



STONEBRANCH

Indesca / Infitran Installation Guide

Indesca / Infitran
Version 4.1.0

Indesca / Infitran

Installation Guide

Indesca / Infitran 4.1.0

Document Name	Indesca / Infitran 4.1.0 Installation Guide				
Document ID	unv-install-4100				
Products	z/OS	UNIX	Windows	OS/400	HP NonStop*
Universal Products	√	√	√	√	√
* Universal Products 2.1.1 is installed on the HP NonStop operating system.					

Stonebranch Documentation Policy

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

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

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

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

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

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

© 2003-2010 by Stonebranch, Inc.

All rights reserved.



Summary of Changes

Changes for Indesca / Infitran 4.1.0 Installation Guide (unv-install-4100) February 10, 2010

Indesca / Infitran 4.1.0.0

- Replaced the following illustrations in [Chapter 2 z/OS Installation](#):
 - [Figure 2.2 Universal Products for z/OS - Sample JCL for Allocating and Cataloging XMIT Data Sets on z/OS](#)
 - [Figure 2.3 Universal Products for z/OS - Sample JCL for Unpacking XMIT Data Sets on z/OS](#)
- Added Universal Products 3.2.0 package, and removed Universal Products 2.2.0 package, in Section [2.7 Universal Products for z/OS – SMP/E Installation of Universal Products](#).
- Removed the Universal Products 2.2.0 Upgrade, Existing CSI section from [Chapter 2 z/OS Installation](#).
- Added Section [2.7.2 Universal Products 3.2.0 Upgrade, Existing CSI](#).
- Modified Section [3.6 Universal Products for z/OS USS – SMP/E Installation](#):
 - Modified Section [3.6.1 New Install, New CSI](#).
 - Added Section [3.6.2 Universal Products for z/OS USS 3.2.0 Upgrade, Existing CSI](#).
- Renamed the following components in [Chapter 4 Windows Installation](#):
 - Renamed Universal Activity Monitor as I-Activity Monitor.
 - Renamed Universal Management Console as I-Management Console.
 - Renamed UEC Administrator as I-Administrator.
- Added [Universal Products 4.1.0](#) in Section [6.6.7 Product Removal](#).
- Added two job log print file entries in [Table 6.4 Universal Products for OS/400 - Product Library \(UNVPRD410\) Object Inventory List](#).

Changes for Universal Products 3.2.0 Installation Guide (unv-install-3209) November 2, 2009

Universal Products 3.2.0.9

- Removed information describing support of the zFS file system for Universal Enterprise Controller in [Chapter 2 z/OS Installation](#).

Changes for Universal Products 3.2.0 Installation Guide (unv-install-3208) October 30, 2009

Universal Products for SOA 3.2.0.4

- Updated Universal Products for SOA file inventory lists, including relocation of **UAC.xml** file, inlocation of **UAC.xml** in:
 - Section [4.25 Universal Products for SOA for Windows – File Inventory Lists](#)
 - Section [5.15 Universal Products for SOA for UNIX – File Inventory Lists](#)

Changes for Universal Products 3.2.0 Installation Guide (unv-install-3208) October 28, 2009

Universal Products 3.2.0.8

- Removed information describing support of the zFS file system for Universal Enterprise Controller in [Chapter 2 z/OS Installation](#).

Changes for Universal Products 3.2.0 Installation Guide (unv-install-3207) September 8, 2009

- Specified, in all Universal Products 3.2.0 chapters, that a Universal Broker must run on all systems in which a Universal Products component is running.
- Specified the requirement for an OMVS segment in Section [2.3 Universal Products for z/OS – Installation Requirements](#).
- Modified file system requirements in Section [3.3 Universal Products for z/OS USS – Installation Requirements](#).

Universal Products 3.2.0.8

- Included information about support of the zFS file system in:
 - [Chapter 2 z/OS Installation](#)
 - [Chapter 3 z/OS USS Installation](#)

- Added the following sub-sections in Section [2.9.5 Universal Products Database Configuration](#):
 - [HFS Configuration](#)
 - [zFS Configuration](#)
 - [Memory Management](#)
- Added Section [4.12 Universal Enterprise Controller – Database Configuration](#) in [Chapter 4 Windows Installation](#).
- Modified Section [5.6.3 Universal Products for Solaris Installation](#) to specify that the Universal Connector package, **UNVusp**, is not included in the Solaris Intel-based package.

Changes for Universal Products 3.2.0 Installation Guide (unv-install-3206) July 29, 2009

Universal Products 3.2.0.1 for OS/400

- Modified [Chapter 6 OS/400 Installation](#) for upgrade from Universal Products 3.1.1 for OS/400 to Universal Products 3.2.0 for OS/400, including:
 - Changed the following OS/400 names throughout the chapter:
 - Universal Broker subsystem name from **UBROKER** to **UNVUBR320**.
 - Universal Broker user profile name from **UBROKER** to **UNVUBR320**.
 - Universal Products installation library name from **UNIVERSAL** to **UNVPRD320**.
 - Universal Products spool library name from **UNVSPPOOL** to **UNVSPL320**.
 - Universal Products temporary directory from **UNVTMP** to **UNVTMP320**.
 - Specified local Broker requirement in Section [6.1 Overview](#).
 - Added the following sections:
 - Section [6.1.1 Naming Conventions](#).
 - Section [6.6.3 Re-Installation of Same Release](#).
 - Section [6.6.6 UCHGRLS \(Change Release Tag\) Program](#).
 - Section [6.6.7 Product Removal](#).
 - Section [6.7.1 Multiple-Installation Implementation](#).
 - Section [6.7.10 National Language Customization](#).
 - Modified Section [6.6.4 Upgrade Installation to New Release](#):
 - Added information for a [Currently Installed Release](#).
 - Added [Pre-Production Tasks](#).

Changes for Universal Products 3.2.0 Installation Guide (unv-install-3205) April 1, 2009

- Added manual mounts information for HFS data sets in Section [2.9.5 Universal Products Database Configuration](#) of [Chapter 2 z/OS Installation](#).

- Identified separate default installation paths for 32-bit and 64-bit systems in Section [4.1.1 Pre-Installation Guidelines](#) of [Chapter 4 Windows Installation](#).
- Added [Appendix A Licenses and Copyrights](#).

Universal Connector 3.2.0.1

- Added DD statement for SAP RFC file used by Universal Connector in [Table 2.5 Universal Broker for z/OS - Data Set Access Levels](#).

Changes for Universal Products 3.2.0 Installation Guide (unv-install-3204) December 17, 2008

- Added the recommendation for SMP/E back-up in Section [2.3.2 SMP/E](#) of [Chapter 2 z/OS Installation](#).
- Added procedure for extracting installation files when installing via command line interface for all Universal Products packages in [Chapter 4 Windows Installation](#).
- Added [4.8.1 Product Compatibility](#) in Section [4.8 Universal Enterprise Controller Package](#).

Universal Products for SOA 3.2.0.2

- Changed the following names throughout [Chapter 4 Windows Installation](#):
 - Universal Products for SOA (previously Universal Command Agent for SOA)
 - Universal Products for SOA distribution file
 - Windows installation file
- Added note and recommendation about upgrade procedures to Section [4.21 Universal Products for SOA for Windows – Pre-Installation / Upgrade Backups](#).
- Changed the following names throughout [Chapter 5 UNIX Installation](#):
 - Universal Products for SOA (previously Universal Command Agent for SOA)
 - Universal Products for SOA distribution file
 - AIX and Linux install script
 - Linux install log
- Added the following sections in [Chapter 5 UNIX Installation](#):
 - [Removing Universal Products for SOA for AIX](#).
 - [Listing Universal Products for SOA for AIX Information](#).
 - [Removing Universal Products for SOA for Linux](#).
 - [Listing Universal Products for SOA for Linux Information](#).

Changes for Universal Products 3.2.0 Installation Guide (unv-install-3203) October 17, 2008

- Added Section [2.15 Universal Products for z/OS - Performance Guidelines](#).

Universal Products 3.2.0.2

- Added information for Linux on Itanium systems in Section [5.1.1 Installation Methods](#).
- Added [Itanium Systems](#) in the [Extracting the Universal Products for Linux Installation Files](#) section of Section [5.6.4 Universal Products for Linux Installation](#).

Changes for Universal Products 3.2.0 Installation Guide

(unv-install-3202)

September 5, 2008

- Added miscellaneous clarifications and corrections.

Changes for Universal Products 3.2.0 Installation Guide

(unv-install-3201)

September 2, 2008

- Added toll-free telephone number for North America in [Appendix B Customer Support](#).
- Added [Removing Universal Products for UNIX](#) in Section [5.6.5 Universal Products for UNIX Installation \(Generic\)](#).

HP NonStop 2.1.1.3

- Specified separate distribution files for NSK S-series system and Integrity system in Section [7.4 Universal Command for HP NonStop – Product Distribution File](#) of [Chapter 7 HP NonStop Installation](#).

Changes for Universal Products 3.2.0 Installation Guide

(unv-install-320)

May 16, 2008

- Updated installation information and procedures for UNIX, Windows, and z/OS.
- Modified [Chapter 2 z/OS Installation](#) with the following information:
 - Updated [Universal Products for z/OS – Installation Requirements](#).
 - Added information about Universal Enterprise Controller database implementation
Added z/OS USS distribution files to [Table 2.3 Universal Products for z/OS - Installation Files](#).
 - Added Section [2.13 Universal Products for z/OS – Configuration of z/OS System SSL](#).
 - Modified space requirements in:
 - [Table 2.13 Universal Products for z/OS - SMP/E Data Sets](#)
 - [Figure 2.2 Universal Products for z/OS - Sample JCL for Allocating and Cataloging XMIT Data Sets on z/OS](#).
 - [Figure 2.3 Universal Products for z/OS - Sample JCL for Unpacking XMIT Data Sets on z/OS](#).

- Modified [Chapter 3 z/OS USS Installation](#).
- Modified [Chapter 4 Windows Installation](#) with the following information:
 - Added [Section 4.19 Universal Products for SOA for Windows Package](#).
 - Modified the following file inventory lists:
 - [Universal Products for Windows – File Inventory Lists](#)
 - [Universal Enterprise Controller – File Inventory Lists](#).
 - [UEC Client Applications – File Inventory List](#)
- Modified [Chapter 5 UNIX Installation](#) with the following information:
 - Updated Product Distribution File information for all operating systems.
 - Added [Universal Products for SOA for UNIX Package](#) to [Chapter 5 UNIX Installation](#).
 - Modified information in the following tables in [Section 5.6.1 Universal Products for AIX Installation](#):
 - [Table 5.2 Universal Products for AIX - Product Distribution File Contents](#)
 - [Table 5.3 Universal Products AIX - Package Filesets](#)
 - Added [Examples of Customized PAM Configuration Files](#) in [Linux PAM Customization](#) of [Section 5.6.4 Universal Products for Linux Installation](#).

Changes for Universal Products 3.1.1 Installation Guide (unv-install-31110) February 28, 2007

- Created [Section 5.12 Data Set Inventory](#) from information previously included in [Section 5.3 Requirements](#).
- Created [Section 6 z/OS USS](#) from information previously included in [Section 5 z/OS](#).
- Removed [USAP Customization](#) from [Section 6.7 Universal Products Customization](#) in [Section 6 z/OS USS](#).
- Added customer support telephone number for Europe to [Appendix A Customer Support](#).

Universal Products 3.1.1.11

- Specified, in the [Requirements](#) section for each platform, that the user performing the installation must be able to write to the directory from which the installation is launched.
- Added un-attended install / un-install information in [Section 1.1.6.1 Installing Universal Command From the Command Line](#) in [Section 1 Microsoft Windows](#).
- Added [Section 5.7.5 Cumulative PTF Maintenance](#) in [Section 5 z/OS](#).
- Updated [Section 6.8 File Inventory List](#) in [Section 6 z/OS USS](#).

Changes for Universal Products 3.1.1 Installation Guide (unv-install-3119) December 15, 2006

- Added information for downloading and FTPing distribution file for an upgrade to maintenance release in Section 2.3 Product Distribution Files.

Changes for UNV Release 3.1.1.8 November 15, 2006

- Added List of Figures and List of Tables.
- Added Customer Support.
- Deleted footnote, from Organization in Document Structure, which specified that installation of Universal Connector was not included in this guide (USAP now is installed as part of the Universal Products package).
- Replaced Release 1.2.1 information with Release 3.1.1 information in Section 2 OS/400.
- Added Section 2.2.1 Various Libraries and Logs.
- Updated Section 2.2 Installation Requirements.
- Added Section 2.4.4 Upgrade Installation for New Release.
- Added procedures for uninstallation of Universal Encrypt 1.2.1 and Universal Command 1.2.1 in Section 2.4.4.2 Uninstall.
- Added Section 2.5 OS/400 System Customization.
- Updated Section 2.6 Object Inventory List.
- Updated Section 4.6.4 Linux Installation of Universal Products to include separate information for Intel-Based Systems and IBM S/390 and zSeries Systems.

January 30, 2006 Release

- Add notes regarding installation on Windows Server 2003 using a Remote Desktop session.
- Add z/OS Universal Event Monitor security profile information.
- Add z/OS Universal Command security profile for standard I/O data set access.

September 9, 2005 Release

- Changed z/OS upgrade JCL to provide for a simpler upgrade process.

April 30, 2005 Release

- Addition of the Universal Certificate utility to create X.509 certificates.
- New Broker configuration option (UCMD_STC_SUPPORT) makes installing the z/OS SMF exit routine UNVACTRT optional.

October 31, 2004

- Universal Products 3.1.0 version upgrade for UNIX, Windows and z/OS.
- Changed the minimum requirements for the Java Runtime from 1.3.1_07 to 1.4.2_01
- Addition of Linux RPM package.

March 19, 2004

- Removed references to installation of UEC Applications in the Universal Enterprise Controller chapter for Windows.
- Added a Universal Enterprise Client Applications installation for Windows section.
- Included section containing special instructions for upgrades from Universal Enterprise Controller v110-3 or earlier.
- Changed the minimum requirement of the Sun JRE from 1.3.1_06 to 1.3.1_07, which is the version of JRE that fixes problems with drawing and font issues inherent in the 1.3.1_06 version and older.

February 13, 2004

- Corrected command line parameter for specifying installation of Universal Data Mover Manager.

January 27, 2004

- Added section for Linux PAM configuration.

January 15, 2004

- Minor terminology corrections in the HP NonStop chapter.

November 24, 2003

- Consolidated Universal Command and Universal Enterprise Controller installation documentation into this single Universal Products Installation Guide.
- Added installation procedures for Universal Data Mover 1.1.0.
- Change the minimum IBM OS/390 release from OS/390 2.4 to OS/390 2.7.

Contents

Summary of Changes	5
Contents	13
List of Figures	27
List of Tables	29
Preface	33
Document Structure	33
Conventions	33
Vendor References	34
Document Organization	35
Chapter 1 Document Overview	36
1.1 Overview	36
1.2 Installation Information	37
1.2.1 Product Distribution	37
1.2.2 Product Licensing	37
1.2.3 Additional Documentation	38
Chapter 2 z/OS Installation	39
2.1 Overview	39
2.1.1 Installation Procedures	40
2.2 Universal Products for z/OS Package	41
2.2.1 Product Compatibility	41
2.3 Universal Products for z/OS – Installation Requirements	42

2.3.1	Data Set Space Requirements	42
2.3.2	SMP/E	43
2.4	Universal Products for z/OS – Installation Upgrades	44
2.5	Universal Products for z/OS – Product Distribution File	45
2.5.1	Distribution File Format	45
2.5.2	Distribution File Contents	46
2.6	Universal Products for z/OS – Transferring Installation Files to z/OS	48
2.6.1	Express/OS Method	49
2.6.2	Manual Method	50
	Extract the Installation Files	50
	Allocate z/OS Data Sets	52
	Transfer the Installation Files	57
	Unpack the Data Sets	58
2.7	Universal Products for z/OS – SMP/E Installation of Universal Products	65
2.7.1	New Install, New CSI	66
2.7.2	Universal Products 3.2.0 Upgrade, Existing CSI	69
2.7.3	Universal Products 3.1.1 Upgrade, Existing CSI	71
2.7.4	Universal Products 3.1.0 Upgrade, Existing CSI	74
2.8	Universal Products for z/OS – Cumulative PTF Maintenance	77
2.8.1	Installation Procedures	77
2.8.2	Cumulative PTF Maintenance Distribution File	77
	Distribution File Content	77
2.8.3	Transferring a Cumulative PTF File to z/OS	78
	Express/OS Method	78
	Manual Method	79
2.8.4	Installing a Cumulative PTF File	80
2.9	Universal Products for z/OS – Configuration	81
2.9.1	Started Task Configuration	81
	Define Started Procedure User and Group Profiles	82
	Permitting Access to Security Resources	83
	Data Set Access	84
2.9.2	Associate Started Tasks with User and Group Profiles	85
	STARTED Class Profile	85
	Started Procedures Table	86
2.9.3	Library and Program Authorization	87
	APF Authorization	87
	Program Control	88
2.9.4	SMF Exit Routine	90
	Defining the SMF SYSSTC.IEFACTRT Exit Point	90
	Installing the UNVACTRT SMF Exit Routine	91
2.9.5	Universal Products Database Configuration	92
	z/OS UNIX File System Introduction	92

HFS Configuration	93
zFS Configuration	93
Mounting and Unmounting the Databases	94
Memory Management	96
2.10 Universal Products for z/OS – Customization	97
2.10.1 Universal Broker Customization	97
Configuration	97
JCL Procedure	97
2.10.2 Universal Certificate Customization	98
JCL Procedure	98
2.10.3 Universal Command Manager Customization	99
Configuration	99
Product License	99
JCL Procedure	99
2.10.4 Universal Command Server Customization	100
Configuration	100
2.10.5 Universal Connector Customization	101
Configuration	101
SAP RFC Configuration	101
Product License	101
JCL Procedure	101
2.10.6 Universal Control Manager Customization	102
Configuration	102
JCL Procedure	102
2.10.7 Universal Control Server Customization	103
Configuration	103
2.10.8 Universal Data Mover Manager Customization	104
Configuration	104
Product License	104
JCL Procedure	104
2.10.9 Universal Data Mover Server Customization	105
Configuration	105
2.10.10 Universal Enterprise Controller Customization	106
Configuration	106
Product License	106
JCL Procedure	106
2.10.11 Universal Event Monitor Manager Customization	107
Configuration	107
JCL Procedure	107
2.10.12 Universal Query Customization	107
Configuration	107
2.11 Universal Products for z/OS – Time Zone Environment Variable	108

2.12 Universal Products for z/OS – TCP/IP Configuration	109
2.12.1 Specifying TCP/IP Affinity	109
2.12.2 TCP/IP Resource Protection	110
Stack Access Control	110
Port Access Control	110
Network Access Control	110
Socket Option Access Control	110
2.13 Universal Products for z/OS – Configuration of z/OS System SSL	111
2.13.1 Integrated Cryptographic Service Facility (ICSF)	112
2.13.2 Universal Broker Digital Certificate (RACF) Set-up	113
2.14 Universal Products for z/OS – Configuration of Universal Products Security	115
2.14.1 RACF Class	115
2.14.2 ACF2 CLASS	117
2.14.3 Universal Command Security	118
Universal Command Remote Access Profile	118
Universal Command Standard I/O Access Profile	119
Universal Command Security Profile Definitions	120
2.14.4 Universal Control Security	122
Universal Control Remote Access Profile Format	122
Universal Control Security Profile Definition	123
2.14.5 Universal Event Monitor Security	124
Universal Event Monitor Remote Access Profile Format	124
Universal Event Monitor Security Profile Definition	125
2.15 Universal Products for z/OS - Performance Guidelines	126
2.15.1 UNIX System Services and Language Environment	126
2.15.2 Universal Products Managers	127
2.15.3 Universal Broker and Universal Products Servers	128
2.15.4 Universal Enterprise Controller	128
2.16 Universal Products for z/OS – Data Set Inventory	129
2.16.1 SMP/E Data Sets	129
2.16.2 Non-SMP/E Data Sets	130
Chapter 3 z/OS USS Installation	131
3.1 Overview	131
3.2 Universal Products for z/OS USS Package	132
3.2.1 Product Compatibility	132
3.3 Universal Products for z/OS USS – Installation Requirements	133
3.3.1 Data Set Space Requirements	133
3.3.2 SMP/E	134
3.4 Universal Products for z/OS USS – Installation Upgrades	135
3.4.1 Universal Products for z/OS USS 3.2.0	135

3.5 Universal Products for z/OS USS – Product Distribution File	136
3.6 Universal Products for z/OS USS – SMP/E Installation	137
3.6.1 New Install, New CSI	137
3.6.2 Universal Products for z/OS USS 3.2.0 Upgrade, Existing CSI	139
3.7 Universal Products for z/OS USS – Customization	140
3.7.1 Universal Command Manager for z/OS USS Customization	140
Configuration	140
Product License	140
3.7.2 Universal Control Manager for z/OS USS Customization	140
Configuration	140
3.7.3 Universal Data Mover Manager for z/OS USS Customization	141
Configuration	141
Product License	141
3.7.4 Universal Event Monitor Manager for z/OS USS Customization	141
Configuration	141
3.7.5 Universal Query for z/OS USS Customization	141
Configuration	141
3.8 Universal Products for z/OS USS – Data Set Inventory	142
3.8.1 SMP/E Data Sets	142
3.8.2 z/OS UNIX Directories	142
Chapter 4 Windows Installation	143
4.1 Overview	143
4.1.1 Pre-Installation Guidelines	144
Installation Paths	144
4.1.2 Product Distribution Files	144
Distribution Files Format	145
4.1.3 Windows Installer	146
Installing Windows Installer with a Universal Products Package	146
Installing Windows Installer Separately from a Universal Products Package	148
4.2 Universal Products for Windows Package	149
4.2.1 Product Compatibility	149
4.3 Universal Products for Windows – Installation Requirements	150
4.3.1 Spool Directory	150
Location	150
Space	151
Security	151
4.4 Universal Products for Windows – Installation	152
4.4.1 Installing Universal Products via the Windows Installer Graphical Interface	152

4.4.2	Modifying a Universal Products Installation via the Windows Installer Graphical Interface	157
	Adding or Removing Universal Products Components	157
	Repairing a Corrupted Universal Products Installation	161
	Removing a Universal Products Installation	163
4.4.3	Installing Universal Products via the Windows Installer Command Line Interface	165
	Windows Installer Command Line Parameters	167
	Windows Installer Command Line Syntax	171
	Windows Installer Installation Commands	172
	Detecting the Completion of a Silent Install	173
	Using the Distribution File for a Silent Install	173
4.4.4	Modifying a Universal Products Installation via the Windows Installer Command Line Interface	174
	Adding or Removing Universal Products Components	174
	Repairing a Corrupted Universal Products Installation	174
	Removing a Universal Products Installation	174
4.5	Universal Products for Windows – Licensing	175
4.5.1	Licensing Methods	175
4.5.2	Licensing via Universal Configuration Manager	175
4.6	Universal Products for Windows – Installing on 64-bit Windows Editions	177
4.6.1	Modifying the Working Folder for Universal Server Components	177
4.6.2	Applications Installed in the Windows System Folder	180
	Example 1	180
	Example 2	180
4.7	Universal Products for Windows – File Inventory Lists	181
	Universal Broker	182
	Universal Command Manager	182
	Universal Command Server	183
	Universal Control Manager	183
	Universal Control Server	184
	Universal Data Mover Manager	184
	Universal Data Mover Server	185
	Universal Event Monitor Manager	185
	Universal Event Monitor Server	186
	Universal Configuration Manager	186
	Universal Copy	186
	Universal Encrypt	187
	Universal Event Log Dump	187
	Universal Message Translator	187
	Universal Products Install Merge	187
	Universal Query	188

Universal Spool Utilities	188
System Files	189
4.8 Universal Enterprise Controller Package	190
4.8.1 Product Compatibility	190
4.9 Universal Enterprise Controller – Installation Requirements	191
4.10 Universal Enterprise Controller – Installation	192
4.10.1 Installing Universal Enterprise Controller via the Windows Installer Graphical Interface	192
4.10.2 Modifying a Universal Enterprise Controller Installation via the Windows Installer Graphical Interface	197
Adding or Removing Universal Enterprise Controller Components	197
Repairing a Corrupted Universal Enterprise Controller Installation	201
Removing a Universal Enterprise Controller Installation	202
4.10.3 Installing Universal Enterprise Controller via the Windows Installer Command Line Interface	204
Windows Installer Parameters	206
Windows Installer Command Line Syntax	208
Windows Installer Installation Commands	209
Detecting the Completion of Silent Installs	210
Using the Distribution File for a Silent Install	210
4.10.4 Modifying a Universal Enterprise Controller Installation via the Windows Installer Command Line Interface	211
Adding or Removing Universal Enterprise Controller Components	211
Repairing a Corrupted Universal Enterprise Controller Installation	211
Removing a Universal Enterprise Controller Installation	211
4.11 Universal Enterprise Controller – Installing on 64-bit Windows Editions	212
4.11.1 Applications Installed in the Windows System Folder	212
4.12 Universal Enterprise Controller – Database Configuration	214
4.13 Universal Enterprise Controller – File Inventory Lists	215
Universal Enterprise Controller	215
Universal Products Install Merge Utility	216
System Files	217
4.14 Universal Enterprise Controller Client Applications Package	218
4.15 UEC Client Applications – Installation Requirements	219
4.16 UEC Client Applications – Installation	220
4.16.1 Installing UEC Client Applications via the Windows Installer Graphical Interface	220
4.16.2 Modifying a UEC Client Applications Installation via the Windows Installer Graphical Interface	227
Adding or Removing UEC Client Applications Components	227
Repairing a Corrupted UEC Client Applications Installation	231
Removing a UEC Client Applications Installation	232

4.16.3	Installing UEC Client Applications via the Windows Installer Command Line Interface	234
	Windows Installer Command Line Parameters	236
	Windows Installer Command Line Syntax	239
	Windows Installer Installation Commands	240
	Detecting the Completion of Silent Installs	241
	Using the Distribution File for a Silent Install	241
4.16.4	Modifying a UEC Client Applications Installation via the Windows Installer Command Line Interface	242
	Adding or Removing Components	242
	Repairing a Corrupted UEC Client Applications Installation	242
	Removing UEC Client Applications from the Command Line	243
4.17	UEC Client Applications – Installing on 64-bit Windows Editions	244
4.18	UEC Client Applications – File Inventory List	245
	I-Administrator	245
	I-Activity Monitor	245
	I-Management Console	246
	System Files	246
4.19	Universal Products for SOA for Windows Package	247
4.20	Universal Products for SOA for Windows – Installation Requirements	248
4.21	Universal Products for SOA for Windows – Pre-Installation / Upgrade Backups	249
4.22	Universal Products for SOA for Windows – Installation	251
4.22.1	Installing Universal Products for SOA via the Windows Installer Graphical Interface	251
4.22.2	Modifying a Universal Products for SOA Installation via the Windows Installer Graphical Interface	255
	Repairing a Corrupted Universal Products for SOA Installation	255
	Removing a Universal Products for SOA Installation	258
4.22.3	Installing Universal Products for SOA via the Windows Installer Command Line Interface	260
	Windows Installer Command Line Parameters	262
	Windows Installer Command Line Syntax	263
	Windows Installer Installation Commands	263
	Detecting the Completion of a Silent Install	264
	Using the Distribution File for a Silent Install	264
4.22.4	Modifying a Universal Products for SOA Installation via the Windows Installer Command Line Interface	265
	Repairing a Corrupted Universal Products for SOA Installation	265
	Removing a Universal Products for SOA Installation	265
4.23	Universal Products for SOA for Windows – Configuring and Starting the Universal Application Container Server	266

4.24 Universal Products for SOA for Windows – Installing on 64-bit Windows Editions	268
4.25 Universal Products for SOA for Windows – File Inventory Lists	269
Universal Application Container Server	269
Universal Application Container	270
Universal Application Interface	270
Sun Java Runtime Environment,1.5.0_14	270
System Files	271
Chapter 5 UNIX Installation	272
5.1 Overview	272
5.1.1 Installation Methods	273
5.2 Universal Products for UNIX Package	274
5.2.1 Product Compatibility	274
5.3 Universal Products for UNIX – Installation Requirements	275
5.3.1 Directories and Files	275
Command Reference Directory	275
Log Directory	276
Trace Directory	276
Spool Directory	277
5.4 Universal Products for UNIX – Installation Upgrades	279
5.4.1 Universal Command 2.2.0	279
Installation Method	279
Product Directories	279
Configuration Files	279
5.4.2 All Universal Product Releases	280
5.5 Universal Products for UNIX – Product Distribution File	281
5.5.1 Distribution File Format	281
5.6 Universal Products for UNIX – Installation	282
5.6.1 Universal Products for AIX Installation	283
Extracting the Universal Products for AIX Installation Files	283
AIX Packages	284
Installing Universal Products for AIX	285
Removing Universal Products for AIX	285
Listing Universal Products for AIX Information	285
5.6.2 Universal Products for HP-UX Installation	286
Pre-Requisite to HP-UX Installation	286
Extracting the Universal Products for HP-UX Installation Files	286
HP-UX Package	287
Installing Universal Products for HP-UX	288
Removing Universal Products for HP-UX	288
Listing Universal Products for HP-UX Information	288

5.6.3 Universal Products for Solaris Installation	289
Extracting the Universal Products for Solaris Installation Files	289
Solaris Package	291
Installing Universal Products for Solaris	292
Removing Universal Products for Solaris	292
Listing Universal Products for Solaris Information	292
5.6.4 Universal Products for Linux Installation	293
Extracting the Universal Products for Linux Installation Files	293
Installing Universal Products for Linux	296
Removing Universal Products for Linux	296
Listing Universal Products for Linux Information	296
Linux PAM Customization	297
5.6.5 Universal Products for UNIX Installation (Generic)	298
Extracting the Universal Products for UNIX Installation Files	298
Installing Universal Products for UNIX	299
Removing Universal Products for UNIX	299
5.7 Universal Products for UNIX – Customization	300
5.7.1 Universal Broker Customization	300
Configuration	300
System Initialization	300
5.7.2 Universal Command Manager Customization	301
Configuration	301
Product License	301
5.7.3 Universal Command Server Customization	301
Configuration	301
5.7.4 Universal Connector Customization	302
Configuration	302
SAP RFC Configuration	302
Product License	302
5.7.5 Universal Control Manager Customization	303
Configuration	303
5.7.6 Universal Control Server Customization	303
Configuration	303
5.7.7 Universal Data Mover Manager Customization	304
Configuration	304
Product License	304
5.7.8 Universal Data Mover Server Customization	304
Configuration	304
5.7.9 Universal Event Monitor Manager Customization	305
Configuration	305
5.7.10 Universal Event Monitor Server Customization	305
Configuration	305
Product License	306

