processing this record is configured to run without security, this parameter is ignored.</p> <p>On UNIX-based systems, a UACL entry can be defined that permits handler execution for some user accounts without requiring authentication. In such a situation, this parameter is ignored.</p> <p>This parameter is encrypted prior to being stored in the database.</p>
Script Type	Script interpreter to invoke to execute script statements contained in the Handler Actions parameter.	<p>On UNIX-based systems, this value may be blank, in which case it will default to <i>sh</i>. A script's execution shell also can be specified as the first statement contained in the Handler Actions parameter.</p> <p>On Windows-based systems, this value will be used as a file extension for the temporary script constructed by UEM Server in order to execute the script statements specified in the Handler Actions parameter. To insure proper execution of the script, a file association should exist on the target system that specifies the application with which the script should be executed.</p> <p>This parameter is ignored if the Handler Type has a value of <i>cmd</i>.</p>
User ID	ID of the user account in whose security context the handler actions will be run.	If the UEM Server processing this record is configured to run without security, the value in this parameter is ignored.

Table 7.8 Event Handler Parameters

---

# Appendix A

# Customer Support

---

Stonebranch, Inc. provides customer support, via telephone and e-mail, for Universal Event Monitor 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](mailto:support@stonebranch.com)**

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

**[www.stonebranch.com](http://www.stonebranch.com)**





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