5.7.11 Universal Query Customization	306
Configuration	306
5.8 Universal Products for UNIX – File Inventory Lists	307
Universal Broker	307
Universal Command Manager	308
Universal Command Server	308
Universal Connector	309
Universal Control Manager	309
Universal Control Server	309
Universal Data Mover Manager	310
Universal Data Mover Server	310
Universal Event Monitor Manager	310
Universal Event Monitor Server	311
Universal Certificate	311
Universal Query	311
Universal Spool Utilities	312
5.9 Universal Products for SOA for UNIX Package	313
5.10 Universal Products for SOA for UNIX – Installation Requirements	314
5.11 Universal Products for SOA for UNIX – Licensing	315
5.12 Universal Products for SOA for UNIX – Deployment Options	316
5.12.1 Single-Server Deployment (SSD)	316
5.12.2 Distributed Server Deployment (DSD)	318
5.13 Universal Products for SOA for UNIX – Pre-Installation / Upgrade Backups	319
5.14 Universal Products for SOA for UNIX – Installation	320
5.14.1 Universal Products for SOA – AIX Installation	321
Product Distribution File	321
Unpacking and Installation Procedures	321
Removing Universal Products for SOA for AIX	322
Listing Universal Products for SOA for AIX Information	322
5.14.2 Universal Products for SOA – Linux Installation	323
Product Distribution File	323
Unpacking and Installation Procedures	323
Removing Universal Products for SOA for Linux	324
Listing Universal Products for SOA for Linux Information	324
5.15 Universal Products for SOA for UNIX – File Inventory Lists	325
5.15.1 Parent Directories	325
5.15.2 Product Directories and Files	326
Chapter 6 OS/400 Installation	328
6.1 Overview	328

6.1.1 Naming Conventions	328
6.2 Universal Products for OS/400 Package	329
6.2.1 Product Compatibility	329
6.3 Universal Products for OS/400 – Installation Requirements	330
6.3.1 Libraries and Logs	331
Universal Broker Job Log	331
Command Reference Library	331
Trace File Location Library	332
Spool Library	333
6.4 Universal Products for OS/400 – Product Distribution File	334
6.4.1 Distribution File Format	334
6.4.2 Distribution File Contents	335
6.5 Universal Products for OS/400 – Transferring Universal Products to OS/400	336
6.5.1 Transferring for a Full Release	336
6.5.2 Transferring for a Maintenance Release	337
6.6 Universal Products for OS/400 – Installation	338
6.6.1 New Installation - Default	339
6.6.2 New Installation - Custom	340
6.6.3 Re-Installation of Same Release	341
Re-Installation Using Same Libraries	341
Re-Installation Using Different Library Names	342
6.6.4 Upgrade Installation to New Release	343
Install (Default or Custom)	343
Post-Install Tasks	344
Pre-Production Tasks	344
6.6.5 Upgrade Installation for Maintenance Release	345
6.6.6 UCHGRLS (Change Release Tag) Program	346
Using UCHGRLS	346
UCHGRLS Examples	347
6.6.7 Product Removal	348
Universal Encrypt 1.2.1	349
Universal Command 1.2.1	349
Universal Products 3.1.1	350
Universal Products 3.2.0	350
Universal Products 4.1.0	351
6.7 Universal Products for OS/400 – Customization	352
6.7.1 Multiple-Installation Implementation	352
Universal Broker	353
Active Jobs	353
6.7.2 Universal Broker Customization	354
Configuration	354
System Initialization	354

6.7.3 Universal Command Manager Customization	355
Configuration	355
Product License	355
6.7.4 Universal Command Server Customization	355
Configuration	355
6.7.5 Universal Control Manager Customization	356
Configuration	356
6.7.6 Universal Control Server Customization	356
Configuration	356
6.7.7 Universal Data Mover Manager Customization	357
Configuration	357
Product License	357
6.7.8 Universal Data Mover Server Customization	357
Configuration	357
6.7.9 Universal Query Customization	358
Configuration	358
6.7.10 National Language Customization	358
6.8 Universal Products for OS/400 – Object Inventory Lists	359
Source File Record Lengths	359
Product Library (UNVPRD410)	360
Temporary Library (UNVTMP410)	362
Spool Library (UNVSPL410)	362
Command Reference Library (UNVCMDREF)	362
QSYS Library	362
Chapter 7 HP NonStop Installation	363
7.1 Overview	363
7.2 Universal Command for HP NonStop Components	364
7.2.1 Product Compatibility	364
7.3 Universal Command for HP NonStop – Installation Requirements	365
7.3.1 \$SYSTEM Volume	365
Log Directory	365
Trace Directory	366
7.4 Universal Command for HP NonStop – Product Distribution File	367
7.4.1 NSK S-series System	367
Extracting the Installation Files	367
7.4.2 Integrity System	368
Extracting the Installation Files	368
7.4.3 Distribution File Contents	368
7.5 Universal Command for HP NonStop – Installation	369
7.6 Universal Command for HP NonStop – Customization	370

7.6.1 Universal Broker Customization	370
Configuration	370
System Initialization	370
7.6.2 Universal Command Manager Customization	371
Configuration	371
Product License	371
7.6.3 Universal Command Server Customization	371
Configuration	371
7.6.4 Universal Control Manager Customization	372
Configuration	372
7.6.5 Universal Control Server Customization	372
Configuration	372
7.6.6 Universal Query Customization	373
Configuration	373
7.7 Universal Command for HP NonStop – File Inventory Lists	374
Universal Broker	374
Universal Command Manager	374
Universal Command Server	375
Universal Control Manager	375
Universal Control Server	375
Universal Query	375
Appendix A Licenses and Copyrights	376
A.1 Overview	376
A.2 Berkeley DB License	377
A.3 OpenSSL License	380
A.4 zlib License	383
Appendix B Customer Support	384
Index	385

List of Figures

Chapter 2 z/OS Installation	39
Figure 2.1 Universal Products for z/OS - Express/OS / Installation Directory dialog	51
Figure 2.2 Universal Products for z/OS - Sample JCL for Allocating and Cataloging XMIT Data Sets on z/OS	57
Figure 2.3 Universal Products for z/OS - Sample JCL for Unpacking XMIT Data Sets on z/OS	64
Chapter 4 Windows Installation	143
Figure 4.1 Windows Installer - Properties / Version dialog	147
Figure 4.2 Universal Products - Welcome dialog	153
Figure 4.3 Universal Products - Custom Setup dialog	154
Figure 4.4 Universal Products - Custom Setup / Selecting to Not Install a Component	155
Figure 4.5 Universal Products - Installation Complete dialog	156
Figure 4.6 Universal Products - Add or Remove Programs dialog	157
Figure 4.7 Windows Programs and Features Dialog (Windows Vista, Windows Server 2008 or later)	158
Figure 4.8 Failure to Access Install Files	159
Figure 4.9 Universal Products - Program Maintenance / Modify dialog	159
Figure 4.10 Universal Products - Custom Setup / Modifying an Installed Component	160
Figure 4.11 Universal Products - Program Maintenance / Repair dialog	162
Figure 4.12 Universal Products - Program Maintenance / Remove dialog	163
Figure 4.13 Universal Products - Location to Save Files dialog	165
Figure 4.14 Universal Products for Windows - Windows Installer Command Line Syntax	171
Figure 4.15 Universal Configuration Manager - License Information page	176
Figure 4.16 Universal Products for Windows - Setting the Working Folder during Installation	178
Figure 4.17 Universal Configuration Manager - Universal Broker / Component Definitions	179

Figure 4.18	Universal Enterprise Controller - Welcome dialog	193
Figure 4.19	Universal Enterprise Controller - Custom Setup dialog	194
Figure 4.20	Universal Enterprise Controller - Custom Setup / Selecting to not Install a Component	195
Figure 4.21	Universal Enterprise Controller - Installation Complete	196
Figure 4.22	Universal Enterprise Controller - Add or Remove Programs dialog	197
Figure 4.23	Failure to Access Install Files	198
Figure 4.24	Universal Enterprise Controller - Program Maintenance / Modify dialog	199
Figure 4.25	Universal Enterprise Controller - Custom Setup / Modifying an Installed Component	200
Figure 4.26	Universal Enterprise Controller - Program Maintenance / Repair dialog	201
Figure 4.27	Universal Enterprise Controller - Program Maintenance / Remove dialog	202
Figure 4.28	Universal Enterprise Controller - Location to Save Files dialog	204
Figure 4.29	Universal Enterprise Controller - Windows Installer Command Line Syntax	208
Figure 4.30	UEC Client Applications - Welcome dialog	221
Figure 4.31	UEC Client Applications - Customer Information dialog (for Administrators accounts)	222
Figure 4.32	UEC Client Applications - Windows Installer / Custom Setup dialog	224
Figure 4.33	UEC Client Applications - Custom Setup / Selecting to not Install a Component	225
Figure 4.34	UEC Client Applications - Installation Complete	226
Figure 4.35	UEC Client Applications - Add or Remove Programs dialog	227
Figure 4.36	Failure to Access Install Files	228
Figure 4.37	UEC Client Applications - Program Maintenance / Modify dialog	229
Figure 4.38	UEC Client Applications - Custom Setup / Modify dialog	230
Figure 4.39	UEC Client Applications - Program Maintenance / Repair dialog	231
Figure 4.40	UEC Client Applications - Program Maintenance / Remove dialog	232
Figure 4.41	UEC Client Applications - Location to Save Files dialog	234
Figure 4.42	UEC Client Applications - Windows Installer Command Line Syntax	239
Figure 4.43	Universal Products for SOA - Welcome dialog	252
Figure 4.44	Universal Products for SOA - Custom Setup Dialog	253
Figure 4.45	Universal Products for SOA - Installation Complete dialog	254
Figure 4.46	Universal Products for SOA - Add or Remove Programs Dialog	256
Figure 4.47	Universal Products for SOA - Program Maintenance / Repair dialog	257
Figure 4.48	Universal Products for SOA - Program Maintenance / Remove dialog	258
Figure 4.49	Universal Products for SOA - Location to Save Files dialog	260
Figure 4.50	Universal Products for SOA for Windows – Windows Installer Command Line Syntax	263
Figure 4.51	Universal Configuration Manager - Universal Application Container Server - License Information	267
Chapter 5 UNIX Installation		272
Figure 5.1	Single-Server Deployment - Logical View	317
Figure 5.2	Distributed-Server Deployment - Logical View	318

List of Tables

Chapter 1 Document Overview	36
Table 1.1 Universal Products Packages / Operating Systems	36
Chapter 2 z/OS Installation	39
Table 2.1 Universal Products for z/OS - Products Compatibility	41
Table 2.2 Universal Products for z/OS - SMP/E FMIDs for Universal Products Package	43
Table 2.3 Universal Products for z/OS - Installation Files	47
Table 2.4 Universal Products for z/OS - Cumulative PTF Maintenance Files	77
Table 2.5 Universal Broker for z/OS - Data Set Access Levels	84
Table 2.6 Universal Enterprise Controller for z/OS - Data Set Access Levels	84
Table 2.7 Universal Products for z/OS - Program Services Requiring APF Authorization	87
Table 2.8 Universal Products for z/OS - Controlled Programs in SUNVLOAD Library	88
Table 2.9 Universal Products for z/OS - Remote Access Profile Fields	118
Table 2.10 Universal Products for z/OS - Standard I/O Access Profile Fields	119
Table 2.11 Universal Control for z/OS - Remote Access Profile Fields	122
Table 2.12 Universal Event Monitor for z/OS - Remote Access Profile Fields	124
Table 2.13 Universal Products for z/OS - SMP/E Data Sets	129
Table 2.14 Universal Products for z/OS - Non-SMP/E Data Sets	130
Chapter 3 z/OS USS Installation	131
Table 3.1 Universal Products for z/OS USS - Product Compatibility	132
Table 3.2 Universal Products for z/OS USS - SMP/E FMIDs for Universal Products Package	134
Table 3.3 Universal Products for z/OS USS - SMP/E Data Sets	142
Table 3.4 Universal Products for z/OS USS - Directories	142
Chapter 4 Windows Installation	143
Table 4.1 Universal Products for Windows - Product Distribution Files	144

Table 4.2	Universal Products for Windows - Product Compatibility	149
Table 4.3	Universal Products for Windows - Windows Installer Command Line Parameters	170
Table 4.4	Universal Products for Windows - Applications Installed in Windows System Folder	180
Table 4.5	Universal Broker for Windows - File Inventory List	182
Table 4.6	Universal Command Manager for Windows - File Inventory List	182
Table 4.7	Universal Command Server for Windows - File Inventory List	183
Table 4.8	Universal Control Manager for Windows - File Inventory List	183
Table 4.9	Universal Control Server for Windows - File Inventory List	184
Table 4.10	Universal Data Mover Manager for Windows - File Inventory List	184
Table 4.11	Universal Data Mover Server for Windows - File Inventory List	185
Table 4.12	Universal Event Monitor Manager for Windows - File Inventory List	185
Table 4.13	Universal Event Monitor Server for Windows - File Inventory List	186
Table 4.14	Universal Configuration Manager for Windows - File Inventory List	186
Table 4.15	Universal Copy for Windows - File Inventory List	186
Table 4.16	Universal Encrypt for Windows - File Inventory List	187
Table 4.17	Universal Event Log Dump for Windows - File Inventory List	187
Table 4.18	Universal Message Translator for Windows - File Inventory List	187
Table 4.19	Universal Products Install Merge for Windows - File Inventory List	187
Table 4.20	Universal Query for Windows - File Inventory List	188
Table 4.21	Universal Spool Utilities for Windows - File Inventory List	188
Table 4.22	Universal Products for Windows System Files - File Inventory List	189
Table 4.23	Universal Enterprise Controller for Windows - Product Compatibility	190
Table 4.24	Universal Enterprise Controller - Windows Installer Command Line Interface Parameters	207
Table 4.25	Universal Enterprise Controller for Windows - Applications Installed in Windows System Folder	212
Table 4.26	Universal Enterprise Controller for Windows - File Inventory List	216
Table 4.27	Universal Products for Windows Install Merge Utility - File Inventory List	216
Table 4.28	Universal Enterprise Controller System Files - File Inventory List	217
Table 4.29	UEC Client Applications - Windows Installer Command Line Interface Parameters	238
Table 4.30	UEC Client Applications - I-Administrator File Inventory List	245
Table 4.31	UEC Client Applications - I-Activity Monitor File Inventory List	245
Table 4.32	UEC Client Applications - I-Management Console File Inventory List	246
Table 4.33	UEC Client Applications - System Files File Inventory List	246
Table 4.34	Universal Products for SOA for Windows - Windows Installer Command Line Parameters	262
Table 4.35	Universal Products for SOA for Windows System Files - File Inventory List	271
Chapter 5 UNIX Installation		272
Table 5.1	Universal Products 4.1.0 for UNIX - Product Compatibility	274
Table 5.2	Universal Products for AIX - Product Distribution File Contents	283
Table 5.3	Universal Products AIX - Package Filesets	284
Table 5.4	Universal Products for HP-UX - Distribution File Contents	286
Table 5.5	Universal Products for HP-UX - Package Subproducts	287

Table 5.6	Universal Products for Solaris (Intel-Based Systems) - Distribution File Contents	289
Table 5.7	Universal Products for Solaris (SPARC-Based Systems) - Product Distribution File	290
Table 5.8	Universal Products for Solaris - Package Names	291
Table 5.9	Universal Products for Linux (Intel-Based Systems) - Product Distribution File	293
Table 5.10	Universal Products for Linux (IBM S/390 and zSeries Systems) - Product Distribution File	294
Table 5.11	Universal Products for Linux (Itanium Systems) - Product Distribution File	295
Table 5.12	Universal Products for UNIX - Product Distribution File Contents	298
Table 5.13	Universal Products for UNIX - Environment Variables	299
Table 5.14	Universal Broker for UNIX - File Inventory List	307
Table 5.15	Universal Command Manager for UNIX - File Inventory List	308
Table 5.16	Universal Command Server for UNIX - File Inventory List	308
Table 5.17	Universal Connector for UNIX - File Inventory List	309
Table 5.18	Universal Control Manager for UNIX - File Inventory List	309
Table 5.19	Universal Control Server for UNIX - File Inventory List	309
Table 5.20	Universal Data Mover Manager for UNIX - File Inventory List	310
Table 5.21	Universal Data Mover Server for UNIX - File Inventory List	310
Table 5.22	Universal Event Monitor Manager for UNIX - File Inventory List	310
Table 5.23	Universal Event Monitor Server for UNIX - File Inventory List	311
Table 5.24	Universal Certificate for UNIX - File Inventory List	311
Table 5.25	Universal Query for UNIX - File Inventory List	311
Table 5.26	Universal Spool Utilities for UNIX - File Inventory List	312
Table 5.27	Product Directories and Files (/opt/universal) – Universal Products for SOA for UNIX	326
Table 5.28	Product Directories and Files (/var/opt/universal) – Universal Products for SOA for UNIX	327
Table 5.29	Product Directories and Files (/etc/universal) – Universal Products for SOA for UNIX	327
Chapter 6 OS/400 Installation		328
Table 6.1	Universal Products for OS/400 - Product Compatibility	329
Table 6.2	Universal Products for OS/400 - Installation Files (Full Release)	335
Table 6.3	Universal Products for OS/400 - Installation Files (Maintenance Release)	335
Table 6.4	Universal Products for OS/400 - Product Library (UNVPRD410) Object Inventory List	361
Table 6.5	Universal Products for OS/400 - Temporary Library (UNVTMP410) Object Inventory List	362
Table 6.6	Universal Products for OS/400 - Spool Library (UNVSPL410) Object Inventory List	362
Table 6.7	Universal Products for OS/400 - Command Reference Library (UNVCMDREF) Object Inventory List	362
Table 6.8	Universal Products for OS/400 - QSYS Library Object Inventory List	362

Chapter 7 HP NonStop Installation	363
Table 7.1 Universal Command for HP NonStop - Product Compatibility	364
Table 7.2 Universal Command for HP NonStop - Installation Files	368
Table 7.3 Universal Products for HP NonStop - TZ Environment Variable Values	370
Table 7.4 Universal Broker for HP NonStop - File Inventory List	374
Table 7.5 Universal Command Manager for HP NonStop - File Inventory List	374
Table 7.6 Universal Command Server for HP NonStop - File Inventory List	375
Table 7.7 Universal Control Manager for HP NonStop - File Inventory List	375
Table 7.8 Universal Control Server for HP NonStop - File Inventory List	375
Table 7.9 Universal Query for HP NonStop - File Inventory List	375

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.

Conventions

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

Typeface and Fonts

This Font identifies specific names of different types of information, such as file names or directories (for example, `\abc\123\help.txt`).

Tips from the Stoneman



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

Stoneman's Tip

Vendor References

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

The following names are used within this document:

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

Note: These names do not imply software support in any manner.

Document Organization

This document is organized into the following chapters:

- [Document Overview](#) (Chapter 1)
Overview of Universal Products to be installed and installation procedures
- [z/OS Installation](#) (Chapter 2)
Installation of Stonebranch products on z/OS operating systems.
- [z/OS USS Installation](#) (Chapter 3)
Installation of Stonebranch products on z/OS UNIX System Services (USS).
- [Windows Installation](#) (Chapter 4)
Installation of Stonebranch products on Windows operating systems.
- [UNIX Installation](#) (Chapter 5)
Installation of Stonebranch products on UNIX operating systems.
- [UNIX Installation](#) (Chapter 6)
Installation of Stonebranch products on OS/400 operating systems.
- [HP NonStop Installation](#) (Chapter 7)
Installation of Stonebranch products on HP NonStop operating systems.
- [Licenses and Copyrights](#) (Appendix A)
License and copyright information for third-party libraries used within our products.
- [Customer Support](#) (Appendix B)
Customer support contact information for Universal Products installation.

Chapter 1

Document Overview

1.1 Overview

This document provides information and instructions for the installation of Universal Products packages on one or more operating systems.

[Table 1.1](#), below, identifies the different operating systems on which Universal Products can be installed, and the packages that can be installed on each system.

Operating System	Universal Products	Universal Enterprise Controller	Universal Enterprise Controller Client Application	Universal Products for SOA
z/OS ¹	✓			
z/OS USS	✓			
Windows	✓	✓	✓	✓
UNIX – AIX	✓			✓
UNIX – HP-UX	✓			
UNIX – Linux	✓			✓
UNIX – Solaris	✓			
UNIX – (generic)	✓			
OS/400	✓			
HP NonStop ²	✓			

1 Universal Products for z/OS contains Universal Enterprise Controller.
2 Universal Products 2.1.1 is installed on HP NonStop.

Table 1.1 Universal Products Packages / Operating Systems

1.2 Installation Information

This information includes, but is not limited to, the following areas:

- Installation package components
- Product compatibility
- Installation requirements
- Installation upgrades
- System upgrades
- Distribution files
- Installation procedures
- Product customization
 - Configuration
 - Licensing
- File inventory lists

1.2.1 Product Distribution

To install a Universal Products 4.1.0 package, you first must download the corresponding product distribution file for your specific operating system version and hardware platform.

All Universal Products product distribution files are available for download in the Support / Current Products area of the Stonebranch, Inc. website, <http://www.stonebranch.com>.

A customer user ID and password – provided by Stonebranch, Inc. – are required to access this area.

The chapters in this book identify the specific product distribution files to be downloaded for the different Universal Products packages.

1.2.2 Product Licensing

A Universal Products package installation may include products that must be licensed before they can be used.

The chapters in this book identify the products that must be licensed, and how the licensing is made, for Universal Products 4.1.0 on each operating system.

Licenses for Universal Products are provided by your Stonebranch, Inc. account representative.

1.2.3 Additional Documentation

Additionally to this installation guide, Stonebranch, Inc. provides a Universal Products 4.1.0 Installation Requirements document.

The Installation Requirements document identifies network requirements for all Universal Products installation and, for each operating system, the following information:

- System requirements
- Installation summary
- Customization

Chapter 2

z/OS Installation

2.1 Overview

This chapter describes the installation of Stonebranch, Inc. Universal Products 4.1.0 on the z/OS operating system. Unless otherwise specified, all references to Universal Products for z/OS in this chapter refer to version 4.1.0.

All Universal Products for z/OS are provided in an easily installed, one-time installation package.

(For information on installation of Universal Products on z/OS UNIX System Services (USS), see [Chapter 3 z/OS USS Installation](#).)

Note: Starting with the 3.2.0 release of Universal Products, a Universal Broker must run on all systems on which a Universal Product component is running, including manager components. The Broker maintains product configuration data for all components that have a configuration file.

2.1.1 Installation Procedures

Installation of Universal Products for z/OS is comprised of the following procedures:

1. Download the Universal Products for z/OS package distribution file.
2. Transfer the installation files from Windows to z/OS.
 - a. Extract the installation files from the distribution file.
 - b. Allocate data sets.
 - c. Transfer the installation files to the data sets on z/OS.
 - d. Unpack the data sets.
3. Install the files.

The installation itself consists of running a number of batch jobs. The output of these batch jobs should be kept until a correct installation has been verified.

2.2 Universal Products for z/OS Package

The Universal Products 4.1.0 for z/OS package contains the following Universal Products components:

- Universal Broker 4.1.0
- Universal Command Manager and Server 4.1.0
- Universal Control Manager and Server 4.1.0
- Universal Data Mover Manager and Server 4.1.0
- Universal Encrypt 4.1.0
- Universal Enterprise Controller 4.1.0
- Universal Event Monitor Manager 4.1.0
- Universal Message to Exit Code Translator 4.1.0
- Universal Query 4.1.0
- Universal Connector 4.1.0
- Universal Certificate 4.1.0

2.2.1 Product Compatibility

[Table 2.1](#), below, identifies the compatibility of Universal Products 4.1.0 for z/OS with previous product versions.

Product	Compatibility
Universal Broker 4.1.0	Universal Product component releases 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Command 4.1.0	Universal Command 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Control 4.1.0	Universal Control 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Data Mover 4.1.0	Universal Data Mover 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Encrypt 4.1.0	Universal Encrypt 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Query 4.1.0	Universal Broker 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Enterprise Controller 4.1.0	Not compatible with previous versions of Universal Broker Monitor or Universal Enterprise Controller Administrator.
Universal Event Monitor 4.1.0	Universal Event Monitor 3.2.0, 3.1.1, and 3.1.0.

Table 2.1 Universal Products for z/OS - Products Compatibility

The product references pertain to all support platforms for that version.

2.3 Universal Products for z/OS – Installation Requirements

Universal Products for z/OS require the following software releases:

- z/OS 1.4 or above.
- IBM Language Environment for z/OS 1.4 or above.
- IBM Communication Server for z/OS 1.4 or above.
- SMP/E 3.2 or above.
- About 1200 cylinders of DASD.

All Universal Products programs use z/OS UNIX System Services. As such, z/OS UNIX requires the user profile with which a program executes to have a properly defined OMVS segment. The OMVS segment should define a unique UID value. The HOME value must specify an existing home directory to which the user ID has read and write access.

Additionally, the group(s) that the user ID is associated with must have an OMVS segment that defines a unique GID value for the group. Refer to IBM's UNIX System Services Planning manual for additional details on defining z/OS UNIX users.

2.3.1 Data Set Space Requirements

As part of the Universal Products for z/OS package installation, a number of SMP/E and non-SMP/E data sets are allocated and cataloged.

The space requirements for these data sets are listed in [Section 2.16 Universal Products for z/OS – Data Set Inventory](#).

2.3.2 SMP/E

Universal Products are installed using SMP/E. The products are installed in the Universal Products SMP/E CSI. This CSI should not be shared with any other vendor products. The Universal Products CSI should be used exclusively for Universal Products.

Before making any changes to the SMP/E environment, back up the environment per your local procedures. IBM recommends backing up the entire SMP/E pack before any SMP/E installation begins. Two sample JCL members are provide in members **UNVBKUP** and **UNVREST**.

As of version 2.2.0, all Universal Products share common SMP/E target and distribution libraries:

- Product ID is **UNV**.
- Last qualifier of target libraries start with **SUNV**.
- Last qualifier of distribution libraries start with **AUNV**.

[Table 2.2](#), below, identifies the SMP/E FMIDs included with the Universal Products for z/OS package.

Note: This package does not include the z/OS USS FMIDs (see [Table 3.2 Universal Products for z/OS USS - SMP/E FMIDs for Universal Products Package](#)).

Product	FMID	SMP/E Requisites
Universal Common 4.1.0	TUNV410	Supersedes and deletes FMID TUNV220, TUNV310, TUNV311, and TUNV320.
Universal Broker 4.1.0	TUBR410	TUNV410 is a prerequisite. Supersedes and deletes FMID TUBR310, TUBR311, and TUBR320.
Universal Command 4.1.0	TUCM410	TUBR410 is a prerequisite. Supersedes and deletes FMID TUCM110, TUCM120, TUCM210, TUCM220, TUCM310, TUCM311, and TUCM320.
Universal Enterprise Controller 4.1.0	TUEC410	TUNV410 is a prerequisite. Supersedes and deletes FMID TUEC110, TUEC310, and TUEC320.
Universal Data Mover 4.1.0	TUDM410	TUBR410 is a prerequisite. Supersedes and deletes FMID TUDM110, TUDM310, TUDM311, and TUDM320.
SAP RFC DLL	TUSD410	TUNV410 is a prerequisite. Supersedes and deletes FMID TUSD310, TUSD311, and TUSD320.
Universal Connector	TUSP410	TUSD410 is a prerequisite. Supersedes and deletes FMID TUSP120, TUSP310, TUSP311, and TUSP320.
Universal Utilities	TUTL410	TUNV410 is a prerequisite. Supersedes and deletes FMID TUEN110, TUEN120, TUEN210, TUEN220, TUEN310, TUEN311, and TUTL320.
Universal Event Monitor	TUEM410	TUBR410 is a prerequisite. Supersedes and deletes FMID TUEM310, TUEM311, and TUEM320.
IBM XML Toolkit	TUXD410	TUNV410 is a prerequisite. Supersedes and deletes TUXD320.

Table 2.2 Universal Products for z/OS - SMP/E FMIDs for Universal Products Package

2.4 Universal Products for z/OS – Installation Upgrades

This section describes changes in the product installation that have occurred with new versions of the product. If a particular version is absent from the list, no change occurred.

Universal Command 1.2.0

Starting with Universal Command 1.2.0, the SMP/E target and distribution library names have changed. The product name qualifier has been changed to the three-character product identifier, and the version, release, modification qualifier (**V1R1M0**) has been removed. The changes comply with the IBM naming conventions for SMP/E libraries. Also, the absence of the version qualifier will make future upgrades much simpler and trouble-free.

Universal Command 2.1.0

Start with Universal Command 2.1.0, the SMP/E target and distribution libraries **SUCMNLS**, **AUCMNLS**, **SUCMCONF**, and **AUCMCONF** have changed their record format from variable block to fixed block.

Universal Products 2.2.0

Products are packaged in the new Universal Products package. Starting with Universal Products 2.2.0, SMP/E target and distribution libraries are now shared between all products in order to make maintenance and administration easier.

Product target and distribution data set names have changed. All products in the Universal Products package use **UNV** as the three character product identifier. As such, the last qualifier of the target libraries start with **SUNV** and the last qualifier of the distribution libraries start with **AUNV**.

Universal Products 3.1.1

Package size has increased significantly as the Universal Broker and Server components are now included as FMIDs. Product SMP/E target library **SUNVLOAD** and SMP/E distribution library **AUNVLOAD** space allocation should be increased to meet the new space requirements. Additionally, SMP/E **SMPLTS** and **SMPPTS** data sets and **SMPTLIB** DDDEF space requirements have increased.

Universal Products 3.2.0

Product configuration files have been moved from SMP/E target libraries **SUNVCONF** and **SUNVCOMP** to non-SMP/E product libraries **UNVCONF** and **UNVCOMP**. The change allows for remote configuration capability of z/OS configuration members. Configuration members are delivered in the SMP/E **SUNVSAMP** library and copied to the appropriate library in an installation job.

All SMP/E target and distribution libraries have been changed from PDS to PDSEs. This includes the load libraries **SUNVLOAD** and **AUNVLOAD**. The installation upgrade jobs include steps to convert existing libraries to PDSE format.

2.5 Universal Products for z/OS – Product Distribution File

The Universal Products for z/OS product distribution file contains all of the files required for the installation of the Universal Products for z/OS package.

To obtain the Universal Products for z/OS package, you must download the corresponding product distribution file from the Support / Current Products area of the Stonebranch, Inc. website, <http://www.stonebranch.com>.

Note: A customer user ID and password — provided by Stonebranch, Inc. — are required to access this area.

After a distribution file has been downloaded, the installation files contained in that file must be extracted before the product can be installed (see Section [2.6 Universal Products for z/OS – Transferring Installation Files to z/OS](#)).

2.5.1 Distribution File Format

The Universal Products for z/OS product distribution file is an Express/OS Windows self-extractable executable file. (Express/OS is a packaging and installation GUI application provided by Widearea Data Systems, Inc. for the Microsoft Windows operating system.)

The name of Universal Products for z/OS distribution file has the following format:

unv-Version.Release.Modification Level.Maintenance Level-operating system.exe

For example: **unv-4.1.0.1-zOS.exe**

In this format:

- **Version** is the current version of Universal Products.
- **Release** is the current release of Universal Products.
- **Modification Level** is the current Universal Products feature set.
- **Maintenance Level** is the Universal Products build level.
- **operating system** is the name of the operating system.

2.5.2 Distribution File Contents

Table 2.3, below, lists the installation files (in XMT file format) included in the Universal Products for z/OS distribution file.

Note: These files include the files for z/OS USS.

File Name	Description
INSTALL .XMT	Universal Products package installation JCL.
README .TXT	Documentation on the package and transfer methods.
SMPMCS .XMT	Universal Products SMP/E MCS statements.
UBR410F1 .XMT	Universal Broker SMP/E FMID TUBR410 relative file 1.
UBR410F2 .XMT	Universal Broker SMP/E FMID TUBR410 relative file 2.
UBR410F3 .XMT	Universal Broker SMP/E FMID TUBR410 relative file 3.
UCM410F1 .XMT	Universal Command SMP/E FMID TUCM410 relative file 1.
UCM410F2 .XMT	Universal Command SMP/E FMID TUCM410 relative file 2.
UCM410F3 .XMT	Universal Command SMP/E FMID TUCM410 relative file 3.
UDM410F1 .XMT	Universal Data Mover SMP/E FMID TUDM410 relative file 1.
UDM410F2 .XMT	Universal Data Mover SMP/E FMID TUDM410 relative file 2.
UDM410F3 .XMT	Universal Data Mover SMP/E FMID TUDM410 relative file 3.
UEC410F1 .XMT	Universal Enterprise Controller SMP/E FMID TUEC410 relative file 1.
UEC410F2 .XMT	Universal Enterprise Controller SMP/E FMID TUEC410 relative file 2.
UEM410F1 .XMT	Universal Event Monitor SMP/E FMID TUEM410 relative file 1.
UEM410F2 .XMT	Universal Event Monitor SMP/E FMID TUEM410 relative file 2.
UEM410F3 .XMT	Universal Event Monitor SMP/E FMID TUEM410 relative file 3.
UNV410F1 .XMT	Universal Common SMP/E FMID TUNV410 relative file 1.
USD410F1 .XMT	SAP RFC DLL FMID TUSD410 relative file 1.
USP410F1 .XMT	Universal Connector FMID TUSP410 relative file 1.
USP410F2 .XMT	Universal Connector FMID TUSP410 relative file 2.
USP410F3 .XMT	Universal Connector SMP/E FMID TUSP410 relative file 3.
UTL410F1 .XMT	Universal Utilities SMP/E FMID TUTL410 relative file 1.
UTL410F2 .XMT	Universal Utilities SMP/E FMID TUTL410 relative file 2.
UXD410F1 .XMT	IBM XML Toolkit FMID TUXD410 relative file 1.
z/OS USS Files	
UBR410U1 .XMT	USS Universal Broker FMID UUBR410 relative file 1.
UBR410U2 .XMT	USS Universal Broker FMID UUBR410 relative file 2.
UCM410U1 .XMT	USS Universal Broker FMID UUCM410 relative file 1.
UCM410U2 .XMT	USS Universal Broker FMID UUCM410 relative file 2.
UDM410U1 .XMT	USS Universal Broker FMID UUDM410 relative file 1.
UDM410U2 .XMT	USS Universal Broker FMID UUDM410 relative file 2.
UEM410U1 .XMT	USS Universal Broker FMID UUEM410 relative file 1.

File Name	Description
UEM410U2.XMT	USS Universal Broker FMID UUEM410 relative file 2.
UNV410U1.XMT	USS Universal Broker FMID UUNV410 relative file 1.
UTL410U1.XMT	USS Universal Broker FMID UUTL410 relative file 1.

Table 2.3 Universal Products for z/OS - Installation Files

2.6 Universal Products for z/OS – Transferring Installation Files to z/OS

The installation files must be transferred to the z/OS system on which the product is being installed. The XMIT (TSO TRANSMIT format) data sets must be unpacked with the TSO RECEIVE command.

There are two methods of transferring the installation files:

Express/OS Method

Express/OS transfers the installation files to the z/OS system using FTP in passive mode. It also will unpack the XMIT data sets on z/OS.

Express/OS is the simplest — and quickest — transfer method.

(See Section [2.6.1 Express/OS Method](#).)

Manual Method

The manual method requires each installation file to be extracted from the distribution file, then transferred manually from the Windows workstation to the z/OS system.

(See Section [2.6.2 Manual Method](#).)

2.6.1 Express/OS Method

This section describes how to transfer the installation files to z/OS using the Express/OS method.

Express/OS performs the following tasks:

- Extracts the installation files from the distribution file.
- Allocates data sets on z/OS for the installation files.
- Transfers the installation files to the z/OS system.
- Submits a z/OS job to unpack the data sets.

To execute Express/OS, double-click the `unv-4.1.0.<level>-zOS.exe` file in Windows Explorer. Express/OS then prompts you for information as it performs the following steps:

1. Express/OS extracts the installation files to a temporary directory. It provides a default directory, although you can provide a directory of your choice.
2. Express/OS prompts you for the z/OS system TCP/IP host name and the FTP port number (default is 21). The host name may be either a domain name or a numeric TCP/IP address.

FTP is not required on the system from which Express/OS is executing. Express/OS implements the FTP protocol itself. However, you do need the ability to open a passive FTP connection to the z/OS system from the computer on which Express/OS is executing.

3. Express/OS prompts you for a z/OS user ID and password. This information is used by Express/OS to log on to FTP in order to transfer the files to the z/OS system.
4. Express/OS prompts you for the data set high-level qualifier with which to allocate the transferred XMT files.
 - For non-SMS data sets, provide the unit and volume serial number on which to allocate the data sets.
 - For SMS data sets, provide the management class and storage class with which to allocate the data sets.
5. Express/OS prompts you for a data set name in which to save the Express/OS JCL required to unpack the data sets. If the data set does not exist, Express/OS creates it.
6. Express/OS provides a default JCL JOB statement used to create the JCL. You can edit the JOB statement at this time to meet local requirements.
7. Express/OS transfers the installation files to the allocated data sets on z/OS using FTP.
8. Express/OS prompts you on whether or not you want to submit the Express/OS job to unpack the data sets. If you answer no, you must submit the job manually. The job is located in the data set provided in Step 5, above.

When the Express/OS application completes, the Universal Products data sets are unpacked on the z/OS system and ready for installation (see Section [2.7 Universal Products for z/OS – SMP/E Installation of Universal Products](#)).

2.6.2 Manual Method

This section describes the manual method of transferring the installation files to z/OS.

The manual method may be required in cases where you cannot use FTP to transfer files from the Windows workstation to the z/OS system. Instead of FTP, another file transfer protocol must be used.

To transfer the installation files to z/OS manually, you must:

1. Extract the installation files from the distribution file (using Express/OS).
2. Allocate data sets on z/OS for the installation files.
3. Transfer the installation files to the data sets.
4. Unpack the data sets.

Extract the Installation Files

To extract the installation files from the distribution file, perform the following steps.

1. Double-click the name of the distribution file, `unv-4.1.0.<1eve1>-zOS.exe`, in Windows Explorer.
2. On the introductory Express/OS screen, click the **Next>** button.
3. The next screen ([Figure 2.1](#)) prompts for a directory in which to extract the installation files.

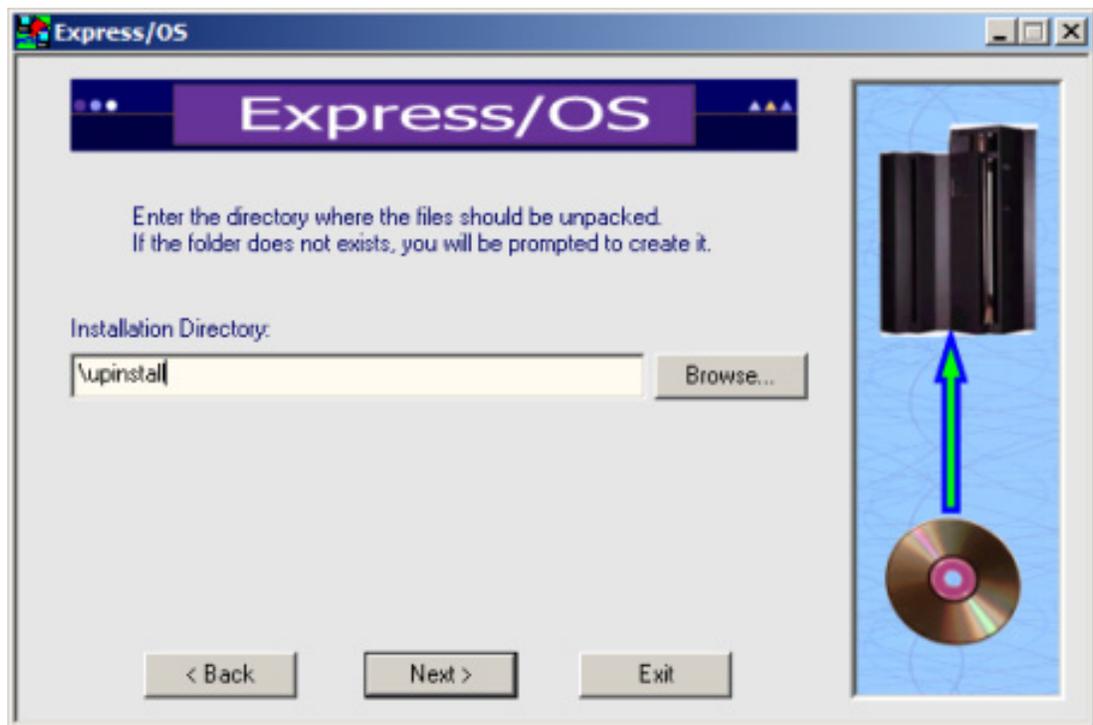


Figure 2.1 Universal Products for z/OS - Express/OS / Installation Directory dialog

When the dialog first displays, the **Installation Directory** field may contain a temporary directory deep within your user profile directory. You can use that directory or **Browse . . .** for another directory.

(Consider replacing that directory name with an easy-to-locate directory, such as **\upinstall**.)

4. Click the **Next >** button to extract the files to the specified directory. If the directory that you selected in Step 3 does not exist, Express/OS prompts you to create it.
5. On the next screen, exit Express/OS by clicking the **Exit** button and replying to the confirmation dialog.

Allocate z/OS Data Sets

The installation files (with a .XMT suffix) have been created using the TSO TRANSMIT command. They must be transferred (in binary mode) to z/OS data sets of the following format:

- Sequential (DSORG=PS)
- Fixed blocked record format (RECFM=FB)
- Record length 80 (LRECL=80)
- Block size 3120 (BLKSIZE=3120)

The following sample JCL ([Figure 2.2](#), below) can be used to allocate and catalog the XMIT data sets on z/OS.

It must be transferred to the z/OS system on which the Universal Products package is being installed. The file must be transferred in text mode to a fixed record format data set with a logical record length of 80.

The JCL includes a MODIFICATIONS section at the beginning that describes what JCL modifications are required prior to submitting the job. Read and complete each of the listed modifications.

Submit the job on z/OS to allocate the XMIT data sets. Return code 0 is expected.

```
//UNVALLOC JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//          COND=(0,NE)
//*****
//* (C) COPYRIGHT 2000-2009 STONEBRANCH, INC. ALL RIGHTS RESERVED.
//*
//* STONEBRANCH, INC.
//* UNIVERSAL PRODUCTS
//*
//* UNVALLOC
//*
//* DESCRIPTION
//* -----
//* ALLOCATE TSO TRANSMIT (XMIT) DATA SETS FOR THE DISTRIBUTION
//* DATA SETS.
//*
```

```

/* MODIFICATIONS
/* -----
/* 1. MODIFY THE JOB STATEMENT TO MEET LOCAL REQUIREMENTS.
/*
/* 2. CHANGE ALL '#HLQ' TO THE HIGH-LEVEL QUALIFIER OF THE
/*    UNIVERSAL PRODUCTS DISTRIBUTION XMIT DATA SETS.
/*
/* 3. CHANGE ALL '#VOLSER' TO THE VOLUME SERIAL NAME ON WHICH TO
/*    ALLOCATE THE DATA SETS.
/******
/*
//STEP1    EXEC PGM=IEFBR14
//SMPMCS   DD  DSN=#HLQ.UNV.V4R1M0.SMPMCS.XMIT,
//          DISP=(,CATLG),
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(10,10))
//INSTALL  DD  DSN=#HLQ.UNV.V4R1M0.INSTALL.XMIT,
//          DISP=(,CATLG),
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(1,1))
//UNVF1    DD  DSN=#HLQ.UNV.V4R1M0.TUNV410.F1.XMIT,
//          DISP=(,CATLG),
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(1,1))
//UTLF1    DD  DSN=#HLQ.UNV.V4R1M0.TUTL410.F1.XMIT,
//          DISP=(,CATLG),
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(1,1))
//UTLF2    DD  DSN=#HLQ.UNV.V4R1M0.TUTL410.F2.XMIT,
//          DISP=(,CATLG),
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(70,1))
//UBRF1    DD  DSN=#HLQ.UNV.V4R1M0.TUBR410.F1.XMIT,
//          DISP=(,CATLG),
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(1,1))

```

```
//UBRF2 DD DSN=#HLQ.UNV.V4R1M0.TUBR410.F2.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(1,1))
//UBRF3 DD DSN=#HLQ.UNV.V4R1M0.TUBR410.F3.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(120,1))
//UCMF1 DD DSN=#HLQ.UNV.V4R1M0.TUCM410.F1.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(1,1))
//UCMF2 DD DSN=#HLQ.UNV.V4R1M0.TUCM410.F2.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(1,1))
//UCMF3 DD DSN=#HLQ.UNV.V4R1M0.TUCM410.F3.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(45,1))
//UDMF1 DD DSN=#HLQ.UNV.V4R1M0.TUDM410.F1.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(1,1))
//UDMF2 DD DSN=#HLQ.UNV.V4R1M0.TUDM410.F2.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(1,1))
//UDMF3 DD DSN=#HLQ.UNV.V4R1M0.TUDM410.F3.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(55,1))
```

```

//UEMF1 DD DSN=#HLQ.UNV.V4R1M0.TUEM410.F1.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(1,1))
//UEMF2 DD DSN=#HLQ.UNV.V4R1M0.TUEM410.F2.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(1,1))
//UEMF3 DD DSN=#HLQ.UNV.V4R1M0.TUEM410.F3.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(15,1))
//UXDF1 DD DSN=#HLQ.UNV.V4R1M0.TUXD410.F1.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(50,1))
//UECF1 DD DSN=#HLQ.UNV.V4R1M0.TUEC410.F1.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(1,1))
//UECF2 DD DSN=#HLQ.UNV.V4R1M0.TUEC410.F2.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(70,1))
//USDF1 DD DSN=#HLQ.UNV.V4R1M0.TUSD410.F1.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(20,1))
//USPF1 DD DSN=#HLQ.UNV.V4R1M0.TUSP410.F1.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(1,1))

```

```
//USPF2 DD DSN=#HLQ.UNV.V4R1M0.TUSP410.F2.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(1,1))
//USPF3 DD DSN=#HLQ.UNV.V4R1M0.TUSP410.F3.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(15,1))
//UBRU1 DD DSN=#HLQ.UNV.V4R1M0.UUBR410.F1.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(1,1))
//UBRU2 DD DSN=#HLQ.UNV.V4R1M0.UUBR410.F2.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(50,1))
//UCMU1 DD DSN=#HLQ.UNV.V4R1M0.UUCM410.F1.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(1,1))
//UCMU2 DD DSN=#HLQ.UNV.V4R1M0.UUCM410.F2.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(30,1))
//UDMU1 DD DSN=#HLQ.UNV.V4R1M0.UUDM410.F1.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(1,1))
//UDMU2 DD DSN=#HLQ.UNV.V4R1M0.UUDM410.F2.XMIT,
//      DISP=(,CATLG),
//      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//      UNIT=SYSALLDA,VOL=SER=#VOLSER,
//      SPACE=(CYL,(30,1))
```

```
//UEMU1    DD  DSN=#HLQ.UNV.V4R1M0.UUEM410.F1.XMIT,
//          DISP=(,CATLG),
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(1,1))
//UEMU2    DD  DSN=#HLQ.UNV.V4R1M0.UUEM410.F2.XMIT,
//          DISP=(,CATLG),
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(30,1))
//UTLU1    DD  DSN=#HLQ.UNV.V4R1M0.UUTL410.F1.XMIT,
//          DISP=(,CATLG),
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(60,1))
//UNVU1    DD  DSN=#HLQ.UNV.V4R1M0.UUNV410.F1.XMIT,
//          DISP=(,CATLG),
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(1,1))
```

Figure 2.2 Universal Products for z/OS - Sample JCL for Allocating and Cataloging XMIT Data Sets on z/OS

Transfer the Installation Files

Transfer the installation files to the allocated data sets (see [Allocate z/OS Data Sets](#)).

Note: A binary transfer must be performed; otherwise, the data sets will not unpack correctly.

Unpack the Data Sets

After the installation files have been transferred to their allocated data sets on z/OS, they must be unpacked using the TSO RECEIVE command. The TSO RECEIVE command can be executed interactively or in batch.

The following sample JCL (Figure 2.3, below) can be used to unpack the data sets. It must be transferred to the z/OS system on which the Universal Products package is being installed. The file must be transferred in text mode to a fixed record format data set with a logical record length of 80.

The JCL includes a MODIFICATIONS section at the beginning of the file that describes what JCL modifications are required prior to submitting the job. Read and complete each of the listed modifications.

Submit the job on z/OS to unpack the data sets. Return code 0 is expected.

After the data sets are successfully unpacked, they are no longer required and can be deleted. The files then are ready to be installed.

```
//UNVRECV JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID,  
//          COND=(0,NE)  
//*****  
//* (C) COPYRIGHT 2000-2009 STONEBRANCH, INC. ALL RIGHTS RESERVED.  
//*  
//* STONEBRANCH, INC.  
//* UNIVERSAL PRODUCTS  
//*  
//* UNVRECV  
//*  
//* DESCRIPTION  
//* -----  
//* PERFORM A TSO RECEIVE ON THE TRANSMITTED DISTRIBUTION DATA SETS.  
//* ALL STEPS MUST END WITH RC 0.  
/*
```

```

/* MODIFICATIONS
/* -----
/* 1  MODIFY THE JOB STATEMENT TO MEET LOCAL REQUIREMENTS.
/*
/* 2  CHANGE ALL '#HLQ' TO THE HIGH-LEVEL QUALIFIER OF THE
/*     UNIVERSAL PRODUCTS DATA SETS.
/*
/* 3  CHANGE ALL '#VOLSER' TO THE VOLUME SERIAL NAME ON WHICH TO
/*     ALLOCATE THE DISTRIBUTION DATA SETS.
/******
/*
//TSO      PROC
//S1       EXEC PGM=IKJEFT01
//SYSLBC   DD  DISP=SHR,DSN=SYS1.BROADCAST
//SYSPROC  DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTEM   DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//         PEND
/*
//SMPMCS   EXEC TSO
//S1.SYSTSIN DD *
    RECEIVE INDA('#HLQ.UNV.V4R1M0.SMPMCS.XMIT')
    DA('#HLQ.UNV.V4R1M0.SMPMCS') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(100 50) RELEASE
//INSTALL  EXEC TSO
//S1.SYSTSIN DD *
    RECEIVE INDA('#HLQ.UNV.V4R1M0.INSTALL.XMIT')
    DA('#HLQ.UNV.V4R1M0.INSTALL') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(15 10)
//UNV410F1 EXEC TSO
//S1.SYSTSIN DD *
    RECEIVE INDA('#HLQ.UNV.V4R1M0.TUNV410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUNV410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(15 1) RELEASE

```

```
//UTL410F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUTL410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUTL410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(15 1) RELEASE
//UTL410F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUTL410.F2.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUTL410.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(910 10) RELEASE
//UBR410F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUBR410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUBR410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(15 1) RELEASE
//UBR410F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUBR410.F2.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUBR410.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(15 1) RELEASE
//UBR410F3 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUBR410.F3.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUBR410.F3') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(1600 15) RELEASE
//UCM410F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUCM410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUCM410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UCM410F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUCM410.F2.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUCM410.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
```

```
//UCM410F3 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUCM410.F3.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUCM410.F3') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(600 15) RELEASE
//UDM410F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUDM410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUDM410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UDM410F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUDM410.F2.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUDM410.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UDM410F3 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUDM410.F3.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUDM410.F3') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(750 15) RELEASE
//UEM410F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUEM410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUEM410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UEM410F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUEM410.F2.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUEM410.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UEM410F3 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUEM410.F3.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUEM410.F3') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(200 15) RELEASE
```

```
//UXD410F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUXD410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUXD410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(700 10) RELEASE
//UEC410F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUEC410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUEC410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UEC410F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUEC410.F2.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUEC410.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(1000 15) RELEASE
//USD410F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUSD410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUSD410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(300 15) RELEASE
//USP410F1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUSP410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUSP410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//USP410F2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUSP410.F2.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUSP410.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//USP410F3 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.TUSP410.F3.XMIT')
    DA('#HLQ.UNV.V4R1M0.TUSP410.F3') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(200 15) RELEASE
```

```
//UBR410U1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.UUBR410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.UUBR410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UBR410U2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.UUBR410.F2.XMIT')
    DA('#HLQ.UNV.V4R1M0.UUBR410.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(500 50) RELEASE
//UCM410U1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.UUCM410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.UUCM410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UCM410U2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.UUCM410.F2.XMIT')
    DA('#HLQ.UNV.V4R1M0.UUCM410.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(500 50) RELEASE
//UDM410U1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.UUDM410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.UUDM410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UDM410U2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.UUDM410.F2.XMIT')
    DA('#HLQ.UNV.V4R1M0.UUDM410.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(500 50) RELEASE
//UEM410U1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.UUEM410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.UUEM410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
```

```
//UEM410U2 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.UUEM410.F2.XMIT')
    DA('#HLQ.UNV.V4R1M0.UUEM410.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(500 50) RELEASE
//UTL410U1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.UUTL410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.UUTL410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(650 15) RELEASE
//UNV410U1 EXEC TSO
//S1.SYSTSIN DD *
  RECEIVE INDA('#HLQ.UNV.V4R1M0.UUNV410.F1.XMIT')
    DA('#HLQ.UNV.V4R1M0.UUNV410.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
```

Figure 2.3 Universal Products for z/OS - Sample JCL for Unpacking XMIT Data Sets on z/OS

2.7 Universal Products for z/OS – SMP/E Installation of Universal Products

This section describe how to perform SMP/E installation of the Universal Products for z/OS package in a step-by-step process.

Four different installation processes are provided. Which installation process used depends on the installation environment.

1. Installing any Universal Product from Stonebranch, Inc. for the first time, or installing a Universal Products package in a new SMP/E CSI.

See Section [2.7.1 New Install, New CSI](#) for installation instructions.

2. Upgrading a Universal Products 3.2.0 package install. In this case, the Universal Products 4.1.0 package is installed into an SMP/E CSI that contains a Universal Products 3.2.0 package.

See Section [2.7.2 Universal Products 3.2.0 Upgrade, Existing CSI](#) for installation instructions.

3. Upgrading a Universal Products 3.1.1 package install. In this case, the Universal Products 4.1.0 package is installed into an SMP/E CSI that contains Universal Products 3.1.1 package.

See Section [2.7.3 Universal Products 3.1.1 Upgrade, Existing CSI](#) for installation instructions.

4. Upgrading a Universal Products 3.1.0 package install. In this case, the Universal Products 4.1.0 package is installed into an SMP/E CSI that contains Universal Products 3.1.0 package.

See Section [2.7.4 Universal Products 3.1.0 Upgrade, Existing CSI](#) for installation instructions.

If you are upgrading from Universal Products 2.2.0 or Universal Command 2.1.0, 1.2.0, or 1.1.0, you first must upgrade to 3.1.0 or later and then upgrade to Universal Products 4.1.0.

2.7.1 New Install, New CSI

The New Install, New CSI installation process describes how to install the Universal Products package in a newly allocated SMP/E CSI.

Use this installation process for either of these environments:

- Installing a Universal Product from Stonebranch, Inc. for the first time.
- Installing a Universal Products package in a different SMP/E CSI than other Universal Products from Stonebranch, Inc.

The installation JCL referenced by the following installation steps is created by the **#SETUP** member in the Universal Products **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

1. Edit and submit the JCL in member **#SETUP**.

The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library.

If you want to change the customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced.

All steps must end with a return code 0.

2. Submit the JCL in member **UNVIN01**. The JCL allocates the SMP/E CSI data sets.
All steps must end with a return code 0.
3. Submit the JCL in member **UNVIN02**. The JCL initializes the SMP/E CSI.
All steps must end with a return code 0.
4. Submit the JCL in member **UNVIN03**. The JCL allocates Universal Products target and distribution data sets and adds SMP/E **DDDEF** definitions to the Universal Products CSI zones.
All steps must end with a return code 0.
5. Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E **RECEIVE** of the product FMIDs and available PTFs from the distribution data sets.
All steps must end with a return code 0.
6. Submit the JCL in member **UNVIN05**. The JCL performs a SMP/E **APPLY** of the product FMIDs and any received PTFs.

Step **APYFMID** must end with a condition code of 0. Step **APYPTFS** is considered successful under any of the following conditions:

- Step ends with condition code 0.
- Step ends with condition code 4, and message GIM42001W is written in ddname **SMPDOUT**.
- Step ends with condition code 12, and message GIM24801S is written in ddname **SMPDOUT**.

7. Submit the JCL in member **UNVIN06**. The JCL member performs a SMP/E ACCEPT of the product FMIDs and any applied PTFs.

Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions:

- Step ends with condition code 0 or 4.
- Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.

8. Submit the JCL in member **UNVIN07**. The JCL member allocates product non-SMP/E data sets and databases.

All steps must end with a return code 0.

9. The product databases are allocated as z/OS UNIX HFS file systems. If you prefer to use zFS file systems, zFS data sets must be allocated. If you prefer to use the HFS data sets, then skip this step.

Submit the JCL in member **UNVINZF**. The JCL renames the HFS data sets, defines their zFS counterparts, and formats the zFS data sets.

All steps must end with return code 0.

Additional product configuration is required to utilize zFS data sets. See Section [2.9.5 Universal Products Database Configuration](#) for details.

10. Submit the JCL in member **UNVIN08**. The JCL copies sample configuration members to the configuration libraries.

All steps must end with a return code 0.

11. Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL.

The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library.

All steps must end with a return code 0.

Note: The following step is required only if you have to convert Universal Enterprise Controller (UEC) 3.1.1 databases to 4.1.0 format.

12. Edit and submit the JCL in member **UECDB311**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL.

UEC 4.1.0 databases are stored in an HFS data set. The UEC HFS data set allocated to the **UNVDB** ddname must be mounted in order to perform the conversion. If it is not mounted when the conversion program starts, the conversion program will attempt to dynamically mount the HFS data set.

In order for the program to dynamically mount the HFS data set, the user ID used to run the job must have an OMVS UID value of 0 or READ access to the **BPX.SUPERUSER** profile in the FACILITY class.

The JCL creates UEC 4.1.0 database files in the root directory of the HFS data set allocated to the **UNVDB** ddname, reads the UEC 3.1.1 databases allocated to ddnames **UNVBKRS**, **UNVGRPS**, and **UNVUSRS**, and converts them to 4.1.0 databases.

The UEC 4.1.0 database files created in the root directory of the HFS data set will be owned by the user ID used to run this job. The owner must be changed to match the user ID with which the UEC STC will execute. The default user ID is **UECUSR**. From a USS shell prompt, enter the command: **chown -R uecusr /tmp/#PHLQ.UNV.UECDB**, where **#PHLQ** is the high-level qualifier used for the UEC HFS data set. If the UEC STC will run with a user ID other than **UECUSR**, replace **uecusr** with that user ID.

The conversion does not modify the UEC 3.1.1 databases.

All steps must end with return code 0.

2.7.2 Universal Products 3.2.0 Upgrade, Existing CSI

The Universal Products 3.2.0 Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Products 3.2.0 package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Products **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

1. Edit and submit the JCL in member **#SETUP**.

The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library.

To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced.

All steps must end with a return code 0.

2. Submit the JCL in member **UNVUG320**. The JCL upgrades the SMP/E Universal Products configuration from 3.2.0 to 4.1.0.

The following changes are made:

- Allocates a new AUNVHBIN SMP/E distribution data set, if necessary.

3. The product databases are allocated as z/OS UNIX HFS file systems. If you prefer to use zFS file systems, zFS data sets must be allocated. If you prefer to use the HFS data sets, then skip this step.

Submit the JCL in member **UNVINZF**. The JCL renames the HFS data sets, defines their zFS counterparts, and formats the zFS data sets.

All steps must end with return code 0.

Additional product configuration is required to utilize zFS data sets. See Section [2.9.5 Universal Products Database Configuration](#) for details.

4. Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets.

All steps must end with a return code 0.

5. Submit the JCL in member **UNVIN05**. The JCL performs a SMP/E APPLY of the product FMIDs and any received PTFs.

Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions:

- Step ends with condition code 0.
- Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**.
- Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.

6. Submit the JCL in member **UNVIN06**. The JCL member performs a SMP/E ACCEPT of product FMIDs and any applied PTFs.

Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions:

- Step ends with condition code 0 or 4.
- Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.

7. Submit the JCL in member **UNVIN08**. The JCL copies sample configuration members to the configuration libraries.

All steps must end with a return code 0.

8. Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL.

The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library.

All steps must end with a return code 0.

2.7.3 Universal Products 3.1.1 Upgrade, Existing CSI

The Universal Products 3.1.1 Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Products 3.1.1 package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Products **INSTALL** library. Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

1. Edit and submit the JCL in member **#SETUP**.

The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library.

To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced.

All steps must end with a return code 0.

2. Submit the JCL in member **UNVUG311**. The JCL upgrades the SMP/E Universal Products configuration from 3.1.1 to 4.1.0.

The following changes are made:

- Allocates new SMP/E target and distribution data sets, and adds SMP/E DDDEFs to the target and distribution zones.
- Allocates new non-SMP/E product data sets and databases.
- Converts existing SMP/E target and distribution libraries from PDSs to PDSEs.

All steps must end with a return code 0.

3. The product databases are allocated as z/OS UNIX HFS file systems. If you prefer to use zFS file systems, zFS data sets must be allocated. If you prefer to use the HFS data sets, then skip this step.

Submit the JCL in member **UNVINZF**. The JCL renames the HFS data sets, defines their zFS counterparts, and formats the zFS data sets.

All steps must end with return code 0.

Additional product configuration is required to utilize zFS data sets. See Section [2.9.5 Universal Products Database Configuration](#) for details.

4. Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets.

All steps must end with a return code 0.

5. Submit the JCL in member **UNVIN05**. The JCL performs a SMP/E APPLY of the product FMIDs and any received PTFs.

Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions:
 - Step ends with condition code 0.
 - Step ends with condition code 4, and message GIM42001W is written in ddname SMPOUT.
 - Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
6. Submit the JCL in member **UNVIN06**. The JCL member performs a SMP/E ACCEPT of product FMIDs and any applied PTFs.

Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions:
 - Step ends with condition code 0 or 4.
 - Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
7. Submit the JCL in member **UNVIN08**. The JCL copies sample configuration members to the configuration libraries.

All steps must end with a return code 0.
8. Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL.

The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library.

All steps must end with a return code 0.

Note: The following step is required only if you have to convert Universal Enterprise Controller (UEC) 3.1.1 databases to 4.1.0 format.

9. Edit and submit the JCL in member **UECDB311**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL.

UEC 4.1.0 databases are stored in an HFS data set. The UEC HFS data set allocated to the **UNVDB** ddname must be mounted in order to perform the conversion. If it is not mounted when the conversion program starts, the conversion program will attempt to dynamically mount the HFS data set.

In order for the program to dynamically mount the HFS data set, the user ID used to run the job must have an OMVS UID value of 0 or READ access to the **BPX.SUPERUSER** profile in the FACILITY class.

The JCL creates UEC 4.1.0 database files in the root directory of the HFS data set allocated to the **UNVDB** ddname, reads the UEC 3.1.1 databases allocated to ddnames **UNVBKRS**, **UNVGRPS**, and **UNVUSRS**, and converts them to 4.1.0 databases.

The UEC 4.1.0 database files created in the root directory of the HFS data set will be owned by the user ID used to run this job. The owner must be changed to match the user ID with which the UEC STC will execute. The default user ID is **UECUSR**. From a USS shell prompt, enter the command **chown -R uecusr /tmp/#PHLQ.UNV.UECDB**, where **#PHLQ** is the high-level qualifier used for the UEC HFS data set. If the UEC STC will run with a user ID other than **UECUSR**, replace **uecusr** with that user ID.

The conversion does not modify the UEC 3.1.1 databases.

All steps must end with return code 0.

2.7.4 Universal Products 3.1.0 Upgrade, Existing CSI

The Universal Products 3.1.0 Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Products 3.1.0 package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Products **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

1. Edit and submit the JCL in member **#SETUP**.

The **#SETUP** JCL creates customized installation JCL used by the following installation steps and for product maintenance and customization. All the JCL is created as members in the **INSTALL** library.

To change customizations after the job is executed, edit the **#SETUP** member with the new customizations and resubmit the job. All **INSTALL** library members will be replaced.

All steps must end with a return code 0.

2. Submit the JCL in member **UNVUG310**. The JCL upgrades the SMP/E Universal Products configuration from 3.1.0 to 4.1.0.

The following changes are made:

- Adds the SMP/E **RETRYDDN UCLIN OPTIONS** parameter to the **GLOBAL** zone. The **RETRYDDN** parameter specifies a list of selected **DDDEFs** that SMP/E automatically compresses should a space related **ABEND (x37)** occur.
- Adds **DDDEF** for **CSSLIB** (Callable System Services library).
- Allocates new SMP/E target and distribution data sets, and adds SMP/E **DDDEFs** to the target and distribution zones.
- Allocates new non-SMP/E product data sets and databases.
- Converts existing SMP/E target and distribution libraries from **PDSs** to **PDSEs**.

All steps must end with a return code 0.

3. The product databases are allocated as z/OS UNIX HFS file systems. If you prefer to use zFS file systems, zFS data sets must be allocated. If you prefer to use the HFS data sets, then skip this step.

Submit the JCL in member **UNVINZF**. The JCL renames the HFS data sets, defines their zFS counterparts, and formats the zFS data sets.

All steps must end with return code 0.

Additional product configuration is required to utilize zFS data sets. See Section [2.9.5 Universal Products Database Configuration](#) for details.

4. Submit the JCL in member **UNVIN04**. The JCL performs an SMP/E **RECEIVE** of the product **FMIDs** and available **PTFs** from the distribution data sets.

All steps must end with a return code 0.

5. Submit the JCL in member **UNVIN05**. The JCL performs a SMP/E APPLY of the product FMIDs and any received PTFs.

Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions:

- Step ends with condition code 0.
- Step ends with condition code 4, and message GIM42001W is written in ddname SMPOUT.
- Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.

6. Submit the JCL in member **UNVIN06**. The JCL member performs a SMP/E ACCEPT of product FMIDs and any applied PTFs.

Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions:

- Step ends with condition code 0 or 4.
- Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.

7. Submit the JCL in member **UNVIN08**. The JCL copies sample configuration members to the configuration libraries.

All steps must end with a return code 0.

8. Edit and submit the JCL in member **UNVIN09**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL.

The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library.

All steps must end with a return code 0.

Note: The following steps is required only if you have to convert Universal Enterprise Controller (UEC) 3.1.1 databases to 4.1.0 format.

9. Edit and submit the JCL in member **UECDB311**. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL.

UEC 4.1.0 databases are stored in an HFS data set. The UEC HFS data set allocated to the **UNVDB** ddname must be mounted in order to perform the conversion. If it is not mounted when the conversion program starts, the conversion program will attempt to dynamically mount the HFS data set.

In order for the program to dynamically mount the HFS data set, the user ID used to run the job must have an OMVS UID value of 0 or READ access to the **BPX.SUPERUSER** profile in the FACILITY class.

The JCL creates UEC 4.1.0 database files in the root directory of the HFS data set allocated to the **UNVDB** ddname, reads the UEC 3.1.1 databases allocated to ddnames **UNVBKRS**, **UNVGRPS**, and **UNVUSRS**, and converts them to 4.1.0 databases.

The UEC 4.1.0 database files created in the root directory of the HFS data set will be owned by the user ID used to run this job. The owner must be changed to match the user ID with which the UEC STC will execute. The default user ID is **UECUSR**. From a USS shell prompt, enter the command **chown -R uecusr /tmp/#PHLQ.UNV.UECDB**, where **#PHLQ** is the high-level qualifier used for the UEC HFS data set. If the UEC STC will run with a user ID other than **UECUSR**, replace **UECUSR** with that user ID.

The conversion does not modify the UEC 3.1.1 databases.

All steps must end with return code 0.

2.8 Universal Products for z/OS – Cumulative PTF Maintenance

This section describes the procedures for installing a cumulative PTF maintenance for Universal Products 4.1.0 for z/OS.

2.8.1 Installation Procedures

Installation of a cumulative PTF maintenance is comprised of the following procedures:

1. Download the cumulative PTF maintenance distribution file.
2. Transfer the cumulative PTF file from Windows to z/OS.
 - a. Extract the cumulative PTF file from the distribution file.
 - b. Allocate a data set for the PTF file.
 - c. Transfer the PTF file to the data set on z/OS.
 - d. Unpack the data set.
3. Install the cumulative PTF file.

2.8.2 Cumulative PTF Maintenance Distribution File

To obtain a cumulative PTF maintenance, you must download the corresponding distribution file from the Support / Maintenance area of the Stonebranch, Inc. website, <http://www.stonebranch.com>.

Note: A customer user ID and password — provided by Stonebranch, Inc. — are required to access this area.

Distribution File Content

Table 2.4, below, lists the files included in a cumulative PTF maintenance distribution file

File Name	Description
UNV410xx.XMT	Cumulative PTF file for Universal Products 4.1.0 level xx.
README.TXT	Documentation on the package and transfer methods.

Table 2.4 Universal Products for z/OS - Cumulative PTF Maintenance Files

If you do not have these files, contact Stonebranch, Inc. [Customer Support](#) for a complete distribution file.

2.8.3 Transferring a Cumulative PTF File to z/OS

A cumulative PTF file must be transferred to the z/OS system on which the maintenance is being installed. There are two methods of transferring the PTF files:

- [Express/OS Method](#)
- [Manual Method](#)

Express/OS Method

The Express/OS method automatically performs the following steps:

1. Unpacks the XMT PTF file into a specified directory.
2. Transfers the XMT PTF file to the z/OS system using FTP in passive mode.
3. Optionally, submits a batch job to perform a TSO RECEIVE of the file.

When the Express/OS package installation is complete, continue with the installation of the product maintenance on the z/OS system (see [Installing a Cumulative PTF File](#), below).

If you cannot use Express/OS to transfer and unpack the file, see [Manual Method](#).

PTF HOLD's

If a PTF requires an action to be taken prior to being APPLY'ed or ACCEPT'ed, the PTF is delivered with an SMP/E HOLD. The PTF's cover letter lists the recommended actions that should be taken to resolve the HOLD. When the HOLD is resolved, the PTF can be APPLY'ed and ACCEPT'ed.

PTF cover letters are printed in the **SMPDOUT** ddname of the job that received the PTFs, or PTF cover letters can be viewed in the **SMPPTS** library after being RECEIVE'd.

When all PTF HOLD's have been resolved, all HOLDs are bypassed with the **BYPASS** parameter.

BYPASS Parameter

On the APPLY command, the BYPASS parameter is specified as follows:

APPLY PTFS

```
BYPASS( HOLDSYSTEM( ACTION ) )
FORFMID(TUNV410, TUTL410, TUBR410, TUCM410, TUEC410, TUDM410,
        TUSD410, TUSP410, TUEM410, TUXD410) .
```

On the ACCEPT command, the BYPASS parameter is specified as follows:

ACCEPT PTFS

```
BYPASS( HOLDSYSTEM( ACTION ) )
FORFMID(TUNV410, TUTL410, TUBR410, TUCM410, TUEC410, TUDM410,
        TUSD410, TUSP410, TUEM410, TUXD410) .
```

If successful, the APPLY or ACCEPT step will end with condition code 4 and message GIM42001W is printed in the **SMPDOUT** ddname.

Manual Method

The manual method may be required in cases where you cannot use FTP to transfer files from the Windows workstation to the z/OS system. Instead of FTP, another file transfer protocol must be used.

The manual method requires the PTF file to be extracted from the distribution file, then transferred manually from the Windows workstation to the z/OS system.

The following steps identify how to perform manual transfer of the cumulative PTF file.

1. Allocate an z/OS data set to receive the PTF file as:
 - DSORG=PS
 - RECFM=FB
 - LRECL=80
 - BLKSIZE=3120
2. Transfer the PTF file - using a binary transfer from your local workstation - to the XMIT data set allocated on z/OS in Step 1.
3. Enter the following TSO command to RECEIVE the XMT file:
TSO RECEIVE INDA('cum PTF data set name allocated in step 1')
4. Press <Enter> at the INMR906A message prompt.
5. Continue with the installation of the maintenance (see Section [2.8.4 Installing a Cumulative PTF File](#)).

Files ending with the XMT suffix are binary files in a TSO TRANSMIT format.

2.8.4 Installing a Cumulative PTF File

Cumulative PTF Maintenance is installed using members **UNVMTRCV**, **UNVMTAPL**, and **UNVMTACC** in the **INSTALL** product library.

The installation itself consists of running a number of batch jobs. The output of these batch jobs should be kept until a correct installation has been verified.

1. Member **UNVMTRCV** performs an SMP/E RECEIVE of the PTFs.

Change the data set allocated on ddname **SMPPTFIN** to the PTF data set.

Modify the JCL to meet local requirements, and submit the job.

2. Member **UNVMTAPL** performs an SMP/E APPLY of the PTFs.

Modify the JCL to meet local requirements, and submit the job.

The job should end with condition code 0. If the job does not end with condition code 0, it is most likely due to a HOLD on one of the PTFs. This is indicated by message GIM30206E printed on ddname **SMPDOUT**. In this case, refer to Section [PTF HOLD's](#).

3. Fully test the maintenance. Only after all testing is completed should you proceed to the next step.
4. Member **UNVMTACC** performs an SMP/E ACCEPT of the PTFs. Modify the JCL to meet local requirements, then submit the job.

The job should end with condition code 0 or 4. If it does not, it is most likely due to a HOLD on one of the PTFs. This is indicated by message GIM30206E printed on ddname **SMPDOUT**. In this case, see Section [PTF HOLD's](#).

2.9 Universal Products for z/OS – Configuration

This section describes the z/OS configuration steps required for the Universal Broker and Universal Enterprise Controller products. The Universal Broker is required.

The following configuration steps are described:

- Installing the started tasks and setting up the started task user IDs and group IDs.
- APF authorizing the **SUNVLOAD** library and setting up program control security.
- Installation of the SMF IEFACTRT exit.
- Configuration of Universal Spool HFS or zFS databases and Universal Enterprise Controller HFS databases.

Member **UNVINRF** in the **INSTALL** library provides a sample JCL that can be used to execute the RACF commands described in the following sections as a TSO batch job.

Carefully read the comments in the member before submitting the JCL. The JCL is provided only as a sample. Local procedures should be followed as necessary for defining and altering security resources.

2.9.1 Started Task Configuration

The Universal Enterprise Controller and Universal Broker products execute as z/OS started tasks.

The following started task JCL procedures are provided in the **SUNVSAMP** library:

- **UECTLR** is the JCL procedure for the Universal Enterprise Controller started task.
- **UBROKER** is the JCL procedure for the Universal Broker started task.

The JCL procedures should have been copied to a system procedure library by installation job **UNVIN09**. If this was not done, copy the JCL members to the appropriate procedure library based on local installation requirements.

This section describes the following configuration tasks:

- Defining the started task user and group profiles.
- Permitting the started task user profile to required security resources.
- Associate started tasks with the user and group profiles.

Define Started Procedure User and Group Profiles

Universal Enterprise Controller and Universal Broker use the z/OS UNIX System Services (USS) environment. The user and group profiles associated with the procedures must contain valid OMVS segments in order to initialize.

The Universal Broker started task requires an OMVS UID value of 0 (superuser). UID 0 is required for the product servers to switch to the appropriate authenticated user IDs using the `setuid` and `setreuid` functions.

The Universal Enterprise Controller (UEC) started task requires either an OMVS UID value of 0 (superuser) or READ access to the `BPX.SUPERUSER` profile in the FACILITY class in order for UEC to mount its HFS data set when started. Alternatively, the HFS data set may be statically mounted. For a detailed explanation of the use of HFS data sets, see [Section 2.9.5 Universal Products Database Configuration](#).

The following procedure describes how to define a user and group profile to be associated with the started procedure.

(All the commands listed in this procedure are TSO RACF commands.)

1. Add a Universal Enterprise Controller group profile with the following command:

```
ADDGROUP UECGRP          +
          OWNER(SYS1)    +
          OMVS(GID(5000))
```

The OMVS segment uses USS group ID (GID) 5000. This is an arbitrary value; it should be replaced with a group ID based on your local USS configuration.

2. Add a Universal Broker group profile with the following command:

```
ADDGROUP UBRGRP          +
          OWNER(SYS1)    +
          OMVS(GID(5001))
```

The OMVS segment uses USS group ID (GID) 5001. This is an arbitrary value; it should be replaced with a group ID based on your local USS configuration.

3. Add a Universal Enterprise Controller user profile with the following command:

```
ADDUSER  UECUSR          +
          DFLTGRP(UECGRP) +
          OWNER(SYS1)    +
          NOPASSWORD     +
          OMVS(UID(0))
```

The OMVS segment uses USS user ID (UID) 0. UID 0 is required only if UEC is to automatically mount its HFS data set or does not have READ access to the `BPX.SUPERUSER` profile in the FACILITY class; otherwise, set the UID value to an unused value that meets your local requirements.

4. Add a Universal Broker user profile with the following command:

```
ADDUSER  UBRUSR          +
          DFLTGRP (UBRGRP) +
          OWNER (SYS1)    +
          NOPASSWORD      +
          OMVS (UID(0))
```

The OMVS segment uses USS user ID (UID) 0. This is required.

Permitting Access to Security Resources

The Universal Broker utilizes z/OS USS facilities in order to switch security environments under which services are performed. The security environment is changed using the `setuid` and `setreuid` USS functions.

Access to the `setuid` and `setreuid` functions can be controlled with the security profile `BPX.DAEMON` in the FACILITY class. (Refer to IBM UNIX System Services Planning manual for complete details on setting up a daemon process.)

z/OS UNIX is configured by a site to use either UNIX native security or z/OS security. If security profiles `BPX.DAEMON` or `BPX.SERVER` are defined to the FACILITY class, than z/OS security is active; otherwise, native UNIX security is in effect.

If native UNIX security is in effect, no access to the `BPX.DAEMON` profile is required. You can skip to the next section.

If z/OS security is in effect, permit the Universal Broker user profile `UBRUSR` read access to the `BPX.DAEMON` profile in the FACILITY class.

RACF commands to permit access are illustrated below. After access is permitted, the FACILITY class is refreshed.

```
PE BPX.DAEMON CLASS(FACILITY) ID(UBRUSR) ACCESS(READ)

SETR RACLIST(FACILITY) REFRESH
```

Data Set Access

The **UBRUSR** and **UECUSR** user profiles require access to data sets allocated in their started task JCL. This section defines the minimum level of access required.

The Universal Broker user profile **UBRUSR** requires the following data set access levels:

ddname	Access	Description
STEPLIB	READ	Program library.
UNVCONF	ALTER	Product configuration data.
UNVRFC	READ	SAP RFC file used by Universal Connector.
UNVCOMP	ALTER	Product component definition data.
UNVNLS	READ	Product national language support data.
UNVCREF	READ	Universal Command command reference definitions.
UNVDB	UPDATE	Universal Broker component database.
UNVSPool	UPDATE	Universal Broker spool database.
UNVTMPL	READ	Universal Broker template configuration data.
UNVTRACE	UPDATE	Universal Broker application trace PDS.
UNVTRMDL	ALTER	Universal Broker application trace model data set. The Broker creates new trace data sets using the model data set as a high-level qualifier and its data set attributes for the created data sets. Additionally, the UBRUSR profile will require UPDATE access to the catalog in which these created data sets are added.
UNVLOG	UPDATE	Universal Broker log messages when message destination is set to LOG.

Table 2.5 Universal Broker for z/OS - Data Set Access Levels

The Universal Enterprise Controller user profile **UECUSR** requires the following data set access levels:

ddname	Access	Description
STEPLIB	READ	Program library.
UNVCONF	READ	Product configuration data.
UNVCOMP	READ	Product component definition data.
UNVNLS	READ	Product national language support data.
UNVDB	UPDATE	Universal Enterprise Controller database.
UNVMSGs	UPDATE	Universal Enterprise Controller message trace data.
UNVPRSR	UPDATE	Universal Enterprise Controller parser trace data.
UNVTRACE	UPDATE	Universal Enterprise Controller application trace data.

Table 2.6 Universal Enterprise Controller for z/OS - Data Set Access Levels

2.9.2 Associate Started Tasks with User and Group Profiles

The started tasks must be associated with their user and group profiles. IBM provides two different methods to accomplish this using RACF:

1. STARTED Class Profile
2. Started Procedures Table

Both methods are described below. Only one, not both, of the methods is required. They are provided as examples for your reference. Your local security procedures and processes should be followed.

STARTED Class Profile

The following procedure describes how to associate a user and group profile with the started procedures using the RACF class STARTED method.

(All the commands listed in the instructions are TSO RACF commands.)

1. Define a STARTED class profile for the Controller started procedure with the following command:

```
RDEFINE  STARTED          +
        UECTLR. *        +
        STDATA(USER(UECUSR) GROUP(UECGRP))
```

The started procedure member name used in the above command is **UECTLR**. If this has been change, the name in the REDEFINE command must also be changed to match.

2. Define a STARTED class profile for the Broker started procedure with the following command:

```
RDEFINE  STARTED          +
        UBROKER. *        +
        STDATA(USER(UBRUSR) GROUP(UBRGRP))
```

The started procedure member name used in the above command is **UBROKER**. If this has been change, the name in the REDEFINE command must also be changed to match.

3. The STARTED class must be refreshed to recognize the new profile definitions. The following command assumes that the STARTED class is active and RACLIST'ed.

```
SETROPTS RACLIST(STARTED) REFRESH
```

Started Procedures Table

This section describes how to associate a user and group profile with the started procedures using the RACF started procedure table ICHRIN03 method.

The ICHRIN03 table resides in a system LPA library, such as **SYS1.LPALIB**. Changes to the table require a system IPL using the CLPA option for them to take effect. RACF loads the table at IPL.

1. Add the following entry to the **ICHRIN03** table.

The table is an assembly language program that is assembled and link edited into a system LPA library.

```

DC      CL8'UECTLR  '   PROC NAME
DC      CL8'UECUSR  '   UEC USER PROFILE
DC      CL8'UECGRP  '   UEC GROUP PROFILE
DC      XL1'00'
DC      XL7'00'
DC      CL8'UBROKER '   PROC NAME
DC      CL8'UBRUSR  '   UBR USER PROFILE
DC      CL8'UBRGRP  '   UBR GROUP PROFILE
DC      XL1'00'
DC      XL7'00'

```

2. Increment the table count field by two. (The count field is the first 2 bytes of the table.)
3. Assemble and link edit the ICHRIN03 table.

IBM provides a sample ICHRIN03 table and the JCL to assemble and link edit it in **SYS1.SAMPLIB(RACTABLE)**.

4. IPL the system with the CLPA option.

2.9.3 Library and Program Authorization

Some of the services provided by the products require proper library and program authorization. APF authorization is required for the **SUNVLOAD** library and RACF program control authorization is required for the server programs.

This section describes how to set up library APF authorization and program control.

APF Authorization

The **SUNVLOAD** library requires APF authorization.

[Table 2.7](#), below, describes which programs and the services they provide that impose this requirement.

Program	Description
UBROKER	Universal Broker program performs the following services that require APF authorization: <ul style="list-style-type: none"> • The USS <code>mount</code> API to mount the HFS databases. • Dynamically install the SMF SYSSTC.IEFACTRT exit routine with the CSVDYNEX macro. • Establish CSA storage for the SMF exit routine routing table. • Establish address space resource manager for resource recovery using RESMGR ADD macro.
UBRRM001	UBROKER resource manager module performs the following services that require APF authorization: <ul style="list-style-type: none"> • Release CSA storage resources.
UCMSCPST	Universal Command Server STC Command Processor performs the following services that require APF authorization: <ul style="list-style-type: none"> • Establish address space resource manager for resource recovery using RESMGR ADD macro. • Activate MCS console using MCSOPER REQUEST=ACTIVATE macro. • Perform inter-address space communication with STC address space via CSA data structures.
UCMSR000	UCMSS00 resource manager module performs the following services that require APF authorization: <ul style="list-style-type: none"> • Access to CSA storage resources.
UCMSS000	Universal Command Server STC Support module performs the following services that require APF authorization: <ul style="list-style-type: none"> • Perform inter-address space communication with STC address space via CSA data structures. • Switches the address space user ID using RACROUTE macro. • Establish address space resource manager for resource recovery using RESMGR ADD macro.
UNVACTRT	Universal IEFACTRT SMF exit requires APF authorization per z/OS.

Table 2.7 Universal Products for z/OS - Program Services Requiring APF Authorization

The **SUNVLOAD** library can be APF authorized using one of the following methods:

- The **SETPROG** APF console command. The library remains APF authorized until the next IPL. The **SETPROG** APF console command is documented in the IBM MVS System Commands manual.
- A **PROGxx** member of **PARMLIB** referenced by a **SET PROG=xx** console command or a **PROG=xx** statement in IEASYSxx **PARMLIB** member. The **PROGxx PARMLIB** member is documented in the IBM MVS Initialization and Tuning Reference manual, and the **SET** console command is documented in the IBM MVS System Commands manual.
- Updating the IEAAPFxx **PARMLIB** member and perform an IPL. The IEFAPFxx **PARMLIB** member is documented in the IBM MVS Initialization and Tuning Reference manual.

The **SETPROG** APF console command to add data set **UNV . SUNVLOAD** to the APF list is illustrated below. **SUNVLOAD APF** authorization should be permanently established based on local site requirements.

```
SETPROG APF ,ADD ,DSNAME=UNV . SUNVLOAD ,VOLUME=STG001
```

The data set can be added permanently to the APF list using the appropriate **PROGxx PARMLIB** member. The APF statement of the **PROGxx PARMLIB** member is illustrated below. The APF statement adds data set **UNV . SUNVLOAD** on volume **STG001** to the APF list.

```
APF ADD DSNAME(UNV .SUNVLOAD) VOLUME(STG001)
```

Program Control

Specific services of the z/OS UNIX environment require all programs loaded in the address space to be defined in the security product as controlled. RACF provides program and library control features. By defining a program as controlled, you are implying a certain level of trust, as opposed to a program being run that is not defined as controlled.

[Table 2.8](#), below, identifies the programs in the **SUNVLOAD** library that must be defined as program controlled.

Program	Description
UCMSRV	Universal Command Server
UDMSRV	Universal Data Mover Server
UCTSRV	Universal Control Server
UECTLR	Universal Enterprise Controller

Table 2.8 Universal Products for z/OS - Controlled Programs in SUNVLOAD Library

The requirement for program control for each of the above programs is the same:

1. The `_passwd` C function used to authenticate user IDs requires it.
2. The `setuid` C function used to switch to the authenticated user ID requires it.

Defining a program or library to RACF program control is accomplished by defining the library to the PROGRAM class.

The following RACF TSO command illustrates defining the library to the PROGRAM class:

```
RDEF PROGRAM ** ADDMEM ('UNV.SUNVLOAD' //NOPADCHK) UACC(READ)
```

After defining the library as RACF program controlled, the PROGRAM class must be refreshed. The following RACF TSO command illustrates:

```
SETROPTS WHEN(PROGRAM) REFRESH
```

When all programs executing in an address space are program controlled, the address space is considered a clean environment. All programs loaded dynamically into a clean environment must also be program controlled else the address space is terminated.

Universal Products may load some IBM modules at runtime to perform requested services. All IBM modules loaded dynamically are loaded from `SYS1.LINKLIB`. Each member or all members of `SYS1.LINKLIB` must be set up as program controlled.

The following modules are loaded from `SYS1.LINKLIB`:

- IGGCSI00 (Catalog Search Interface)
- IEBCOPY (PDS/E copy utility)

`SYS1.LINKLIB` normally is set up so that all members are program controlled. Verify how your installation has defined `SYS1.LINKLIB` in regards to the program control facility. If necessary, define the individual members listed above, or all members of `SYS1.LINKLIB`, as program controlled.

2.9.4 SMF Exit Routine

The Universal Broker establishes the environment to support Universal Command started task execution. Part of the environment consists of SMF exit routine **UNVACTRT** added to SMF exit point **SYSSTC.IEFACTRT**.

The Broker's **UCMD_STC_SUPPORT** option specifies whether or not the environment is required:

- **yes** (the default) specifies that the environment should be set up.
- **no** specifies that the environment should not be set up.

This section explains what is required for the Universal Broker to utilize the SMF exit routine **UNVACTRT** in support for Universal Command start task execution.

Note: If Universal Command started task support is not required, skip this section.

Defining the SMF SYSSTC.IEFACTRT Exit Point

In order for any **IEFACTRT** exit to be given control for STC subsystems, the SMF exit point **IEFACTRT** must be defined in the **SMFPRMxx** PARMLIB member for STC subsystems with the **SUBSYS STC EXITS** parameter either implicitly, by excluding the **EXITS** parameter, or explicitly.

The required **SMFPRMxx** **SUBSYS** statement is:

```
SUBSYS (STC, EXITS (IEFACTRT, . . .))
```

If the **SMFPRMxx** PARMLIB member has to be changed to include **IEFACTRT** in the **SUBSYS STC** exits list, the **SMFPRMxx** member can be activated dynamically.

The following command will restart SMF using the PARMLIB member **SMFPRMxx**.

```
SET SMF=xx
```

The **SYSSTC.IEFACTRT** exit point and its exit routines can be displayed with the **DISPLAY PROG,EXIT** console command. The command is documented in the IBM MVS System Commands manual.

The console command is:

```
D PROG,EXIT,EN=SYSSTC.IEFACTRT,DIAG
```

Installing the UNVACTRT SMF Exit Routine

The SMF exit routine **UNVACTRT** is provided as part of the Universal Products package. **UNVACTRT** must be added to SMF exit point **SYSSTC.IEFACTRT**. The exit routine can be added automatically by the Broker STC when it starts or statically in a PARMLIB member.

Automatic Installation

The Broker dynamically adds SMF exit routine **UNVACTRT** to the **SYSSTC.IEFACTRT** exit point if the **SMF_EXIT_LOAD_LIBRARY** configuration option is defined in the Broker configuration file. The option indicates that the Broker dynamically add **UNVACTRT** and specifies the load library from which **UNVACTRT** is loaded.

Static Installation

If the Universal Broker configuration option **SMF_EXIT_LOAD_LIBRARY** is not specified in the Universal Broker configuration file, Universal Broker assumes that exit routine **UNVACTRT** already is added and will not attempt to add it when it starts.

Exit routine **UNVACTRT** can be added to **SYSSTC.IEFACTRT** permanently (instead of Universal Broker adding it dynamically) using the appropriate **PROGxx** PARMLIB member.

The EXIT statement of the **PROGxx** PARMLIB member is illustrated below. The EXIT statement adds exit routine **UNVACTRT** from the **UNV.SUNVLOAD** load library to the SMF exit point **SYSSTC.IEFACTRT**. The exit is no longer given control if it encounters an ABEND.

```
EXIT ADD EXITNAME(SYSSTC.IEFACTRT) MODNAME(UNVACTRT)
      DSNAME(UNV.SUNVLOAD) ABENDNUM(1)
```

With a static installation, the **UNVACTRT** exit routine is added when the system is IPL'ed and the **PROGxx** member is processed. SMF does support dynamic addition and deletion of SMF exit routines. Dynamic exit support enables you to add an SMF exit routine without an IPL.

SMF supports dynamic addition and deletion of SMF exits using the following methods:

The **SETPROG** EXIT console command. The exit remains active until the next IPL. The **SETPROG** EXIT console command is documented in the IBM MVS System Commands manual.

A **PROGxx** PARMLIB member referenced by a SET console command. The **PROGxx** PARMLIB member is documented in the IBM MVS Initialization and Tuning Reference manual. The SET console command is documented in the IBM MVS System Commands manual.

2.9.5 Universal Products Database Configuration

z/OS UNIX File System Introduction

The z/OS implementation of Universal Products databases utilizes the z/OS UNIX file system. HFS or zFS data sets are used by Universal Broker started tasks; HFS data sets are used by Universal Enterprise Controller started tasks.

The Universal Enterprise Controller started task JCL allocates one HFS data set to ddname **UNVDB**.

The hierarchical file system is a file system used by z/OS UNIX System Services (USS). It is a POSIX conforming hierarchical file system stored in one or more HFS or zFS data sets bound together into one hierarchical directory structure. A single HFS or zFS data set consists of directory tree and files. Refer to the IBM UNIX System Services Planning manual for a complete discussion of the z/OS UNIX file system and its administration.

An HFS or zFS data set must be mounted before a program can access any file or directory within it. A mount operation binds the root directory of the HFS or zFS data set to an existing directory in the hierarchical file system referred to as the mount point. After the mount operation completes, the HFS or zFS data set's directory structure becomes part of the file system hierarchy starting at the mount point. An HFS or zFS data set can only be mounted once at a time.

The mount operation makes the files and directories within the HFS or zFS data set accessible to all users. User access is controlled with directory and file permissions contained within the HFS or zFS data set. Initially, an HFS or zFS data set's root directory is owned by the user that allocated the data set and the directory permissions are set so that only that user has read, write, and execute permissions (permission mode 700). No other users have access.

HFS Configuration

HFS data sets are created by the installation JCL. The HFS data sets are used by Universal Broker and Universal Enterprise Controller. These started tasks allocate the HFS data sets to specific ddnames. Universal Broker allocates the HFS data set used for the databases to ddname **UNVDB** and the HFS data set used for the spool to ddname **UNVSPPOOL**. Universal Enterprise Controller allocates the HFS data set used for the databases to ddname **UNVDB**. No further HFS configuration is required beyond allocating the appropriate HFS data set to the ddnames.

Alternatively, the HFS data set names can be specified using the configuration options **UNIX_DB_DATA_SET** and **UNIX_SPOOL_DATA_SET**. When these configuration options are used, the ddnames **UNVDB** and **UNVSPPOOL** are not used. The ddname statements can be removed from the started task procedures.

zFS Configuration

An alternative to using HFS data sets is to use zFS data sets. zFS data sets are created optionally as part of the installation steps with JCL member **UNVINZF**.

A zFS data set is referred to as a zFS aggregate. There are two types of aggregates, depending on whether it contains one or more read / write zFS file systems:

1. A zFS aggregate that contains only one file system is a compatibility mode aggregate.
2. A zFS aggregate that contains more than one file system is a multi-file system aggregate.

Universal Products uses compatibility mode aggregates only.

Note: IBM has recommended the use of zFS over HFS data sets. Future versions of Universal Products shall make zFS the default and preferred installation for z/OS UNIX file systems.

When a zFS data set is mounted, the ZFS address space obtains an exclusive enqueue on the data set, preventing it from being allocated by another address space. For this reason, when using zFS data sets in the Universal Broker started tasks, the data sets cannot be allocated to the **UNVDB** or **UNVSPPOOL** ddnames.

When using zFS data sets instead of HFS data sets, remove the **UNVDB** and **UNVSPPOOL** ddname statements from the Universal Broker started task procedure.

Note: If your Universal Broker started task procedure contains a PARM reference to **DBHLQ**, it also must be removed.

The Universal Broker zFS data set names must be specified with the configuration options **UNIX_DB_DATA_SET** and **UNIX_SPOOL_DATA_SET**. The configuration options can be used to specify HFS data sets as well if this method is preferred over specifying them as a ddname allocation.

Mounting and Unmounting the Databases

When the Universal Broker and Universal Enterprise Controller started tasks are started, they check if their HFS or zFS (for Universal Broker) data sets have been mounted. If they are mounted, the started tasks will attempt to use them. If they are not mounted, the started tasks will mount the data sets dynamically.

Dynamic Mounts

The started tasks will mount the HFS or zFS data sets if they are not mounted. The data sets are mounted at mount points defined in the directory specified by configuration option MOUNT_POINT, which defaults to the /tmp directory. The mount points are subdirectories named after the data set names. For example, if the HFS or zFS data set name is UNV.UNVDB, the mount point is /tmp/UNV.UNVDB.

When the started tasks mount a zFS data set, the mount parameter **AGGRGROW** is used to specify that the zFS data set should automatically utilize secondary extents to expand if it runs out of allocated space.

The HFS or zFS data sets are not unmounted when the started tasks are stopped. It is not known whether or not other users are using the mounted data sets.

Manual Mounts

The started tasks will use the existing mounts of the HFS or zFS data sets. Dynamic mounts provide the easiest administration, but you may want to manually mount the data sets to take advantage of several available mount options. For example, the FSFULL PARM value can be used to issue operator messages when a file system reaches a specified percent full.

When mounting zFS data sets, the mount parameter **AGGRGROW** should be used to specify that the zFS data set should automatically utilize secondary extents to expand if it runs out of allocated space.

When the HFS or zFS data sets are manually mounted, the mount point can be any z/OS UNIX directory. The name of the directory does not matter. The started tasks will locate the mount point regardless of location or name.

HFS or zFS data sets can be mounted using the TSO MOUNT command or with PARMLIB member **BPXPRMxx** at IPL. The TSO MOUNT command mounts it for the current IPL only while the **BPXPRMxx** member will mount the data set for each IPL.

HFS or zFS data sets can be unmounted using the TSO UNMOUNT command or with the MODIFY BPXOINIT console command.

TSO Commands

The TSO commands to mount and unmount HFS data set **UNV.UNVDB** at mount point **/opt/unvdb** are illustrated below:

HFS Mount Command:

```
MOUNT FILESYSTEM('UNV.UNVDB') MOUNTPOINT('/opt/unvdb') TYPE(HFS)
```

zFS Mount Command:

```
MOUNT FILESYSTEM('UNV.UNVDB') MOUNTPOINT('/opt/unvdb') TYPE(ZFS)
  PARM(AGGRGROW)
```

HFS and zFS Unmount Command:

```
UNMOUNT FILESYSTEM('UNV.UNVDB')
```

The user ID that issues the mount or unmount commands must have an OMVS UID of 0 or READ access to the **BPX.SUPERUSER** profile in the FACILITY class.

Console Commands

The console commands to unmount HFS or zFS data **UNV.UNVDB** is illustrated below in addition to the console command to list currently mounted HFS data sets.

Unmount Command:

```
F BPX0INIT,FILESYS=UNMOUNT,FILESYSTEM=UNV.UNVDB
```

Note: A console reply message will ask for confirmation.

Display Command:

```
D OMVS,FILE
```

BPXPRMxx

The BPXPRMxx statement to mount HFS or zFS data set **UNV.UNVDB** at mount point **/opt/unvdb** is illustrated below:

HFS Mount:

```
MOUNT FILESYSTEM('UNV.UNVDB') TYPE(HFS) MODE(RDWR)
  MOUNTPOINT('/opt/unvdb')
```

zFS Mount:

```
MOUNT FILESYSTEM('UNV.UNVDB') TYPE(ZFS) MODE(RDWR)
  MOUNTPOINT('/opt/unvdb') PARM('AGGRGROW')
```

Both of the HFS or zFS data sets must be mounted with mode read/write, which is the default.

Data Set Initialization

When the started tasks start, they find the mount point for their HFS or zFS data sets. Regardless of whether the HFS or zFS data sets were dynamically mounted or statically mounted, the started tasks check for an initialization flag file named `.inited` in the root directory of the mounted data set.

If the file is not found, which is the case when they are first mounted, the started tasks change the owner of the root directory to the user ID with which they are executing and change the permission mode to the configuration value of `MOUNT_POINT_MODE`, which defaults to 750.

If you want to customize either the owner or permission of the directories, manually create the `.inited` file in the root directory of the HFS or zFS data set to prevent the started tasks from performing the initialization when they start. The USS command `touch .inited` can be used to create an empty file.

Memory Management

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

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

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

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

1. Mount the **UECDB** HFS data set.
2. Inside the mount point, create a text file named **DB_CONFIG**.
3. Inside the **DB_CONFIG** file, add the following string:

```
set_tmp_dir *dbpath*
```

Where **dbpath** is the path to the location in which the database files reside.

4. Start / restart UEC.

2.10 Universal Products for z/OS – Customization

This section provides the following information for the customization of Universal Products components:

- Configuration
- Product License
- JCL procedure

2.10.1 Universal Broker Customization

Configuration

Universal Broker for z/OS uses a configuration file for its customizations. The configuration file is a sequential file that is allocated to ddname **UNVCONF** in the started procedure's JCL. The default configuration file is member **UBRCFG00** of the **UNVCONF** library.

JCL Procedure

A JCL procedure is provided in member **UBROKER** in library **SUNVSAMP**.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information.

See Section [2.11 Universal Products for z/OS – Time Zone Environment Variable](#) for more information the TZ environment variable.

2.10.2 Universal Certificate Customization

JCL Procedure

A JCL procedure is provided in member **UCRPRC** in library **SUNVSAMP**. Using the procedure simplifies future product upgrades and reduces the amount of JCL statements required in the job JCL.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See Section [2.11 Universal Products for z/OS – Time Zone Environment Variable](#) for more information the TZ environment variable.

The JCL procedure can be made available for use by either:

- Copying it to a JES procedure library, such as **SYS1 . PROCLIB**
- Using the **JCLLIB** JCL statement in the job JCL to include the **SUNVSAMP** library in the procedure library search order.

For example:

```
//jobname JOB . . .  
//          JCLLIB ORDER=UNV.SUNVSAMP  
//STEP1    EXEC UCRPRC
```

2.10.3 Universal Command Manager Customization

Configuration

Universal Command Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is a sequential file that is allocated to ddname **UNVCONF** in the batch job's JCL. The default configuration file is member **UCMCFG00** of the **UNVCONF** library.

See the Universal Command 4.1.0 User Guide for details on configuring Universal Command Manager.

Product License

Universal Command Manager must be configured with a Universal Command product license before it is used. It will not run without a valid license.

The product license contains the license parameter keywords and their values, which must be entered into the Universal Command Manager configuration file. The values are specified in the same syntax as all other configuration options.

Product licenses are provided by your Stonebranch, Inc. account representative.

JCL Procedure

A JCL procedure is provided in member **UCMDPRC** in library **SUNVSAMP**. Using the procedure simplifies future product upgrades and reduces the amount of JCL statements required in the job JCL.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See Section [2.11 Universal Products for z/OS – Time Zone Environment Variable](#) for more information the TZ environment variable.

The JCL procedure can be made available for use by either:

- Copying it to a JES procedure library, such as **SYS1 . PROCLIB**
- Using the **JCLLIB** JCL statement in the job JCL to include the **SUNVSAMP** library in the procedure library search order. For example:

```
//jobname JOB . . .  
//          JCLLIB ORDER=UNV.SUNVSAMP  
//STEP1    EXEC UCMDPRC
```

2.10.4 Universal Command Server Customization

Configuration

There are two files used in defining the Universal Command Server configuration:

- Server's component definition file
- Server's configuration file

The execution of the Server is managed by Universal Broker. The component definition file defines the Server attributes to the Broker. The Broker uses the component definition file to manage the execution of the server. The component definition file is a member of the **UNVCOMP** library. The library is allocated to the **UNVCOMP** ddname of the Universal Broker started task. The member name is **UCSCMP00**.

The Universal Command Server configuration file defines system-wide customizations for Server features and resources. The configuration file is a member of the **UNVCONF** library. The library is allocated to the **UNVCONF** ddname of the Universal Broker started task. The Broker passes the data set name to the Server when it is started by the Broker. The Server component definition file defines which member of the **UNVCONF** library to use as the configuration file. The default member name is **UCSCFG00**.

See the Universal Command 4.1.0 User Guide for details on configuring Universal Command Server.

2.10.5 Universal Connector Customization

Configuration

Universal Connector for z/OS uses a configuration file for system-wide customizations. The configuration file is a sequential file that is allocated to ddname **UNVCONF** in the batch job's JCL. The default configuration file is member **USPCFG00** of the **UNVCONF** library.

SAP RFC Configuration

Universal Connector utilizes SAP's RFC interface. The RFC interface must be configured to meet your local SAP environment. RFC configuration is defined in configuration member **USPRFC00** in the **UNVCONF** library. The member is provided by SAP and contains complete documentation on its configuration.

Product License

Universal Connector must be configured with a Universal Connector product license before it is used. It will not run without a valid license.

The product license contains the license parameter keywords and their values, which must be entered into the Universal Connector configuration file. The values are specified in the same syntax as all other configuration options.

Product licenses are provided by your Stonebranch, Inc. account representative.

JCL Procedure

A JCL procedure is provided in member **USPPRC** in library **SUNVSAMP**. Using the procedure simplifies future product upgrades and reduces the amount of JCL statements required in the job JCL.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See [Section 2.11 Universal Products for z/OS – Time Zone Environment Variable](#) for more information the TZ environment variable.

The JCL procedure can be made available for use by either:

- Copying it to a JES procedure library, such as **SYS1.PROCLIB**
- Using the **JCLLIB** JCL statement in the job JCL to include the **SUNVSAMP** library in the procedure library search order.

For example:

```
//jobname JOB . . .  
//          JCLLIB ORDER=UNV.SUNVSAMP  
//STEP1    EXEC USPPRC
```

2.10.6 Universal Control Manager Customization

Configuration

Universal Control Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is a sequential file that is allocated to ddname **UNVCONF** in the batch job's JCL. The default configuration file is member **UCTCFG00** of the **UNVCONF** library.

See the Universal Products Utilities 4.1.0 User Guide for details on configuring Universal Control Manager.

JCL Procedure

A JCL procedure is provided in member **UCTLPRC** in library **SUNVSAMP**. Using the procedure simplifies future product upgrades and reduces the amount of JCL statements required in the job JCL.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See Section [2.11 Universal Products for z/OS – Time Zone Environment Variable](#) for more information the TZ environment variable.

The JCL procedure can be made available for use by either:

- Copying it to a JES procedure library, such as **SYS1 . PROCLIB**
- Using the **JCLLIB** JCL statement in the job JCL to include the **SUNVSAMP** library in the procedure library search order.

For example:

```
//jobname JOB . . .  
//          JCLLIB ORDER=UNV.SUNVSAMP  
//STEP1    EXEC UCTLPRC
```

2.10.7 Universal Control Server Customization

Configuration

There are two files used in defining the Universal Control Server configuration:

- Server's component definition file.
- Server's configuration file.

The execution of the Server is managed by Universal Broker. The component definition file defines the Server attributes to the Broker. The Broker uses the component definition file to manage the execution of the server. The component definition file is a member of the **UNVCOMP** library. The library is allocated to the **UNVCOMP** ddname of the Universal Broker started task. The member name is **UTSCMP00**.

The Universal Control Server configuration file defines system-wide customizations for Server features and resources. The configuration file is a member of the **UNVCONF** library. The library is allocated to the **UNVCONF** ddname of the Universal Broker started task. The Broker passes the data set name to the Server when it is started by the Broker. The Server component definition file defines which member of the **UNVCONF** library to use as the configuration file. The default member name is **UTSCFG00**.

See the Universal Products Utilities 4.1.0 User Guide for details on configuring Universal Control Server.

2.10.8 Universal Data Mover Manager Customization

Configuration

Universal Data Mover Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is a sequential file that is allocated to ddname **UNVCONF** in the batch job's JCL. The default configuration file is member **UDMCFG00** of the **UNVCONF** library.

See the Universal Data Mover 4.1.0 User Guide for details on configuring Universal Data Mover.

Product License

Universal Data Mover Manager must be configured with a Universal Data Mover product license before it is used. It will not run without a valid license.

The product license contains the license parameter keywords and their values, which must be entered into the Universal Data Mover Manager configuration file. The values are specified in the same syntax as all other configuration options.

Product licenses are provided by your Stonebranch, Inc. account representative.

JCL Procedure

A JCL procedure is provided in member **UDMPRC** in library **SUNVSAMP**. Using the procedure simplifies future product upgrades and reduces the amount of JCL statements required in the job JCL.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See Section [2.11 Universal Products for z/OS – Time Zone Environment Variable](#) for more information the TZ environment variable.

The JCL procedure can be made available for use by either:

- Copying it to a JES procedure library, such as **SYS1.PROCLIB**
- Using the **JCLLIB** JCL statement in the job JCL to include the **SUNVSAMP** library in the procedure library search order.

For example:

```
//jobname JOB . . .
//          JCLLIB ORDER=UNV.SUNVSAMP
//STEP1    EXEC UDMPRC
```

2.10.9 Universal Data Mover Server Customization

Configuration

There are two files used in defining the Universal Data Mover Server configuration:

- Server's component definition file
- Server's configuration file

The execution of the Server is managed by Universal Broker. The component definition file defines the Server attributes to the Broker. The Broker uses the component definition file to manage the execution of the server. The component definition file is a member of the **UNVCOMP** library. The library is allocated to the **UNVCOMP** ddname of the Universal Broker started task. The member name is **UDSCMP00**.

The Universal Data Mover Server configuration file defines system-wide customizations for Server features and resources. The configuration file is a member of the **UNVCONF** library. The library is allocated to the **UNVCONF** ddname of the Universal Broker started task. The Broker passes the data set name to the Server when it is started by the Broker. The Server component definition file defines which member of the **UNVCONF** library to use as the configuration file. The default member name is **UDSCFG00**.

See the Universal Data Mover 4.1.0 User Guide for details on configuring Universal Data Mover Server.

2.10.10 Universal Enterprise Controller Customization

Configuration

Universal Enterprise Controller for z/OS uses a configuration file for its customizations. The configuration file is a sequential file that is allocated to ddname **UNVCONF** in the started procedure's JCL. The default configuration file is member **UECCFG00** of the **UNVCONF** library.

Product License

Universal Enterprise Controller must be configured with a UEC product license before it is used. It will not run without a valid license.

The product license contains the license parameter keywords and their values, which must be entered into the UEC configuration file. The values are specified in the same syntax as all other configuration options.

Product licenses are provided by your Stonebranch, Inc. account representative.

JCL Procedure

A JCL procedure is provided in member **UECTLR** in library **SUNVSAMP**.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See [Section 2.11 Universal Products for z/OS – Time Zone Environment Variable](#) for more information the TZ environment variable.

2.10.11 Universal Event Monitor Manager Customization

Configuration

Universal Event Monitor Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is a sequential file that is allocated to ddname **UNVCONF** in the batch job's JCL. The default configuration file is member **UEMCFG00** of the **UNVCONF** library.

See the Universal Event Monitor 4.1.0 User Guide for details on configuring Universal Event Monitor Manager.

JCL Procedure

A JCL procedure is provided in member **UEMPRC** in library **SUNVSAMP**. Using the procedure simplifies future product upgrades and reduces the amount of JCL statements required in the job JCL.

Edit the JCL procedure to meet local JCL installation requirements. The TZ environment variable should be modified to meet your local time zone information. See Section [2.11 Universal Products for z/OS – Time Zone Environment Variable](#) for more information the TZ environment variable.

The JCL procedure can be made available for use by either:

- Copying it to a JES procedure library, such as **SYS1 . PROCLIB**
- Using the **JCLLIB** JCL statement in the job JCL to include the **SUNVSAMP** library in the procedure library search order.

For example:

```
//jobname JOB . . .  
//          JCLLIB ORDER=UNV.SUNVSAMP  
//STEP1    EXEC UEMPRC
```

2.10.12 Universal Query Customization

Configuration

Universal Query for z/OS uses a configuration file for system-wide customizations. The configuration file is a sequential file that is allocated to ddname **UNVCONF** in the batch job's JCL. The default configuration file is member **UQRCFG00** of the **UNVCONF** library.

See the Universal Products Utilities 4.1.0 Reference Guide for details on Universal Query configuration options.

2.11 Universal Products for z/OS – Time Zone Environment Variable

Universal Products programs execute in a z/OS UNIX environment, also known as a POSIX(ON) Language Environment.

The z/OS UNIX environment assumes that the z/OS system time is GMT or UTC format. It then uses the TZ environment variable value to determine the adjustments required for local time.

The TZ environment variable is an IBM Language Environment (LE) environment variable with a value set to the appropriate time zone and offset information so that time values are properly processed. TZ allows you to set the standard and daylight savings (or summer time) time zones and the offset from the local time zone to the UTC time.

LE environment variables, such as TZ, are set with the LE option ENVAR using the JCL step PARM keyword. For example, the following statement sets the standard time zone to Eastern Standard Time (EST) with an offset of 5 hours and the daylight saving time zone to Eastern Daylight Saving Time (EDT).

```
//PS1      EXEC PGM=UCMD, PARM='ENVAR (TZ=EST5EDT) /&UPARM'
```

Note: The forward slash (/) character separates the LE runtime parameters from the program parameters.

North American Values

TZ environment variables values for North America are listed below.

- EST5EDT Eastern Standard Time, Eastern Daylight Time
- CST6CDT Central Standard Time, Central Daylight Time
- MST7MDT Mountain Standard Time, Mountain Daylight Time
- PST8PDT Pacific Standard Time, Pacific Daylight Time
- AKST9AKDT Alaska Standard Time, Alaska Daylight Time

European Values

TZ environment variables values for Europe are listed below.

- GMT0BST Greenwich Mean Time, British Summer Time
- WET0WEST Western Europe Time, Western Europe Summer Time
- CET-1CEST Central Europe Time, Central Europe Summer Time
- EET-2EEST Eastern Europe Time, Eastern Europe Summer Time

Other common time zone abbreviations may be used. IBM does not document all possible values.

2.12 Universal Products for z/OS – TCP/IP Configuration

This section describes optional Universal Products and TCP/IP configuration topics. Whether these steps are required or desirable depends on your local TCP/IP configuration.

2.12.1 Specifying TCP/IP Affinity

Universal Product programs are considered generic client and server programs in IBM's TCP/IP terminology. They do not have an affinity for a specific transport provider (TCP/IP started task). The programs will utilize the appropriate transport provider based on TCP/IP configuration.

An affinity to a specific transport provider can be established for any Universal Product program using Language Environment variable `_BPXK_SETIBMOPT_TRANSPORT`. The variable value specifies the TCP/IP started task name the program must use as its transport provider.

The JCL illustration below establishes affinity to TCP/IP started task TCPIPA:

```
//PS1 EXEC PGM=UCMD,  
// PARM='ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIPA") /&UPARM'
```

2.12.2 TCP/IP Resource Protection

The IBM TCP/IP product (Communications Server) offers optional protection to TCP/IP resources using SAF interfaces. The protection is implemented with a set of resource profiles defined in the SERVAUTH class.

If you are using the TCP/IP resource protection, you must permit appropriate privileges to the user profiles with which Universal Products execute. Refer to the IBM Communications Server: IP Configuration Guide for complete details on TCP/IP resource protection. The TCP/IP resource profiles and the Universal Products required access are discussed in the following sections.

Stack Access Control

The SAF resource profile `EZB.STACKACCESS.sysname.tcpname` in the SERVAUTH class controls which user profiles have access to a TCP/IP stack. All user profiles with which Universal Products execute require READ access to the appropriate TCP/IP stack access profile.

Port Access Control

The SAF resource profile `EZB.PORTACCESS.sysname.tcpname.SAF` keyword in the SERVAUTH class controls access to specific non-ephemeral port ranges.

The Universal Broker binds to a service port (defaults to 7887). The user profile with which the Universal Broker started task executes requires READ access to any resource that protects this port.

The Universal Enterprise Controller binds to a service port (defaults to 8778). The user profile with which the Universal Enterprise Controller started task executes requires READ access to any resource that protects this port.

Network Access Control

The SAF resource profile `EZB.NETACCESS.sysname.tcpname.zonename` in the SERVAUTH class controls access to security zones. A security zone defines networks and hosts by IP address. All user profiles with which Universal Products execute require READ access to the appropriate TCP/IP security zones profile.

Socket Option Access Control

The SAF resource profile `EZB.SOCKOPT.sysname.tcpname.SO_BROADCAST` in the SERVAUTH class controls access to the socket `SO_BROADCAST` option.

No Universal Products programs use the `SO_BROADCAST` socket option, so no user profiles require access.

2.13 Universal Products for z/OS – Configuration of z/OS System SSL

Universal Products can use the IBM z/OS System SSL library or the OpenSSL SSL library for its SSL network communications. The SSL library selection is made with Universal Products' `SSL_IMPLEMENTATION` configuration option.

z/OS System SSL requires the IBM System SSL Cryptographic Services base element. In addition, Universal Products requires Cryptographic Services Security Level 3 element, which includes the cryptographically strong SSL cipher suites.

System SSL provides the following benefits:

- Utilizes any cryptographic hardware features available reducing the amount CPU resources used by Universal Products.
- Seamless integration with RACF certificate management features.

Note: If RACF digital certificates are new to you or your site, refer to the following documentation for complete details:

- z/OS Security Server RACF Security Administrator's Guide
- z/OS Security Server RACF Command Language Reference

In order for Universal Products to use z/OS System SSL, the following conditions must be met:

1. Universal Products supports z/OS System SSL on z/OS 1.4 and above.
2. Universal Product `SSL_IMPLEMENTATION` configuration value must be set to `SYSTEM`.
3. User profiles with which the Universal Products component executes must have `READ` access to the RACF profile `IRR.DIGTCERT.LISTRING` in the `FACILITY` class.
4. User profiles with which the Universal Products component executes must have a certificate key ring associated with them that includes the user's certificate and the CA's certificate.

2.13.1 Integrated Cryptographic Service Facility (ICSF)

z/OS System SSL will use ICSF when available. The ICSF started task must be running and ICSF configuration completed.

The user profile with which the System SSL application executes must have access to the following ICSF resources in the CSFSERV class:

- CSFCKI, clear key import
- CSFCKM, multiple clear key import
- CSFDEC, symmetric key decrypt
- CSFDSG, digital signature generate
- CSFDSV, digital signature verify
- CSFENC, symmetric key encrypt
- CSFPKB, PKA key build
- CSFPKD, PKA decrypt
- CSFPKE, PKE encrypt
- CSFPKI, PKA key import

Refer to the IBM z/OS ICSF Administrator's Guide for managing access to ICSF resources.

2.13.2 Universal Broker Digital Certificate (RACF) Set-up

Setting up a digital certificate infrastructure in a production environment for the first time requires careful planning by the Security organization and Universal Products administrator. The instructions provided in this section have been simplified for illustration purposes.

You work with RACF Digital Certificates using the RACF command **RACDCERT**. RACF profiles control access to the functions provided by **RACDCERT**. The user profile with which the following commands are executed require either:

- SPECIAL attribute
- Appropriate access to the **IRR.DIGTCERT.function** profile in the FACILITY class.
 - READ access is required to **IRR.DIGTCERT.function** to issue **RACDCERT** commands for the executing user.
 - UPDATE access is required to **IRR.DIGTCERT.function** to issue **RACDCERT** commands for other users.
 - CONTROL access is required to **IRR.DIGTCERT.function** to issue **RACDCERT** command for SITE or CERTAUTH certificates.

The member **UNVINDC** in the **INSTALL** library contains the JCL to execute the RACF commands listed in the following steps.

1. Create a Certificate Authority (CA) certificate and private key using the following **RACDCERT** command:

```
RACDCERT CERTAUTH GENCERT +
  SUBJECTSDN(CN('Certificate Authority') +
    OU('Security') +
    O('Company Name, Inc.') +
    C('US')) +
  NOTAFTER(DATE(2030-01-01)) +
  KEYUSAGE(HANDSHAKE CERTSIGN) +
  WITHLABEL('Company CA')
```

Change the subject and label names to meet local requirements.

2. Create a certificate for the Universal Broker STC and sign it with the CA certificate created in Step 1 using the following **RACDCERT** command:

```
RACDCERT ID(UBRUSR) GENCERT +
  SUBJECTSDN(CN('broker.company.com') +
    OU('Operations') +
    O('Company Name, Inc.') +
    C('US')) +
  KEYUSAGE(HANDSHAKE) +
  WITHLABEL('Broker') +
  SIGNWITH(CERTAUTH LABEL('Company CA'))
```

Change the subject and label names to meet local requirements. The subject's Common Name (CN) value should uniquely identify this instance of the broker in the enterprise.

3. Create a certificate key ring for the user profile UBRUSR with the following RACDCERT command:

```
RACDCERT ID(UBRUSR) ADDRING(BROKER)
```

4. Connect the CA certificate and the Universal Broker certificate to the key ring with the following RACDCERT command:

```
RACDCERT ID(UBRUSR) CONNECT(CERTAUTH LABEL('Company CA') +
RING(BROKER)
RACDCERT ID(UBRUSR) CONNECT(LABEL('Broker') RING(BROKER) DEFAULT)
```

Change the labels to match the values used in previous steps.

5. If the resource profile `IRR.DIGTCERT.LISTRING` in the FACILITY class is not defined, define it with the following RDEFINE command:

```
RDEFINE FACILITY (IRR.DIGTCERT.LISTRING) UACC(NONE)
```

6. Permit the Broker user profile UBRUSR READ access to the RACF profile `IRR.DIGTCERT.LISTRING` in the FACILITY class using the following PERMIT command:

```
PE IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(UBRUSR) ACCESS(READ)
```

7. Modify the Universal Broker configuration member `UBRCFG00` as follows:

```
ssl_implementation    system
saf_key_ring          BROKER
```

8. The CA certificate must be distributed to the remote systems from which Universal Products managers are executed. The managers must be configured with the CA certificate in their list of Trusted CA certificates using the `CA_CERTIFICATES` configuration option.

The CA certificate is exported out of the RACF data base into a data set in a PEM (or base64) format with the following RACDCERT command:

```
RACDCERT CERTAUTH EXPORT (LABEL('Company CA')) +
DSN(TEST.CA.CERT) FORMAT(CERTB64)
```

Change the label to match the value used in previous steps.

The `tsoprefix.TEST.CA.CERT` data set contains a PEM formatted certificate. The format is a text format that transfers safely across the network in text mode.

Note that the CA private key is not exported. The CA certificate does not contain any private data.

2.14 Universal Products for z/OS – Configuration of Universal Products Security

Some Universal Products utilize z/OS security services to control access to product functions and resources. This section describes the installation steps to activate product security and define resource controls.

The use of the security resource is optional. If you do not want to use it at this time, this section can be skipped.

The following sections specify the configuration methods for IBM's RACF and Computer Associate's ACF2 security products.

2.14.1 RACF Class

All products use the Universal Products RACF class **\$UNV**. The class name can be changed if local requirements require it.

To install the Universal Products RACF class, perform the following steps:

1. Modify your local RACF class descriptor table by adding the following entry to ICHRRRCDE:

```

ICHERCDE CLASS=$UNV,          +
      CASE=ASIS,              +
      FIRST=ANY,              +
      OTHER=ANY,              +
      MAXLNTH=246,           +
      DFTUACC=NONE,          +
      ID=128,                 +
      POSIT=128

```

2. Modify the ID and POSIT values to meet local requirements.

Support for the CASE parameter was added in z/OS 1.2. Case-sensitive profile names are necessary for USS path name support in the standard I/O access profiles. See Section [Universal Command Standard I/O Access Profile](#) for details.

Care must be taken when defining case-sensitive profile names so that the profile characters are typed with the desired case. RACF will not automatically upper case the profile names. If the wrong case is used, the desired profile will not be found. If the standard I/O profiles are not necessary or USS path name support is not necessary, the CASE parameter can be removed.

Sample assemble and link edit JCL is provided in member **UCMRRRCDE** in the **INSTALL** library.

IBM provides a sample ICHRRRCDE table and the JCL to assemble and link edit it in **SYS1.PARMLIB(RACTABLE)**.

3. Modify your local RACF routing table by adding the following entry to ICHRFRTB.

```
ICHRFRTB CLASS=$UNV,      +  
        ACTION=RACF
```

Sample assemble and link edit JCL is provided in member **UCMRFR01** in the **INSTALL** library.

IBM provides a sample ICHRRRCDE table and the JCL to assemble and link edit it in **SYS1.PARMLIB(RACTABLE)**.

4. If you changed the default RACF class name **\$UNV** to another name, the UCMD, UCTL, and UEM load modules must be modified to include the same class name.

(If you did not change the default class name, skip this step.)

- a. Member **UCMCMVS** in the **INSTALL** library contains an SMP/E USERMOD to apply the modification to the UCMD load module. Change the USERMOD to the new class name and submit the job.
- b. Member **UCTCMVS** in the **INSTALL** library contains an SMP/E USERMOD to apply the modification to the UCTL load module. Change the USERMOD to the new class name and submit the job.
- c. Member **UEMCMVS** in the **INSTALL** library contains an SMP/E USERMOD to apply the modification to the UEM load module. Change the USERMOD to the new class name and submit the job.

All steps must end with return code 0.

5. IPL the z/OS system to install the new class descriptor and routing tables.
6. Activate the new class and define it for generic profile checking by entering the following RACF SETROPTS commands:

```
SETROPTS CLASSACT($UNV)  
SETROPTS GENERIC($UNV)
```

2.14.2 ACF2 CLASS

All products use the Universal Products ACF2 class **\$UNV**. The class name may be changed if local requirements require it.

To install the Universal Products ACF2 class, perform the following steps:

1. Create a CLASMAP with the following ACF2 command:

```
SET CONTROL(GSO)
INSERT CLASSMAP.$UNV RESOURCE($UNV) RSRCTYPE(UNV) ENTITYLN(246)
```

2. Rebuild and refresh the directories with the following console command:

```
F ACF2, REFRESH(CLASMAP), SYSID(sysid), CLASS(C), TYPE(GSO)
```

3. Define the following resource rules for Universal Command and Universal Control with the following commands:

```
SET RESOURCE(OSM)
COMPILE * STORE
$KEY(UCMD) TYPE(UNV)
UID(ucmd_admin_id) ALLOW
- UID(ucmd_admin_id) SERVICE(READ) ALLOW
```

```
SET RESOURCE(OSM)
COMPILE * STORE
$KEY(UCTL) TYPE(UNV)
UID(uctl_admin_id) ALLOW
- UID(uctl_admin_id) SERVICE(READ) ALLOW
```

2.14.3 Universal Command Security

Universal Command access controls consist of the following profile types:

- Remote Access profiles control Universal Command's access to remote systems.
- Stdio Access profiles control Universal Command's access to local data sets used as standard input, output or error.

Universal Command Remote Access Profile

The remote access profile controls Universal Command Manager's access to specific remote systems running Universal Products. The remote system is identified by the IP address of the remote system, the port number on which the Manager is connecting to the remote system, and the remote system user ID with which the Manager is requesting the command to be executed.

Universal Command Manager identifies a remote system with the REMOTE_HOST and REMOTE_PORT configuration options, and the remote user ID with the USER_ID configuration option.

The remote access profile has the following format:

UCMD.Iipaddress.Pport.userid

Table 2.9, below, identifies the fields that comprise the profile name.

Field	Description
ipaddress	Numeric dotted-form IP address of the remote host as specified by the REMOTE_HOST option. The format of the IP address is four three-character numeric fields. Each field represents one number of the IP address. For example, IP address 256.10.2.123 is formatted as 256010002123 .
port	Numeric port number on which Universal Broker is listening as specified by the REMOTE_PORT option. The default Universal Broker port number is 7887. The format of the port number is a five-digit number. For example, port 7887 is formatted as 07887 .
userid	Remote user identifier with which Universal Command Manager will execute the remote command as specified by the USER_ID option. Whether or not a user identifier is required depends on the Universal Command Server configuration. If no user identifier is specified for the Manager, the userid value is <BLANK>. The value userid is upper case. Some remote hosts may have case-sensitive user identifiers. All user identifiers specified on the Manager are translated to upper case for building the profile.

Table 2.9 Universal Products for z/OS - Remote Access Profile Fields

Universal Command Standard I/O Access Profile

The standard I/O (**stdio**) access profile controls Universal Command Manager's access to data sets allocated for standard I/O. The standard I/O ddnames are **UNVIN** for standard input, **UNVOUT** for standard output, and **UNVERR** for standard error, unless otherwise changed with the **SIO_LOCAL_FILE** configuration option.

The standard I/O access profile has the following format:

`UCMD.stdio.Ipaddress.Pport.allocation`

Table 2.10, below, identifies the fields that comprise the profile name.

Field	Description
stdio	Standard I/O file which the profile is for. Valid values are: <ul style="list-style-type: none"> • STDIN for the standard input file. • STDOUT for the standard output file. • STDERR for the standard error file.
ipaddress	Numeric dotted-form IP address of the remote host as specified by the REMOTE_HOST option. The format of the IP address is four three-character numeric fields. Each field represents one number of the IP address. For example, IP address 256.10.2.123 is formatted as 256010002123 .
port	Numeric port number on which Universal Broker is listening as specified by the REMOTE_PORT option. The default Universal Broker port number is 7887. The format of the port number is a five-digit number. For example, port 7887 is formatted as 07887 .
allocation	Entity that is allocated to the standard I/O ddname. There are a number of different items that may be allocated to a ddname. The following formats are available: A data set allocation is represented by Ddsn , where dsn is the data set name. For example, a ddname allocation of PROD.APPL.DATA is formatted as DPROD.APPL.DATA . If allocating a PDS and a member name is specified, do not include the member name in the profile name, only the PDS name. If allocating a relative generation data set, do not include the relative number in the profile name, only the GDG name. A JES SYSIN, SYSOUT or SUBSYS= allocation is represented by the value SUBSYS. A temporary data set allocation is represented by the value TEMPORARY. A DUMMY or NULLFILE allocation is represented by the value NULLFILE. A USS path name allocation is represented by Upath , where path is the USS path name. For example, a ddname allocation of /prod/app1/data is formatted as U/prod/app1/data . USS path name support starts at z/OS 1.4. Prior to z/OS 1.4, the path name is not available to format the profile name. The value used for path in this case is UNKNOWNUSSPATHNAME . USS path names are case sensitive. The Universal Products class must be defined with the CASE=ASIS parameter to support case sensitive profile names. The maximum length of the profile name is 246 characters. The maximum length of a USS path name allocated to a ddname is 256 characters. It is possible the path name may be truncated in formatting the profile name. Truncation does not result in an error condition.

Table 2.10 Universal Products for z/OS - Standard I/O Access Profile Fields

Universal Command Security Profile Definitions

This section provides sample RACF commands that may be used to define Universal Command security profiles and permit z/OS user identifiers access to those profiles. Refer to the IBM RACF documentation for complete details on RACF commands.

Example 1

Assume that you want to restrict Universal Command Manager for z/OS access to remote host **10.23.90.2**. The following profile would restrict access to only those z/OS users who have read access to the profile **UCMD.I010023090002.*.***.

The following TSO commands define the required profile and permits access to TSO user TSO555.

```
RDEF $UNV (UCMD.I010023090002.*.*) UACC(NONE)
PE UCMD.I010023090002.*.* CLASS($UNV) ID(TSO555) ACCESS(READ)
```

Example 2

Assume that you run all Universal Brokers on privileged port **1000**. To enforce the policy that z/OS Managers connect only to port 1000, define profile **UCMD.*.*** with universal access none and define **UCMD.*.P01000.*** with universal access read.

The following TSO commands define the required profiles.

```
RDEF $UNV (UCMD.*.*.*) UACC(NONE)
RDEF $UNV (UCMD.*.P01000.*) UACC(READ)
```

Example 3

Assume that you want to restrict root access to all hosts from Universal Command Manager for z/OS. The following profile would restrict root access to only those z/OS users who have read access to the profile **UCMD.*.*.ROOT**.

The following TSO command defines the required RACF profile.

```
RDEF $UNV (UCMD.*.*.ROOT) UACC(NONE)
```

Example 4

Assume that you want to restrict Universal Command Manager access to data sets **PROD.***. You also don't want Universal Command Manager to use any temporary data sets. The following profiles would restrict access to only those z/OS users who have read access to the profile.

The following TSO command defines the required RACF profile.

```
RDEF $UNV (UCMD.STD*.*.*.DPROD.***) UACC(NONE)
RDEF $UNV (UCMD.STD*.*.*.TEMPORARY) UACC(NONE)
```

2.14.4 Universal Control Security

Universal Control access controls consist of the following profile types:

- Remote Access profiles control Universal Control's access to remote systems.

Universal Control Remote Access Profile Format

The remote access profile controls Universal Controls Manager's access to specific remote systems running Universal Products. The remote system is identified by the IP address of the remote system, the port number on which the Manager is connecting to the remote system, and the command to be executed.

Universal Control Manager identifies a remote system with the REMOTE_HOST and REMOTE_PORT configuration options, and the command as one of the command options.

The remote access profile has the following format:

UCTL . I ipaddress . P port . command

The profile name is composed of the following fields:

Field	Description
ipaddress	Numeric dotted-form IP address of the remote host as identified by the REMOTE_HOST option. The format of the IP address is four three-character numeric fields. Each field represents one number of the IP address. For example, IP address 256 . 10 . 2 . 123 is formatted as 256010002123 .
port	Numeric port number on which Universal Broker is listening as identified by the REMOTE_PORT option. The default Universal Broker port number is 7887. The format of the port number is a five-digit number. For example, port 7887 is formatted as 07887 .
command	Universal Control command that the Manager is requesting execution. Possible command values are START, STOP, and REFRESH.

Table 2.11 Universal Control for z/OS - Remote Access Profile Fields

Universal Control Security Profile Definition

This section provides sample RACF commands that may be used to define Universal Control security profiles and permit z/OS user identifiers access to those profiles. Refer to the IBM RACF documentation for complete details on RACF commands.

Example 1

Assume you wish to restrict Universal Control Manager for z/OS access to remote host **10.23.90.2**. The following profile would restrict access to only those z/OS users who have read access to the profile **UCTL.I010023090002.*.***.

The following TSO commands define the required profile and permits access to TSO user TSO555.

```
RDEF $UNV (UCTL.I010023090002.*.*) UACC(NONE)
PE UCTL.I010023090002.*.* CLASS($UNV) ID(TSO555) ACCESS(READ)
```

2.14.5 Universal Event Monitor Security

Universal Event Monitor access controls consist of the following profile types:

- Remote Access profiles control Universal Event Monitor's access to remote systems.

Universal Event Monitor Remote Access Profile Format

The remote access profile controls Universal Event Monitor Manager's access to specific remote systems running Universal Products. The remote system is identified by three elements:

1. System's IP address
2. Port number that the Manager uses to connects to a Universal Broker executing on the system
3. User account specified from the UEM Manager, which is defined on the remote system

The Universal Event Monitor Manager identifies a remote system with the REMOTE_HOST and REMOTE_PORT configuration options, and the remote user account with the USER_ID configuration option.

The remote access profile has the following format:

`UEM.Iipaddress.Pport.userid`

The profile name is composed of the following fields:

Field	Description
Ipaddress	IP address of the remote host, in dotted-decimal notation, as identified by the REMOTE_HOST configuration option. The format of the IP address is four three-character numeric fields. Each field represents one number of the IP address. For example, IP address 256 . 10 . 2 . 123 is formatted as 256010002123 .
Port	Numeric port number on which Universal Broker is listening as identified by the REMOTE_PORT configuration option. The default Universal Broker port number is 7887. The format of the port number is a five-digit number. For example, port 7887 is formatted as 07887 .
Userid	ID of a remote user account, which was specified by the Manager with the USER_ID option. If the Universal Event Monitor Server is configured to not require a user ID, the value for this field is <BLANK>. The value <i>userid</i> is upper case. Some remote hosts may have case-sensitive user identifiers. All user identifiers specified on the Manager are translated to upper case for building the profile.

Table 2.12 Universal Event Monitor for z/OS - Remote Access Profile Fields

Universal Event Monitor Security Profile Definition

This section provides sample RACF commands that may be used to define Universal Event Monitor security profiles and permit local user accounts access to those profiles. Refer to the IBM RACF documentation for complete details on RACF commands.

Example 1

Assume you wish to restrict Universal Event Monitor Manager for z/OS access to remote host **10.23.90.2**. The following profile would restrict access to only those z/OS users who have read access to the profile **UEM.I010023090002.*.***.

The following TSO commands define the required profile and permits access to TSO user TSO555.

```
RDEF $UNV (UEM.I010023090002.*.*) UACC(NONE)PE UEM.I010023090002.*.*  
CLASS($UNV) ID(TSO555) ACCESS(READ)
```

2.15 Universal Products for z/OS - Performance Guidelines

Universal Products consists of product components distributed throughout the enterprise communicating with each other over the computer network using the TCP/IP communication protocol.

Universal Products offers reliable, fault tolerant, secure, and efficient communications between its distributed components. In order for product components to effectively meet their communication requirements, z/OS must provide sufficient execution time for the product components.

The execution of the communication protocol is a real-time activity and communication time-outs may be exceeded if product components are not dispatched appropriately while executing the communication protocol.

The following sections provide performance guidelines for z/OS Universal Product components.

2.15.1 UNIX System Services and Language Environment

All Universal Products components are written in C/C++ and utilize z/OS Language Environment (LE) and z/OS UNIX System Services (USS).

The IBM z/OS UNIX System Services Planning manual includes a "Tuning performance" chapter that should be reviewed to improve USS performance in general.

Universal Products components do not attempt to use the USS `setpriority`, `chpriority`, or `nice` functions to adjust their performance group or service class.

2.15.2 Universal Products Managers

Universal Products Managers consist of Universal Command, Universal Data Mover and Universal Event Monitor managers. They typically execute in the JES subsystem as a batch job or the OMVS subsystem as a USS shell command.

The managers communicate with remote Universal Brokers and their corresponding Universal Product Server components on remote systems using the TCP/IP protocol.

In cases where the z/OS workload requires more resources than are available, z/OS will favor workload with a higher dispatch priority over workload with a lower dispatch priority. If a Universal Products manager is being executed with a lower dispatch priority than other workload competing for resources, it may not be given sufficient execution time to meet its network communication timing requirements. The result will be false network time-out errors in the Universal Products manager.

The effect of a network time-out condition depends on whether or not the Universal Products manager is using the Network Fault Tolerant (NFT) feature. If NFT is used, the manager reestablishes the communication session and continues; otherwise, the manager ends with an error.

False network communication time-out errors can be addressed using one or more of the following options:

1. Increase the NETWORK_DELAY configuration option value (default is 120 seconds). NETWORK_DELAY specifies the maximum amount of time to wait for data on a communication session before considering the session timed out. Increasing the value allows for the manager batch job to be swapped out for a longer period of time before the session will be considered timed out.

However, in cases where a condition truly exists in the network that would result in a true network time-out, a larger NETWORK_DELAY value would result in a longer period of time before the manager would detect and respond to the time-out condition.

2. Raise the Universal Products manager workload dispatch priority by placing it in a higher performance group or service class. Raising the workload dispatch priority will allow z/OS to provide sufficient CPU resources to the manager to meet network timing requirements.

2.15.3 Universal Broker and Universal Products Servers

The Universal Broker started task is the center of activity on each system on which Universal Products is installed. Almost all components communicate with a locally installed Universal Broker during their execution, including managers and servers.

Universal Broker is responsible for managing Universal Products servers that are performing work on z/OS on behalf of remote Universal Products managers. Universal Product servers are created by the Universal Broker using the USS spawn function. The servers run as USS child processes of the Universal Broker started task in the OMVS subsystem.

The Universal Broker started task must execute with a sufficiently high performance group or service class in order to service all manager and server requests in a timely manner to avoid false network time-out conditions.

On heavily loaded systems, it is recommended to make the Universal Broker started task non-swappable to help overall improvement of Universal Broker.

2.15.4 Universal Enterprise Controller

The Universal Enterprise Controller (UEC) started task performs real-time monitoring of Agents distributed throughout the network. UEC communicates with each Agent on a defined polling interval.

UEC is a USS, multi-threaded application written in C/C++ that heavily utilizes TCP / IP. The amount of work that UEC performs depends directly on the number of Agents defined to it. UEC maintains Agent status information and Universal Event Subsystem information in the UEC database. The database is an HFS database that is mounted in the HFS file system.

UEC must execute with a sufficiently high performance group or service class in order to perform its Agent monitoring service effectively. False Agent time-out alerts can result if UEC is not dispatched in a timely manner.

On heavily loaded systems, it is recommended to make the UEC started task non-swappable to help overall performance of UEC.

2.16 Universal Products for z/OS – Data Set Inventory

As part of the Universal Products for z/OS package installation, two types of data sets are allocated and cataloged:

- SMP/E data sets
- Non-SMP/E data sets

2.16.1 SMP/E Data Sets

[Table 2.13](#), below, lists the SMP/E data sets — and their space requirements — that are allocated and cataloged as part of the Universal Products for z/OS package installation.

Depending on your installation choices, the data set high-level qualifiers may be different.

Data Set Name	Space (tracks)	Description
UNV . AUNVLOAD	5000	SMP/E distribution library for the product load library.
UNV . AUNVNLS	30	SMP/E distribution library for the product national language support library.
UNV . AUNVSAMP	30	SMP/E distribution library for the product sample library.
UNV . AUNVTMPL	15	SMP/E distribution library for configuration template files
UNV . SUNVLOAD	5000	SMP/E target library for the product load library.
UNV . SUNVNLS	30	SMP/E target library for product national language support library.
UNV . SUNVSAMP	30	SMP/E target library for the product sample library.
UNV . SUNVTMPL	15	SMP/E target library for configuration template file.
UNV . SMP . CSI		SMP/E CSI VSAM cluster for Universal Products SMP/E zones.
UNV . SMP . CSI . DATA	75	SMP/E CSI VSAM data for Universal Products SMP/E zones.
UNV . SMP . CSI . INDEX	15	SMP/E CSI VSAM index for Universal Products SMP/E zones.
UNV . SMP . SMPLOG	30	SMP/E log file.
UNV . SMP . SMPLOGA	30	SMP/E backup log file.
UNV . SMP . SMP LTS	5000	SMP/E target library for base versions of load modules using a SYSLIB allocation.
UNV . SMP . SMPPTS	30	SMP/E target library for macros existing only in the distribution libraries.
UNV . SMP . SMPPTS	500	SMP/E temporary SYSMOD storage.
UNV . SMP . SMPSCDS	30	SMP/E zone backup library.
UNV . SMP . SMPSTS	30	SMP/E target library for source existing only in the distribution libraries.

Table 2.13 Universal Products for z/OS - SMP/E Data Sets

2.16.2 Non-SMP/E Data Sets

Table 2.14, below, lists the non-SMP/E data sets – and their space requirements – that are allocated and cataloged as part of the Universal Products for z/OS package installation.

Depending on your installation choices, the data set high-level qualifiers may be different.

Data Set Name	Space (tracks)	Description
UNV . UECDB	4,500	Universal Enterprise Controller databases.
UNV . MDL	1	Universal Broker sequential trace data set allocation model.
UNV . UNVCREF	75	Universal Command Server command reference library.
UNV . UNVDB	150	Universal Broker HFS or zFS database.
UNV . UCRDB	15	Universal Certificate database.
UNV . UNVSPool	3000	Universal Products HFS or zFS spool database.
UNV . UNVTRACE	150	Universal Broker PDS/E trace data set.
UNV . V3R2M0 . INSTALL	45	Universal Products package installation and maintenance JCL.
UNV . UNVCONF	15	Universal Products configuration library.
UNV . UNVCOMP	15	Universal Products component definition library.

Table 2.14 Universal Products for z/OS - Non-SMP/E Data Sets

Chapter 3

z/OS USS Installation

3.1 Overview

This chapter describes the installation of Stonebranch, Inc. Universal Products for z/OS UNIX System Services (USS) on the z/OS operating system. Unless otherwise specified, all references to Universal Products for z/OS USS in this chapter refer to version 4.1.0.

The installation consists of running a number of batch jobs. The output of these batch jobs should be kept until a proper installation has been verified.

Note: All Universal Products for z/OS USS are provided in the Universal Products for z/OS SMP/E package.

Note: Starting with the 3.2.0 release of Universal Products, a Universal Broker must run on all systems on which a Universal Product component is running, including manager components. The Broker maintains product configuration data for all components that have a configuration file.

3.2 Universal Products for z/OS USS Package

Universal Products for z/OS USS contains the following Universal Products components:

- Universal Command Manager 4.1.0
- Universal Control Manager 4.1.0
- Universal Data Mover Manager 4.1.0
- Universal Encrypt 4.1.0
- Universal Event Monitor Manager 4.1.0
- Universal Message to Exit Code Translator 4.1.0
- Universal Query 4.1.0
- Universal Certificate 4.1.0
- Universal WTO 4.1.0
- Universal Copy 4.1.0

3.2.1 Product Compatibility

[Table 3.1](#), below, identifies the compatibility of Universal Products 4.1.0 for z/OS USS with previous product versions.

Product	Compatibility
Universal Command 4.1.0	Universal Command 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Control 4.1.0	Universal Control 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Data Mover 4.1.0	Universal Data Mover 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0
Universal Encrypt 4.1.0	Universal Encrypt 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Query 4.1.0	Universal Broker 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Event Monitor 4.1.0	Universal Event Monitor 3.2.0, 3.1.1 and 3.1.0.

Table 3.1 Universal Products for z/OS USS - Product Compatibility

The product references pertain to all support platforms for that version.

3.3 Universal Products for z/OS USS – Installation Requirements

Universal Products for z/OS USS require the following software releases:

- z/OS 1.4 or above.
- IBM Language Environment for z/OS 1.4 or above.
- IBM Communication Server for z/OS 1.4 or above.
- SMP/E 3.2 or above.
- Minimum 100 cylinders of DASD and 81M bytes on a z/OS UNIX file system.

The user identifier used to execute the installation jobs must meet the following requirements:

1. User ID must have a properly defined OMVS segment.
2. User ID must have READ access to the `BPX.FILEATTR.APF` profile in the `FACILITY` class.

IMPORTANT: All Universal Products 4.1.0 for z/OS USS are provided in the Universal Products for z/OS SMP/E package. Universal Products for z/OS components must be installed before the z/OS USS components. See [Chapter 2 z/OS Installation](#) for information on completing the z/OS installation.

3.3.1 Data Set Space Requirements

As part of the Universal Products for z/OS USS installation, a number of SMP/E data sets are allocated and cataloged and a number of z/OS UNIX directories are created.

The space requirements for these data sets and directories are listed in [Section 3.8 Universal Products for z/OS USS – Data Set Inventory](#).

3.3.2 SMP/E

Universal Products for z/OS USS are installed using SMP/E. The products are installed in the Universal Products SMP/E CSI.

[Table 3.2](#), below, identifies the SMP/E FMIDs for the Universal Products for z/OS USS components.

Product	FMID	SMP/E Requisites
USS Universal Common 4.1.0	UUNV410	No prerequisites. Supersedes and deletes FMID UUNV320.
USS Universal Broker Utilities 4.1.0	UUBR410	UUNV410 and TUBR410 are prerequisites. Supersedes and deletes FMID UUBR320.
USS Universal Command Manager 4.1.0	UUCM410	UUNV410 and TUBR410 are prerequisites. Supersedes and deletes FMID UUCM320.
USS Universal Data Mover Manager 4.1.0	UUDM410	UUNV410 and TUBR410 are prerequisites. Supersedes and deletes FMID UUDM320.
USS Universal Utilities 4.1.0	UUTL410	UUNV410 is a prerequisite. Supersedes and deletes FMID UUTL320.
USS Universal Event Monitor Manager 4.1.0	UUEM410	UUNV410 and TUBR410 are prerequisites. Supersedes and deletes FMID UUEM320.

Table 3.2 Universal Products for z/OS USS - SMP/E FMIDs for Universal Products Package

3.4 Universal Products for z/OS USS – Installation Upgrades

This section describes changes in the product installation that have occurred with new versions of the product. If a particular version is absent from the list, no change occurred.

3.4.1 Universal Products for z/OS USS 3.2.0

This is the first release where USS components are packaged as an SMP/E installation.

Universal Products for z/OS USS versions prior to 3.2.0 were installed with a product installation script. There is no automatic upgrade path to 3.2.0. Any previously installed version must be manually un-installed.

If the 3.2.0 version will be installed in the same directories as the previous version, the previous version must be un-installed before the 3.2.0 version is installed. If the 3.2.0 version will not be installed in the same directories, the previous version can be un-installed after the 3.2.0 version is installed.

To un-install a previous version, simply remove all z/OS UNIX file system product directories. Refer to the previous version's Installation Guide for a list of product directories.

Product configuration files have been moved from the `/etc/universal` directory to the non-SMP/E product library **UNVCONF**. The change allows for remote configuration capability of z/OS USS configuration members. Configuration members are delivered in the SMP/E **SUNVSAMP** library and copied to the appropriate library in an installation job.

3.5 Universal Products for z/OS USS – Product Distribution File

The Universal Products for z/OS USS installation files are distributed with the Universal Products for z/OS distribution file.

See [Chapter 2 z/OS Installation](#) for information on downloading the Universal Products for z/OS distribution file.

3.6 Universal Products for z/OS USS – SMP/E Installation

The installation steps describe how to perform the SMP/E install of Universal Products for z/OS USS in a step-by-step process.

Note: Installation of z/OS USS is dependent upon completion of the z/OS package installation, and must be installed in the same CSI as the z/OS package.

Two different installation processes are provided. Which installation process used depends on the installation environment.

1. Installing any Universal Product from Stonebranch, Inc. for the first time, or installing a Universal Products for z/OS USS package in a new SMP/E CSI.

See Section [3.6.1 New Install, New CSI](#) for installation instructions.

2. Upgrading a Universal Products 3.2.0 for z/OS USS package install. In this case, the Universal Products for z/OS USS 4.1.0 package is installed into an SMP/E CSI that contains a Universal Products 3.2.0 package.

See Section [3.6.2 Universal Products for z/OS USS 3.2.0 Upgrade, Existing CSI](#) for installation instructions.

3.6.1 New Install, New CSI

The New Install, New CSI installation process describes how to install the Universal Products for z/OS USS package in a newly allocated SMP/E CSI.

Use this installation process for either of these environments:

- Installing a Universal Product for z/OS USS from Stonebranch, Inc. for the first time.
- Installing a Universal Products for z/OS USS package in a different SMP/E CSI than other Universal Products from Stonebranch, Inc.

The installation JCL referenced by the installation steps is created by the `#SETUP` member in the Universal Products `INSTALL` library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the `BPX.FILEATTR.APF` profile in the `FACILITY` class.

1. Submit the JCL in member **UNVUN01**. The JCL allocates the SMP/E target and distribution data sets, creates the z/OS UNIX directories, and defines the DDDEFs in the SMP/E zones. All steps must end with a return code 0.

Step **ALLOCHFS** creates the z/OS UNIX directories required by the USS components. The path prefix where the directories are created was defined when the **#SETUP** member of the **INSTALL** library was modified and run as part of the z/OS installation. IBM recommends the path prefix to be **/usr/1pp**. Review the path prefix used in the **ALLOCHFS** step **PARM** value to be sure it meets local requirements.

The z/OS UNIX directory where the Universal Products directories are create must be mounted in read/write mode. The user ID used to run this job must have write access to the directory.

If the z/OS UNIX directories have already been created, the **ALLOCHFS** step will indicate this in its report and end successfully.

2. Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
3. Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step **APYFMID** must end with a condition code of 0.

Step **APYPTFS** is considered successful under any of the following conditions:

- Step ends with condition code 0.
- Step ends with condition code 4, and message GIM42001W is written in ddname SMPOUT.
- Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.

The user ID used to run the job must have **READ** access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the **APPLY** command.

4. Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step **ACCFMID** must end with a condition code of 0. Step **ACCPPTFS** is considered successful under any of the following conditions:
 - Step ends with condition code 0 or 4.
 - Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.
5. Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

3.6.2 Universal Products for z/OS USS 3.2.0 Upgrade, Existing CSI

The Universal Products 3.2.0 for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Universal Products 3.2.0 package in an existing SMP/E CSI.

The Universal Products for z/OS package must be upgraded first to 4.1.0 before the Universal Products for z/OS USS package is upgraded.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Universal Products **INSTALL** library.

Each step consists of running a batch job. The batch job must end with the appropriate return code before proceeding to the next step.

The user ID used to run the installation jobs must have a valid OMVS segment defined and have READ access to the **BPX.FILEATTR.APF** profile in the **FACILITY** class.

1. Submit the JCL in member **UNVUN02**. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets. All steps must end with a return code 0.
2. Submit the JCL in member **UNVUN03**. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step APYFMID must end with a condition code of 0.

Step APYPTFS is considered successful under any of the following conditions:

- Step ends with condition code 0.
- Step ends with condition code 4, and message GIM42001W is written in ddname **SMPOUT**.
- Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.

The user ID used to run the job must have READ access to **BPX.FILEATTR.APF** profile in the **FACILITY** class in order to set the APF attribute on the UDM file when it is created in the z/OS UNIX directory by the APPLY command.

3. Submit the JCL in member **UNVUN04**. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions:
 - Step ends with condition code 0 or 4.
 - Step ends with condition code 12, and message GIM24801S is written in ddname **SMPOUT**.
4. Submit the JCL in member **UNVUN05**. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

3.7 Universal Products for z/OS USS – Customization

The product executable files intended for command line use are located in the directory **#HFSPRE/universal/bin**, where **#HFSPRE** is the path prefix where the USS component directories were created. This directory must be added to the **PATH** environment variable for intended users of the executable files.

3.7.1 Universal Command Manager for z/OS USS Customization

Configuration

USS Universal Command Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is member **UCMCFGU0** of the **UNVCONF** library.

See the Universal Command 4.1.0 User Guide for details on configuring Universal Command Manager.

Product License

Universal Command Manager must be configured with a Universal Command product license before it is used. It will not run without a valid license.

The product license contains the license parameter keywords and their values, which must be entered into the Universal Command Manager configuration file. The values are specified in the same syntax as all other configuration options.

Product licenses are provided by your Stonebranch, Inc. account representative.

3.7.2 Universal Control Manager for z/OS USS Customization

Configuration

USS Universal Control Manager for z/OS uses a configuration file for system-wide customizations. The default configuration file is member **UCTCFGU0** of the **UNVCONF** library.

See the Universal Products Utilities 4.1.0 User Guide for details on configuring Universal Control Manager.

3.7.3 Universal Data Mover Manager for z/OS USS Customization

Configuration

USS Universal Data Mover Manager for z/OS uses a configuration file for system-wide customizations. The default configuration file is member **UDMCFGU0** of the **UNVCONF** library.

See the Universal Data Mover 4.1.0 User Guide for details on configuring Universal Data Mover.

Product License

Universal Data Mover Manager must be configured with a Universal Data Mover product license before it is used. It will not run without a valid license.

The product license contains the license parameter keywords and their values, which must be entered into the Universal Data Mover Manager configuration file. The values are specified in the same syntax as all other configuration options.

Product licenses are provided by your Stonebranch, Inc. account representative.

3.7.4 Universal Event Monitor Manager for z/OS USS Customization

Configuration

USS Universal Event Monitor Manager for z/OS uses a configuration file for system-wide customizations. The default configuration file is member **UEMCFGU0** of the **UNVCONF** library.

See the Universal Event Monitor 4.1.0 User Guide for details on configuring Universal Event Monitor Manager.

3.7.5 Universal Query for z/OS USS Customization

Configuration

USS Universal Query z/OS uses a configuration file for system-wide customizations. The default configuration file is member **UQRCFGU0** of the **UNVCONF** library.

See the Universal Products Utilities 4.1.0 Reference Guide for details on Universal Query configuration options.

3.8 Universal Products for z/OS USS – Data Set Inventory

3.8.1 SMP/E Data Sets

A number of SMP/E data sets are allocated and cataloged as part of the Universal Products for z/OS USS installation.

Table 3.3, below, lists the data sets and their space requirements.

Depending on your installation choices, the data set high-level qualifiers may be different.

Data Set Name	Space (tracks)	Description
UNV.AUNVHBIN	3000	SMP/E distribution library for z/OS UNIX executables.
UNV.AUNVHMLS	30	SMP/E distribution library for z/OS UNIX product national language support files.

Table 3.3 Universal Products for z/OS USS - SMP/E Data Sets

3.8.2 z/OS UNIX Directories

A number of z/OS UNIX directories are created as part of the Universal Products for z/OS USS installation.

Table 3.4, below, lists the directories and their space requirements.

Directory Name	Space (MB)	Description
universal/bin/IBM	80	SMP/E target directory for the SUNVHMLN DDDEF.
universal/bin	0	USS Universal Product program files.
universal/nls/IBM	0.5	SMP/E target directory for the SUNVHMLS DDDEF.
universal/nls	0	USS Universal Products national language support files.
universal/ucmdmgr	0	Universal Command Manager installation directory.
universal/uctlmgrr	0	Universal Control Manager installation directory.
universal/udmmgr	0	Universal Data Mover Manager installation directory.
universal/uemmgr	0	Universal Event Monitor Manager installation directory.
universal/uquery	0	Universal Query installation directory.

Table 3.4 Universal Products for z/OS USS - Directories

Chapter 4

Windows Installation

4.1 Overview

This chapter provides information on the installation of Stonebranch, Inc. Universal Products 4.1.0 on Microsoft Windows operating systems. Unless otherwise specified, all references to Universal Products for Windows in this chapter refer to version 4.1.0.

Universal Products 4.1.0 for Windows are provided in four separate easily installed, one-time installation packages:

- [Universal Products for Windows Package](#)
- [Universal Enterprise Controller Package](#)
- [Universal Enterprise Controller Client Applications Package](#)
- [Universal Products for SOA for Windows Package](#)

This chapter contains separate sections for each package.

Note: Starting with the 3.2.0 release of Universal Products, a Universal Broker must run on all systems on which a Universal Product component is running, including manager components. The Broker maintains product configuration data for all components that have a configuration file.

The Broker is required to run as **Local System account** on all Windows platforms.

4.1.1 Pre-Installation Guidelines

Before starting any Universal Products 4.1.0 for Windows installation, it is recommended that you close all other Windows programs. Active programs may be using system files that the Universal Products installation may update.

If a system file is in use during the installation, it is placed in a temporary location. A reboot is required to free the file and move it from the temporary location to its final destination. Closing all programs before the installation reduces the possibility that a file will be in use and that you will have to reboot your computer.

Note: In order to install these products, you must be able to write to the directory from which the installation is launched.

Installation Paths

On 32-bit Windows systems, the default installation path is:

- "C:\Program Files"

On 64-bit Windows systems, the default installation path is:

- "C:\Program Files (x86)"

4.1.2 Product Distribution Files

[Table 4.1](#), below, identifies the product distribution files that Stonebranch, Inc. provides for its four Universal Products 4.1.0 for Windows packages.

Package	Distribution File
Universal Products	unv-4.1.0.<level>-nt-i386.exe
Universal Enterprise Controller	UECt1r-4.1.0.<level>-nt-i386.exe
UEC Client Applications	UECClient-4.1.0.<level>-nt-i386.exe
Universal Products for SOA	ups-4.1.0.<level>-nt-i386.exe
Note: level is the product maintenance level.	

Table 4.1 Universal Products for Windows - Product Distribution Files

To obtain a Universal Products 4.1.0 package, you must download the corresponding product distribution file from the Support / Current Products area of the Stonebranch, Inc. website, <http://www.stonebranch.com>.

A customer user ID and password – provided by Stonebranch, Inc. – are required to access this area.

Distribution Files Format

The product distribution files are in a Windows self-extracting archive file format.

The names of the distribution files have the following format:

Product-Version.Release.Modification Level.Maintenance Level-operating system-platform.exe

For example: `unv-4.1.0.1-nt-i386.exe`

In this format:

- **Product** is the name of the product.
- **Version** is the current version of Universal Products.
- **Release** is the current release of Universal Products.
- **Modification Level** is the current Universal Products feature set.
- **Maintenance Level** is the Universal Products build level.
- **operating system** is the name of the operating system.
- **platform** is the targeted hardware platform.

4.1.3 Windows Installer

The installation of each Universal Products for Windows package requires Windows Installer from Microsoft. Windows Installer is a service application that is a standard component of Windows operating systems. (It also can be obtained from Microsoft's website.)

Each Universal Products for Windows package provides version **3.1.4000.1823** of Windows Installer (`msiexec.exe`).

Installing Windows Installer with a Universal Products Package

If Windows Installer is not present on your Windows operating system, the Universal Products package being installed will install it before any of its own files are copied.

If Windows Installer is present on your Windows operating system, but it is a version prior to **3.1.4000.1823**, the installation will upgrade it.

Determining if Windows Installer will be Installed or Upgraded

To determine if the installation of a Universal Products package also will install or upgrade Windows Installer, perform the following steps:

1. From Windows Explorer, go to the 32-bit Windows system directory (for example, `\Windows\System32`, `\Windows\SysWow64`).
2. Search for file `msiexec.exe` (Windows Installer).
 - If the file exists, continue to the Step 3.
 - If the file does not exist, Windows Installer is not installed on your system. The Universal Products installation will install it.
3. Right-click on the `msiexec.exe` file name to display a pop-up menu.
4. Click **Properties** to display the Properties dialog for `msiexec.exe`.
5. Click the **Version** tab. File Version: identifies the currently installed version of Windows Installer (Figure 4.1, below). If the version is prior to **3.1.4000.1823**, the package installation will upgrade it.

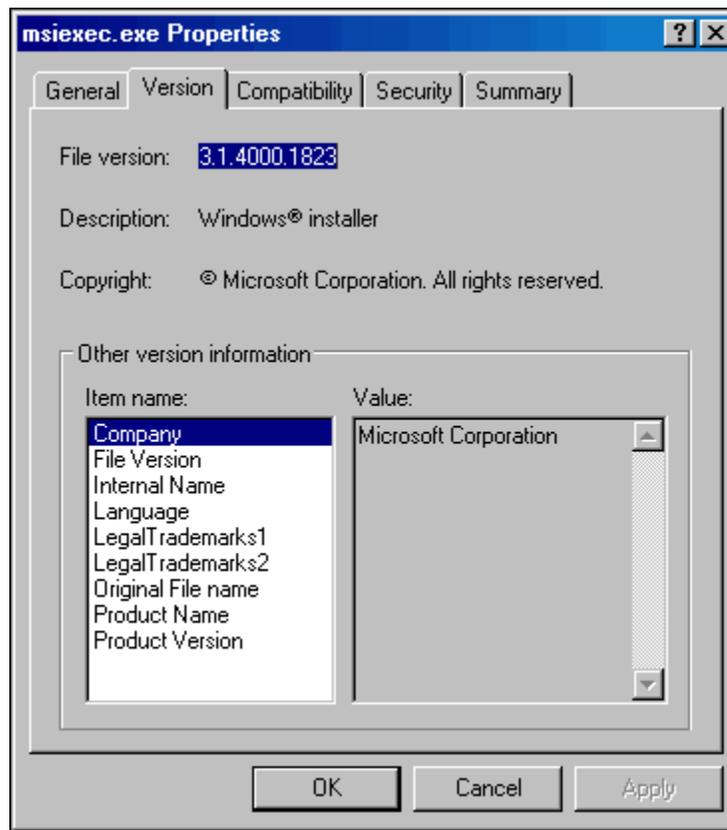


Figure 4.1 Windows Installer - Properties / Version dialog

(In [Figure 4.1](#), the installed Windows Installer is identified as version 3.1.4000.1823. Since this is the same version provided by the package installation, it will not be upgraded.)

Rebooting the System

If Windows Installer is installed or upgraded during the installation of the package, a reboot of the system is required.

The installation provides an option to either:

- Reboot after the installation is complete.
- Reboot immediately. The installation will resume automatically after Windows restarts.

If Windows Installer version 3.1.4000.1823 or above already exists on your system, it is not upgraded, and no reboot is required.

Installing Windows Installer Separately from a Universal Products Package

The Windows Installer version provided with each Universal Products package also can be installed separately from the package; that is, prior to the installation of the package.

To install Windows Installer separately, perform the following steps:

1. Execute the package distribution file downloaded from the Stonebranch, Inc. website. Make note of the directory into which the installation files are extracted.

After extraction of the files is complete, the installation will begin.

2. Cancel the installation.
3. Go to the directory where the extracted files reside.
4. Execute **WindowsInstaller-KB893803-x86.exe**.
 - If Windows Installer was upgraded, reboot the system.
 - If Windows Installer was installed new, no reboot is required.

4.2 Universal Products for Windows Package

The Universal Products 4.1.0 for Windows package includes the following product components:

- Universal Broker 4.1.0
- Universal Command Manager and Server 4.1.0
- Universal Control Manager and Server 4.1.0
- Universal Data Mover Manager and Server 4.1.0
- Universal Encrypt 4.1.0
- Universal Event Log Dump 4.1.0
- Universal Event Monitor Manager and Server 4.1.0
- Universal Message to Exit Code Translator 4.1.0
- Universal Query 4.1.0

Note: Universal Enterprise Controller 4.1.0, Universal Enterprise Controller 4.1.0 Client Applications, and Universal Products for SOA 4.1.0 are packaged separately (see sections [4.8 Universal Enterprise Controller Package](#), [4.14 Universal Enterprise Controller Client Applications Package](#), and [4.19 Universal Products for SOA for Windows Package](#)).

4.2.1 Product Compatibility

[Table 4.2](#), below, identifies the compatibility of Universal Products 4.1.0 for Windows with previous product versions.

Product	Compatibility
Universal Command Manager 4.1.0	Universal Command Server 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0 and 1.2.0
Universal Command Server 4.1.0	Universal Command Manager 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0 and 1.2.0
Note: Universal Command Manager and Server 1.1.0 are not compatible with Manager and Server 2.1.0 and above.	

Table 4.2 Universal Products for Windows - Product Compatibility

4.3 Universal Products for Windows – Installation Requirements

To install Universal Products for Windows, you must have one of the following versions of Windows:

- Windows 2000 Professional or Server, with Service Pack 4
- Windows Server 2003
- Windows XP
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows Server 2008 R2

In addition, you must have:

- Approximately 50 megabytes of disk space for the installation
- An account with administrative privileges

4.3.1 Spool Directory

The spool directory is used to store the following types of information:

- Execution information for Universal Products components started by Universal Broker.
- Event definitions and event handlers managed by Universal Broker and used by Universal Event Monitor.
- Results of events tracked by Universal Event Monitor.
- Redirected standard I/O files (stdin, stdout, and stderr) captured by Universal Command when run with manager fault tolerance enabled.
- Configuration information for Universal Products components, when a local Universal Broker is operating in managed mode.

The default location for the spooled standard I/O files is `.\Universal\spool\ucmdsrv`.

The default location for the other database files is `.\Universal\spool\ubroker`.

Location

The spool directory must reside on a local device. It cannot reside on any network device, including network drives that may be mapped to a local drive. By default, the spool files are located in directory `.\Universal\spool\ucmdsrv`.

Space

The amount of disk space required for the spool directory depend on the following factors:

- Number of spooling user processes that will be executing simultaneously. A user process is created for each command requested by a Universal Command Manager. The default maximum number is 50.
- When a user process ends and a Manager has received all the spool files, the spool files themselves are deleted.
- Average size of the user processes standard input, output, and error files. Keep in mind that spooling is not a feature for file transfer purposes. File transfer-related processes should execute without spooling enabled.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

$\text{MAX-PROCESSES} \times \text{AVERAGE-STDIO-SIZE} \times 2 = \text{required disk space.}$

For example, if the maximum number of simultaneous user processes is estimated at 20 and the average size of processes standard I/O files is 100,000 bytes, the average amount of required disk space is 4MB (20 x 100000 x 2).

The Universal Command Server is configured with spooling disabled to prevent unintentional disk utilization. This feature can be enabled through the Universal Configuration Manager.

Details on the Manager Fault Tolerant feature, the spooling of redirected standard I/O files, and the Universal Configuration Manager can be found in the Universal Command 4.1.0 User Guide.

Security

Universal Broker and Universal Command Server require read/write access to the spool directory. No other Universal Products access the spool directory directly.

During installation, the LocalSystem account is granted full control to the spool directory. In addition, all administrators are given read / write / modify access. No general user access is granted.

4.4 Universal Products for Windows – Installation

Windows Installer provides two interfaces that you can use to install Universal Products: graphical and command line.

This section provides the following procedures regarding these two interfaces:

- [Installing Universal Products via the Windows Installer Graphical Interface](#)
- [Modifying a Universal Products Installation via the Windows Installer Graphical Interface](#)
- [Installing Universal Products via the Windows Installer Command Line Interface](#)
- [Modifying a Universal Products Installation via the Windows Installer Command Line Interface](#)

4.4.1 Installing Universal Products via the Windows Installer Graphical Interface

To install Universal Products for Windows using the Windows Installer graphical interface, perform the following steps:

1. Download the Universal Products for Windows product distribution file, `unv-4.1.0.<level>-nt-i386.exe`, to your work station.
2. Execute the distribution file to extract the files.

Note: If you already have extracted the files from the distribution file, but cancelled installation in order to separately install Windows Installer (see Section [4.1.3 Windows Installer](#)), you can simply double-click the extracted Universal Products installation file, `Ucmd.msi`, to begin the installation.

Installing Over a Remote Desktop Session

Starting with Windows Server 2003, Remote Desktop provides distinct session environments for each logged-in user. This means extraction may use an environment setting that is not available once the Remote Desktop session ends.

For example, the distribution file's default extraction location is based on the TEMP environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed.

To ensure that extracted files and other required resources are accessible after the initial install, extract the files to a well-known location that is not likely to change between Remote Desktop sessions.

Refer to the Microsoft documentation on the Remote Desktop feature for additional information.

The installation starts after the files are extracted.

It first will verify that your machine meets the minimum system requirements (see Section 4.3 [Universal Products for Windows – Installation Requirements](#)). If the requirements are met, a Welcome dialog displays ([Figure 4.2](#)).

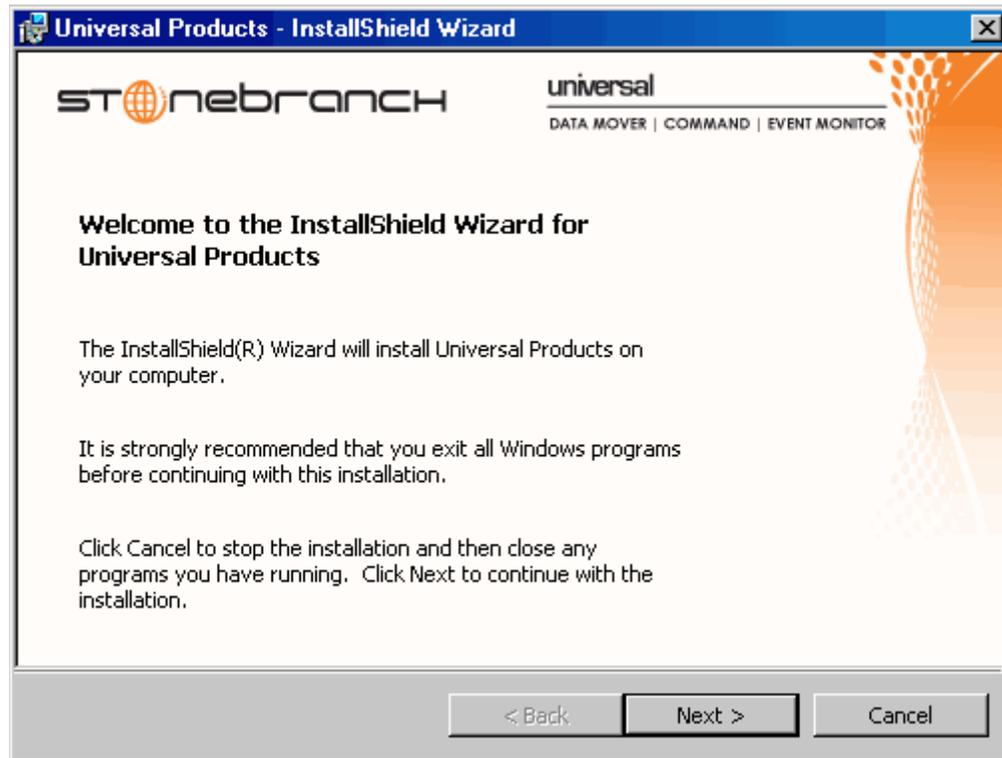


Figure 4.2 Universal Products - Welcome dialog

3. Click the **Next** button. A list of Universal Products components included in the installation package then displays ([Figure 4.3](#)). It is from this list that you can select which components to install.

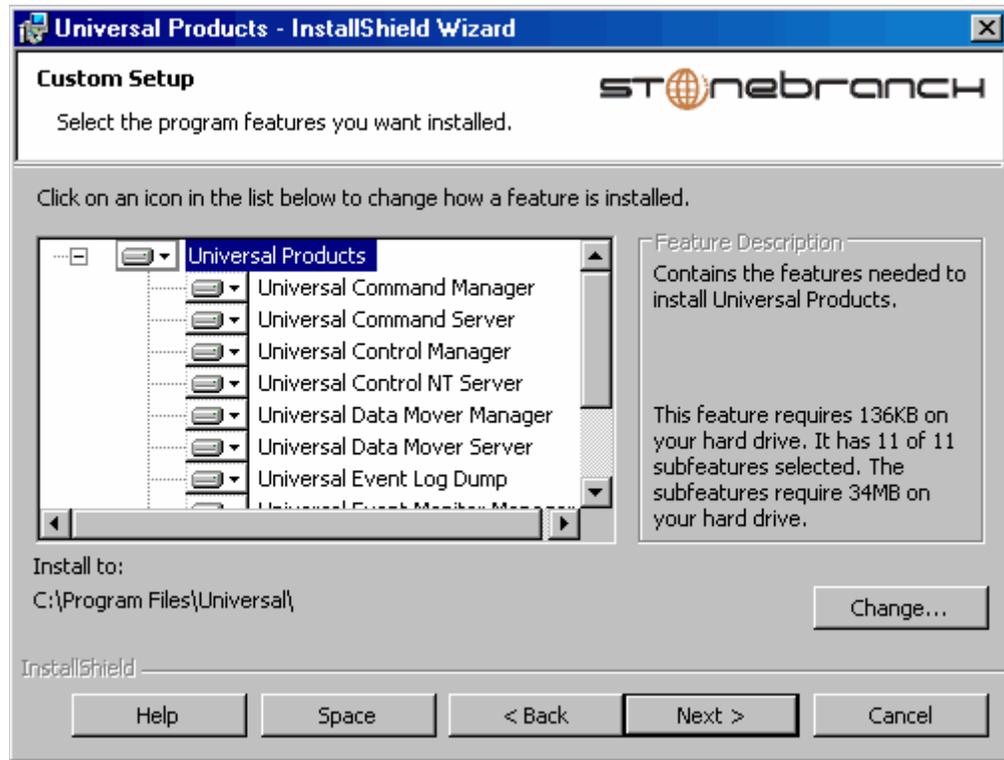


Figure 4.3 Universal Products - Custom Setup dialog

For a new installation, a drive icon  displays next to each item in the list, indicating that the component will be installed.

For an upgrade installation, either of the following icons displays next to an item:

- A drive icon  indicates that the component is either:
 - New to the installation and will be installed.
 - Currently is installed and will be upgraded.
- An **X** icon indicates that the component is either:
 - Currently not installed (but previously was available).
 - Previously installed but removed.



Stonemason's Tip

If the installation detects an existing Universal Products installation, currently installed components may be upgraded.
(Currently, there is no way to specify that the state of a currently installed component remain unchanged.)

If a component is selected for installation, and the version of the installed component is less than the version of the component being installed, the installed component will be replaced by the component being installed.

If a component is not selected for installation (that is, the **X** icon is selected), and it currently is installed, the new installation will remove that component.

4. [Figure 4.3](#) indicates that all Universal Products components will be installed in their respective directories under the `C:\Program Files\Universal` directory.
 - a. If you want to select a different location, click the **Change...** button.
 - b. If you want to check the amount of disk space required for the installation, and the amount of available disk space on the selected directory, click the **Space** button.
5. If you do not want to install a component:
 - a. Click the drive icon  next to that component name.
 - b. From the drop-down list that displays, select the **X** icon to mark the component as one not to be installed.

For example, [Figure 4.4](#), below, indicates that Universal Event Log Dump has been selected to not be installed.

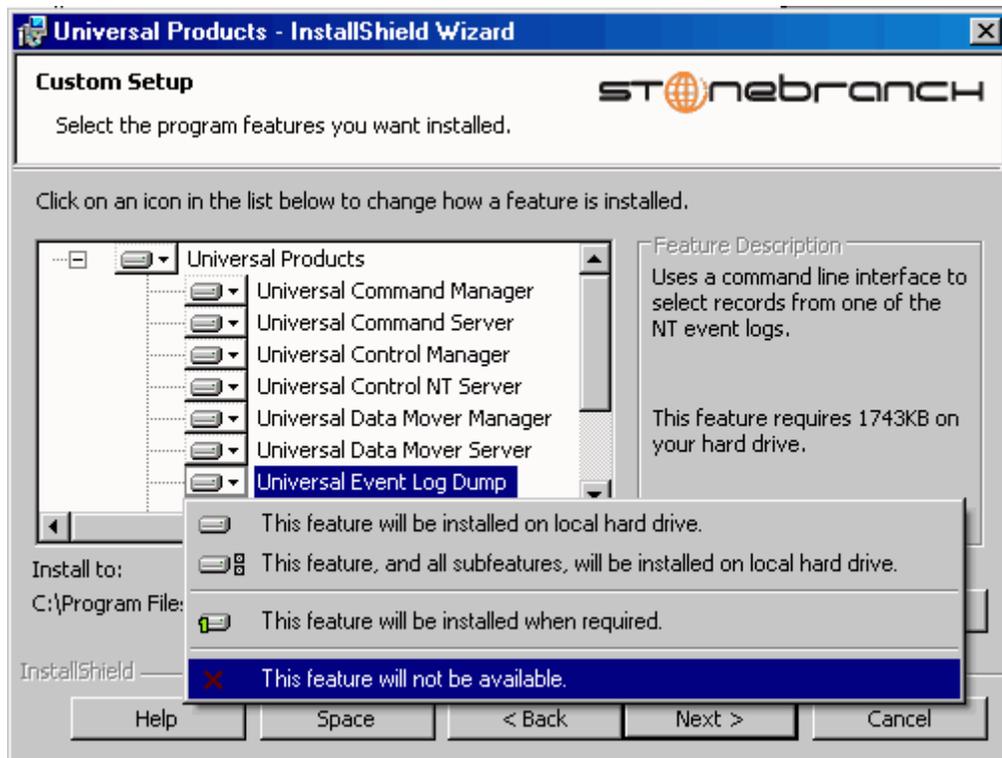


Figure 4.4 Universal Products - Custom Setup / Selecting to Not Install a Component

6. When you have selected the components (and their installation destinations) that you want to install, click the **Next>** button to continue the installation process.

Depending on the components selected, the installation may prompt for additional values, such as working folders for Server components.

Follow the directions provided with each dialog to complete the installation.

When the installation completes successfully, the Installation Complete dialog displays ([Figure 4.5](#)).

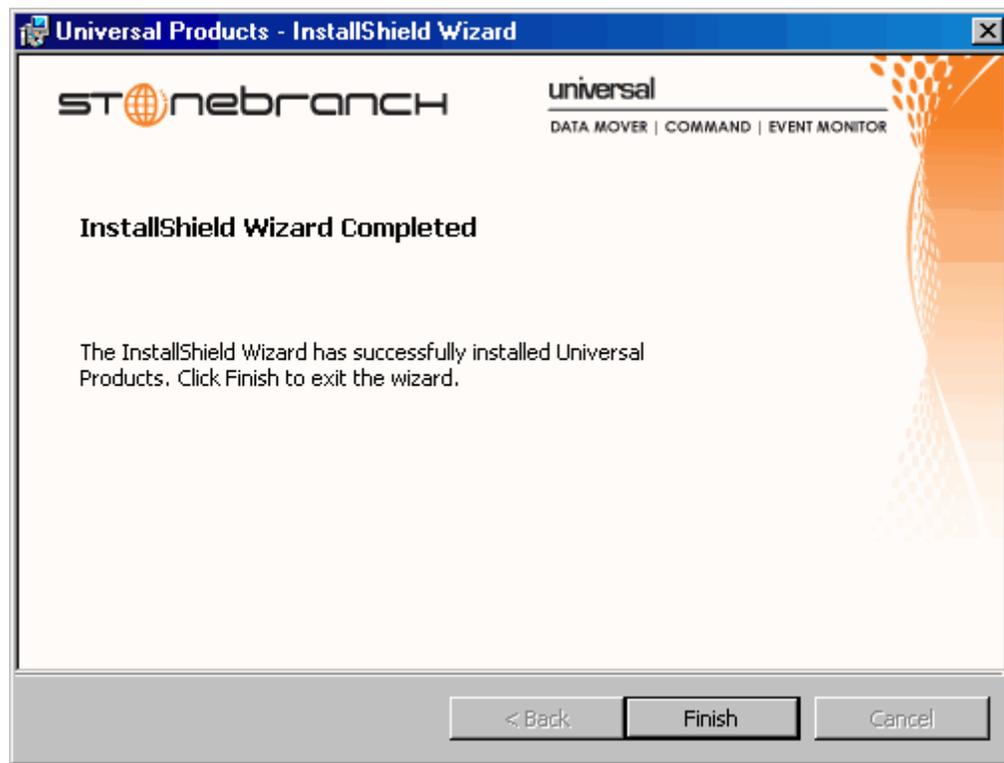


Figure 4.5 Universal Products - Installation Complete dialog

7. Click the **Finish** button to exit Windows Installation.

4.4.2 Modifying a Universal Products Installation via the Windows Installer Graphical Interface

This section describes how to modify a Universal Products installation via the Windows Installer graphical interface.

After Universal Products is installed, the Windows Installer can be run as many times as needed to modify the installation by:

- [Adding or Removing Universal Products Components](#)
- [Repairing a Corrupted Universal Products Installation](#)
- [Removing a Universal Products Installation](#)

Adding or Removing Universal Products Components

To add or remove components from a Universal Products installation, perform the following steps:

1. On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays (Figure 4.6, below).
2. From the list of installed programs, select **Universal Products**.

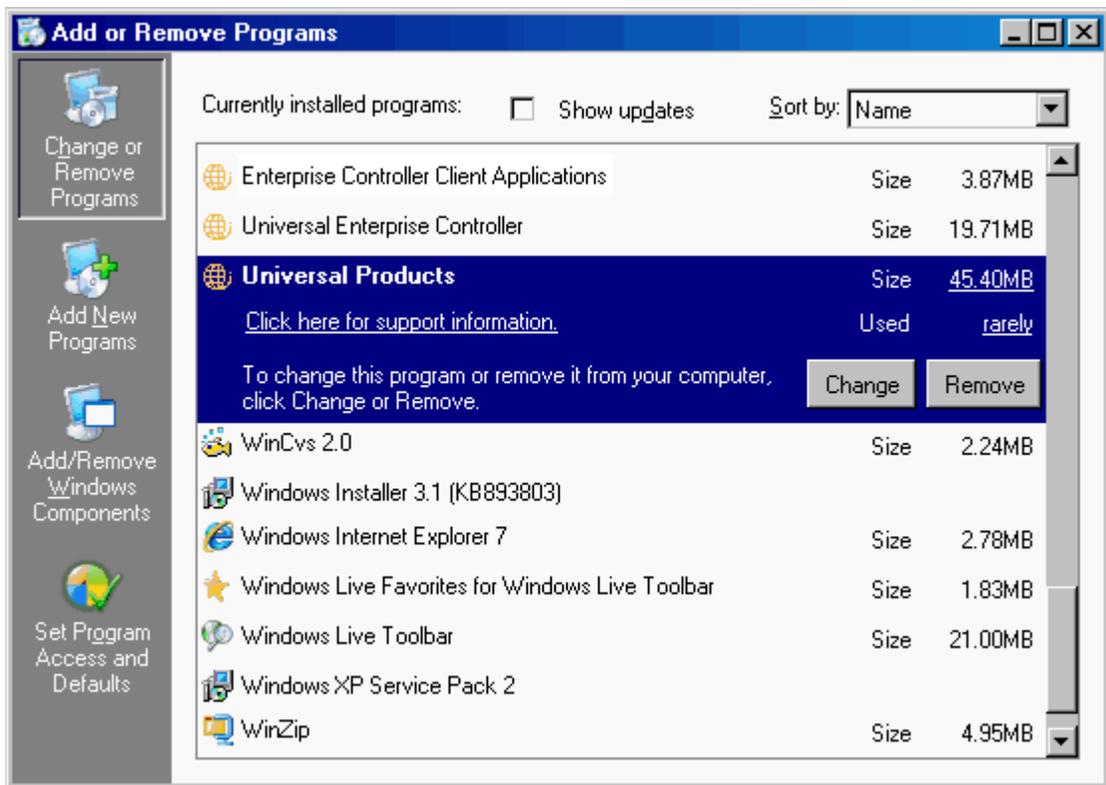


Figure 4.6 Universal Products - Add or Remove Programs dialog

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

The Programs and Features dialog (Figure 4.7) replaces the Add or Remove Programs dialog.

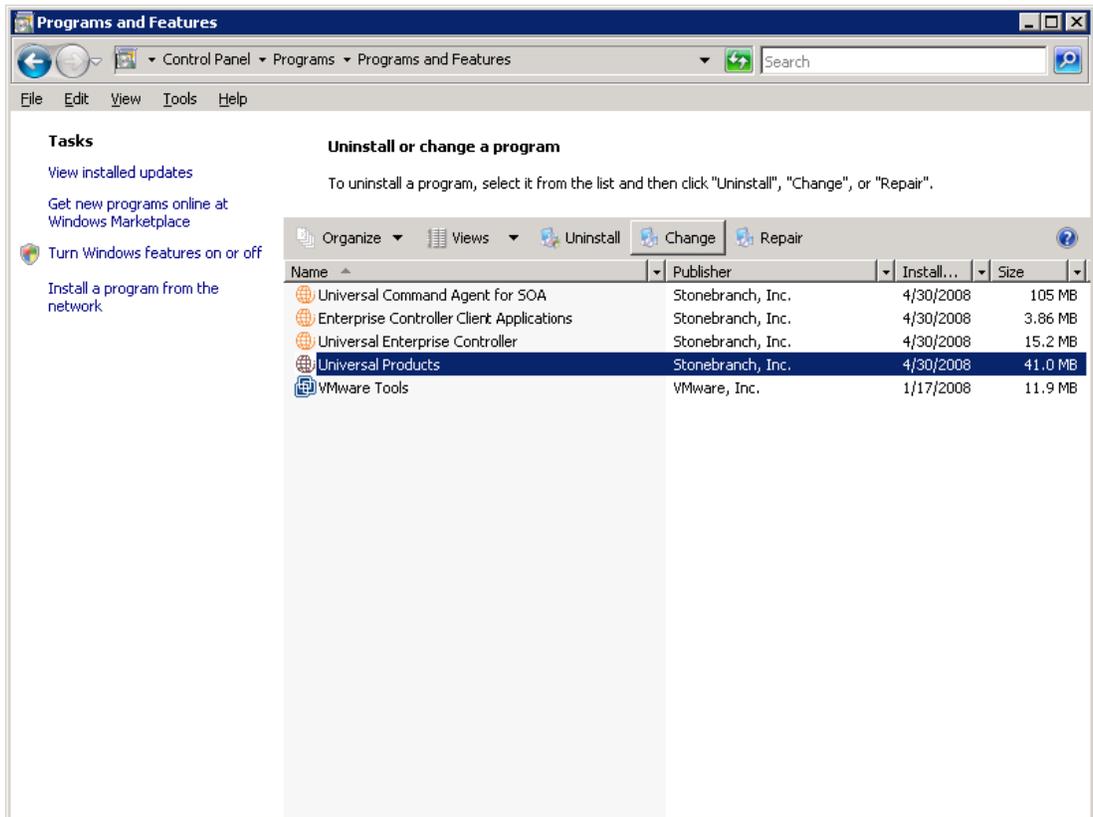


Figure 4.7 Windows Programs and Features Dialog (Windows Vista, Windows Server 2008 or later)

3. Click the **Change** button to start Windows Installer.
4. On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays (Figure 4.9).

Installing Over a Remote Desktop Session

If Universal Products was installed via a Remote Desktop connection, the environment settings used during that session may no longer be available.

Starting with Windows Server 2003, Remote Desktop provides distinct session environments for each logged-in user.

The distribution file's default extraction location is based on the TEMP environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed. Consequently, any attempts to modify the installation may fail because Windows Installer cannot locate the installation's source files (a dialog similar to the one shown in Figure 4.8, below, may display).

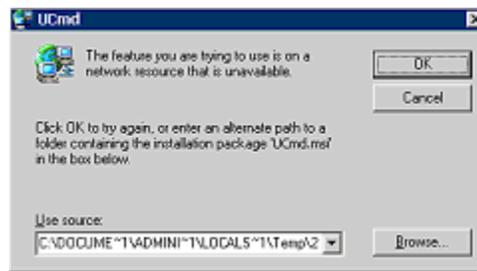


Figure 4.8 Failure to Access Install Files

To resolve this issue, re-extract the installation files to a location that is independent of a Remote Desktop environment and specify that location in the dialog above. Keep in mind, however, that the extracted files must come from the same distribution file used for the initial installation. If matching installation files cannot be found, Universal Products must be uninstalled and then reinstalled with the desired modifications.

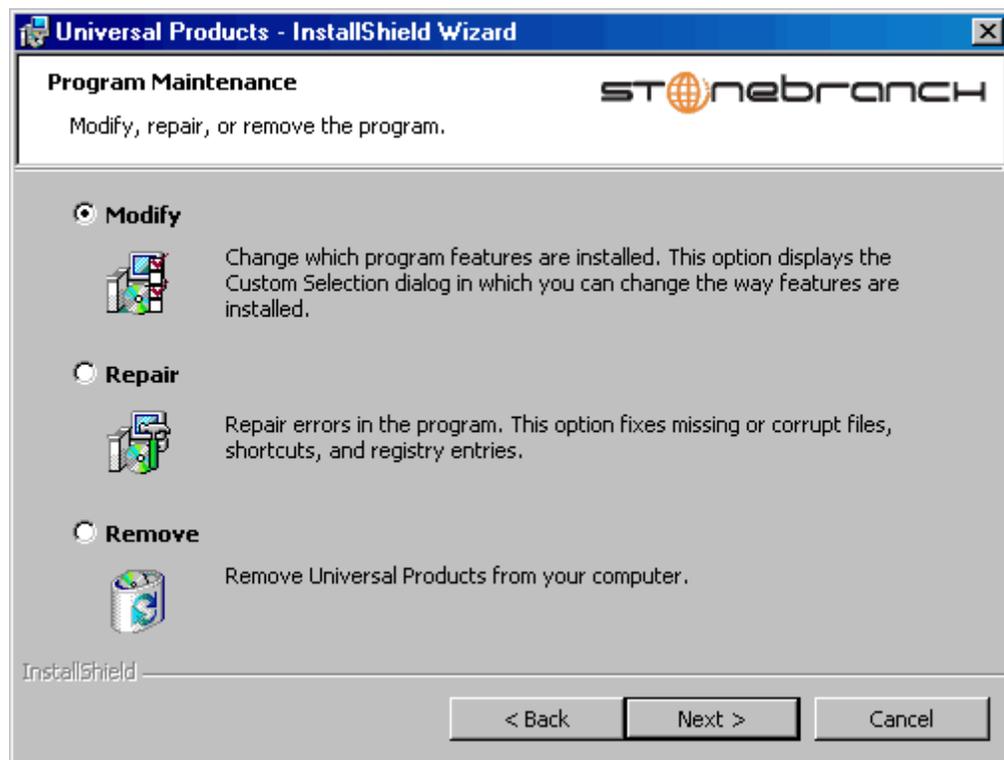


Figure 4.9 Universal Products - Program Maintenance / Modify dialog

5. Click the **Modify** radio button, and then the **Next>** button, to display the Custom Setup dialog (see [Figure 4.10](#)).

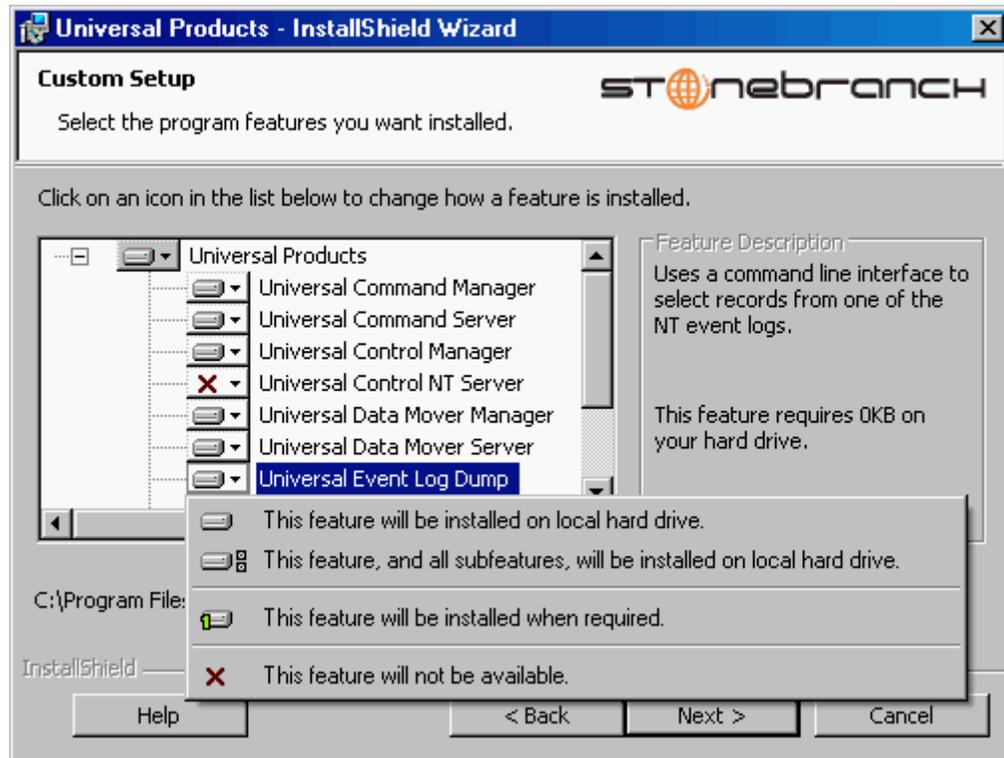


Figure 4.10 Universal Products - Custom Setup / Modifying an Installed Component

Currently installed components are identified by a drive icon .

Uninstalled components are identified by an **X** icon.

6. To remove a currently installed component:
 - a. Click the drive icon  next to that component.
 - b. Select the **X** icon from the drop-down list to mark the component for removal.
7. To add an uninstalled component:
 - a. Click the **X** icon next to that component.
 - b. Select the drive icon  from the drop-down list to mark the component for installation.
8. Click the **Next>** button to continue with the modification.

When the modifications are complete, the following actions will be taken:

- Components marked with a drive icon  will:
 - Remain installed if they already are installed.
 - Be installed if they are not already installed.
- Components marked with an **X** will:
 - Remain uninstalled if they are not currently installed
 - Be removed if they currently are installed.

Repairing a Corrupted Universal Products Installation

Windows Installer has the ability to recover accidentally deleted application files, configuration and component definition files, and registry entries required by Universal Products. This repair feature will re-install the missing items, making a complete re-install unnecessary.

During a repair, any options stored in configuration and component definition files are preserved. If a component definition or configuration file was deleted, the installation will create a new configuration file with default values.

To repair an installation, perform the following steps:

1. On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays (see [Figure 4.6](#)).

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

The Programs and Features dialog (see [Figure 4.7](#)) replaces the Add or Remove Programs dialog.

2. From the list of installed programs, select **Universal Products**.

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

To skip the remaining steps, simply select the **Repair** button (see [Figure 4.7](#)).

3. Click the **Change** button to start Windows Installer.

4. On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays (Figure 4.11, below).

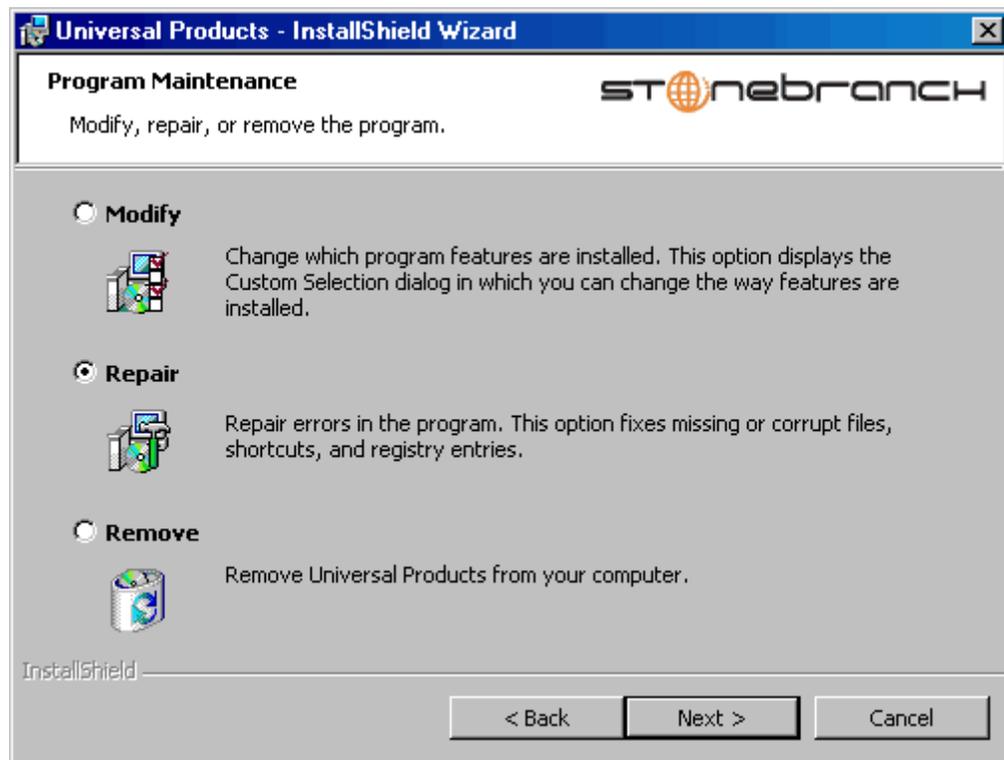


Figure 4.11 Universal Products - Program Maintenance / Repair dialog

5. Click the **Repair** radio button, and then the **Next>** button, to display the Ready to Repair dialog.
6. Follow the instructions displayed in successive dialogs to complete the repair.

Removing a Universal Products Installation

To uninstall a Universal Products installation, perform the following steps:

1. On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays (see [Figure 4.6](#)).

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

The Programs and Features dialog (see [Figure 4.7](#)) replaces the Add or Remove Programs dialog.

2. From the list of installed programs, select **Universal Products**.

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

To skip the remaining steps, simply select the **Uninstall** button (see [Figure 4.7](#)).

3. Click the **Change** button to start Windows Installer.
4. On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays ([Figure 4.12](#), below).

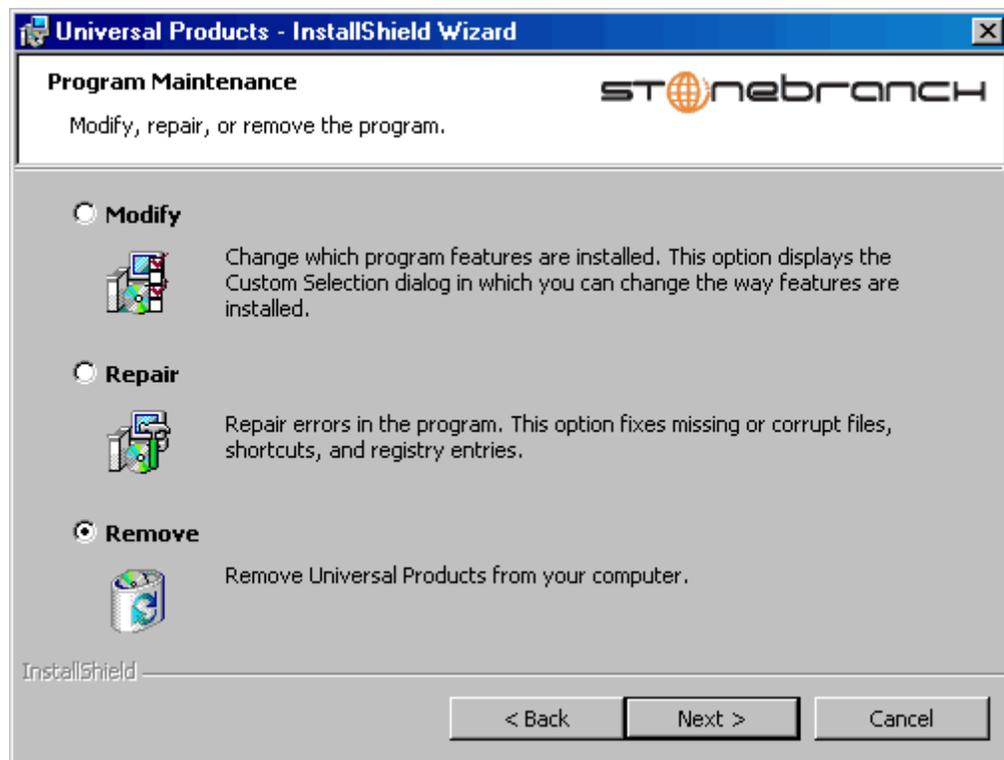


Figure 4.12 Universal Products - Program Maintenance / Remove dialog

5. Select **Remove** and click the **Next>** button.
6. On the subsequent dialog, click the **Remove** button to remove the Universal Products installation.

Un-Installed Files

The un-install process will remove only those files created during the installation. Some files stored under the `.\Universa1` install directory by Universal Products, such as trace files, files created under the Universal Command Server working directory, and spool files, may be left behind after the un- install. In this situation, those files and/or directories may simply be deleted.

Before deleting the entire `.\Universa1` directory, make sure that no other Stonebranch, Inc. products are installed there. (See Section [4.7 Universal Products for Windows – File Inventory Lists](#) for a list of files and directories installed with Universal Products.)

In addition to those files and directories created by the Universal Products installation, there may be some shared files (for example, codepage files) and Universal Products components (for example, Universal Configuration Manager and Universal Encrypt) that may be left behind following an un-install. These components will be removed when the last Stonebranch Inc. product that uses them is un-installed.

4.4.3 Installing Universal Products via the Windows Installer Command Line Interface

This section describes how to install Universal Products using the Windows Installer command line interface.

The command line interface is useful in situations where:

1. Several Universal Products installations must be deployed across many different systems.
2. It is not practical or convenient to perform the graphical interface installation.

To use the Windows Installer command line interface, the Universal Products installation files first must be extracted from the product distribution file. Since there is no command line option available to unpack the distribution file, this must be done via the graphical user interface:

1. Download the Universal Products for Windows product distribution file, `unv-4.1.0.<level>-nt-i386.exe`, to your work station.
2. Execute the distribution file to extract the installation files.
3. On the Location to Save Files dialog (Figure 4.13, below), select a location in which to store the installation files and click the **Next>** button to extract the files.

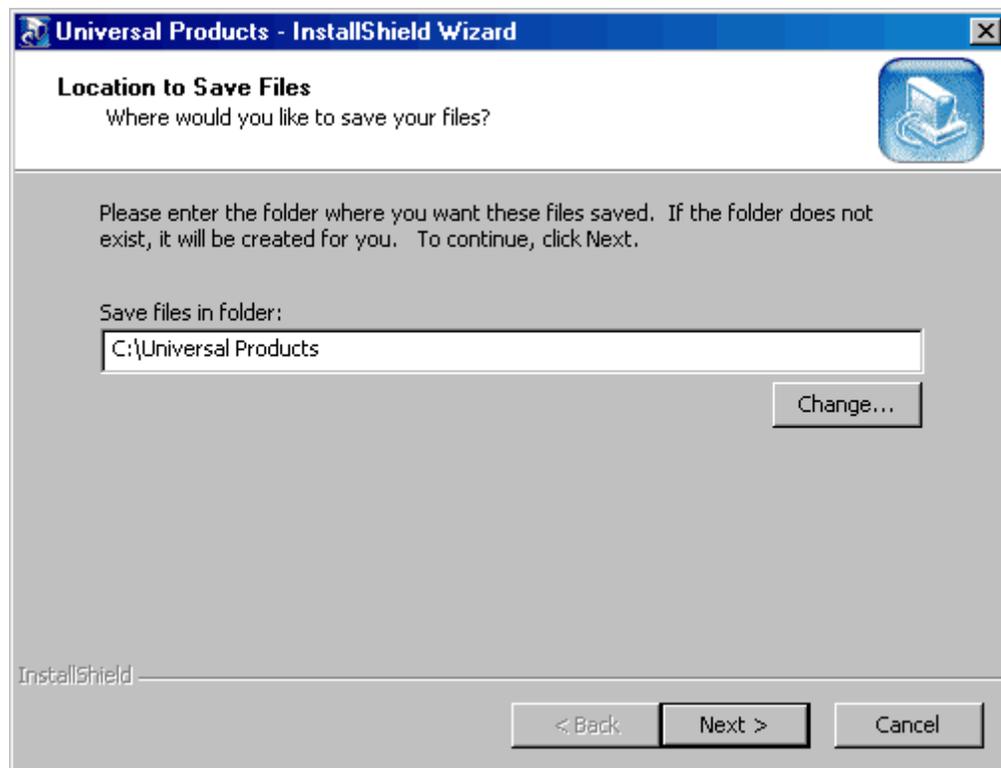


Figure 4.13 Universal Products - Location to Save Files dialog

4. On the Welcome dialog (see [Figure 4.2](#)), click the **Cancel** button.

At this point, all the installation files have been extracted into the location specified, but nothing has been installed. You may now complete the install from the command line, with no additional graphical interaction required.

Typically, to install an application that uses Windows Installer, a program such as **setup.exe** is executed. This program usually checks to make sure that a supported version of Windows Installer is installed. If one is not, the program will install it before proceeding. In fact, this is exactly what happens when Universal Products for Windows is installed from the product distribution file.

There is a command line interface available for **setup.exe**, but running it is not really ideal for unattended installations. If Windows Installer is installed or upgraded on the target system, a reboot may be required before the installation of Universal Products can continue (see [Determining if Windows Installer will be Installed or Upgraded](#) in [Section 4.1.3 Windows Installer](#)).

A better way to install Universal Products from the command line is to execute Windows Installer, **msiexec.exe** (see [Windows Installer Installation Commands](#)).

Note: A silent install also can be performed from the Universal Products for Windows distribution file (see [Using the Distribution File for a Silent Install](#)).

Windows Installer Command Line Parameters

Table 4.3, below, describes the parameters available when using Windows Installer to install Universal Products from the command line.

Parameter	Description
/i	Installs Universal Products. This parameter cannot be used with either the /f (repair) or /x (remove) parameter.
/fom	Repairs a Universal Products installation. The om after the /f are actually options used by the repair. There are other options available, but for behavior that matches the repair done from the graphical install, the om options must be used. This parameter cannot be used with either the /i (install) or /x (remove) parameter.
/x	Removes Universal Products. This parameter cannot be used with either the /i (install) or /f (repair) parameter.
SetupPath\Ucmd.msi	Specifies the path to the Ucmd.msi file. If the product distribution file is used, SetupPath will be whatever directory into which the files are extracted. Of course, the files may be copied from the original extraction directory into any location desired. In that case, SetupPath must point to the new location.
INSTALLDIR	Sets the root installation directory to InstallDir . Each component will be installed under this directory. This optional parameter is required only if you want to install Universal Products under a directory different from the one specified by the PROGRAMFILES environment variable (typically C:\Program Files\Universal). This parameter is valid only when the /i switch is used. Otherwise, it is ignored. If the directory contains spaces, you must use double (") quotation marks around the path name.
UCMSRVWORKFOLDER	Sets the Universal Command Server working folder. This optional parameter is required only if you want to set the Universal Command Server working directory to something other than INSTALLDIR\UcmdHome . This value is used only if the Universal Command Server is being installed. It is valid only when the /i switch is used. Otherwise, it is ignored. If the directory contains spaces, you must use double (") quotation marks around the path name.
UCTLSRVWORKFOLDER	Sets the Universal Control Server working folder. This optional parameter is required only if you want to set the Universal Control Server working directory to something other than INSTALLDIR\UcmdHome . This value is used only if the Universal Control Server is being installed. It is valid only when the /i switch is used. Otherwise, it is ignored. If the directory contains spaces, you must use double (") quotation marks around the path name.

Parameter	Description
<p>UDMSRVWORKFOLDER</p>	<p>Sets the Universal Data Mover Server working folder.</p> <p>This optional parameter is required only if you want to set the Universal Data Mover Server working directory to something other than INSTALLDIR\UdmHome. This value is used only if the Universal Data Mover Server is being installed.</p> <p>It is valid only when the /i switch is used. Otherwise, it is ignored. If the directory contains spaces, you must use double (") quotation marks around the path name.</p>
<p>UEMSRVWORKFOLDER</p>	<p>Sets the Universal Event Monitor Server working folder.</p> <p>This optional parameter is required only if you want to set the Universal Event Monitor Server working directory to something other than INSTALLDIR\UemHome. This value is used only if the Universal Event Monitor Server is being installed.</p> <p>It is valid only when the /i switch is used. Otherwise, it is ignored. If the directory contains spaces, you must use double (") quotation marks around the path name.</p>
<p>UCMDMGR</p>	<p>Prevents Universal Command Manager from being installed (if value is <i>no</i>). This has the same effect as choosing not to install the Universal Command Manager from the Custom Setup dialog (see Section 4.4 Universal Products for Windows – Installation).</p> <p>This optional parameter is required only if you do not want to install the Universal Command Manager.</p> <p>It is valid only when the /i switch is used. Otherwise, it is ignored.</p> <p>[Default is yes.]</p>
<p>UCMDSRV</p>	<p>Prevents Universal Command Server from being installed (if value is <i>no</i>). This has the same effect as choosing not to install the Universal Command Server from the Custom Setup dialog (see Section 4.4 Universal Products for Windows – Installation).</p> <p>This optional parameter is required only if you do not want to install the Universal Command Server.</p> <p>It is valid only when the /i switch is used. Otherwise, it is ignored.</p> <p>[Default is yes.]</p>
<p>UCTLMGR</p>	<p>Prevents Universal Control Manager from being installed (if value is <i>no</i>). This has the same effect as choosing not to install the Universal Control Manager from the Custom Setup dialog (see Section 4.4 Universal Products for Windows – Installation).</p> <p>This optional parameter is required only if you do not want to install the Universal Control Manager.</p> <p>It is valid only when the /i switch is used. Otherwise, it is ignored.</p> <p>[Default is yes.]</p>
<p>UCTLSRV</p>	<p>Prevents Universal Control Server from being installed (if value is <i>no</i>). This has the same effect as choosing not to install the Universal Control Server from the Custom Setup dialog (see Section 4.4 Universal Products for Windows – Installation).</p> <p>This optional parameter is required only if you do not want to install the Universal Control Server.</p> <p>It is valid only when the /i switch is used. Otherwise, it is ignored.</p> <p>[Default is yes.]</p>

Parameter	Description
UDMMGR	<p>Prevents Universal Data Mover Manager from being installed (if value is <i>no</i>). This has the same effect as choosing not to install the Universal Data Mover Manager from the Custom Setup dialog (see Section 4.4 Universal Products for Windows – Installation).</p> <p>This optional parameter is required only if you do not want to install the Universal Data Mover Manager.</p> <p>It is valid only when the /i switch is used. Otherwise, it is ignored.</p> <p>[Default is yes.]</p>
UDMSRV	<p>Prevents Universal Data Mover Server from being installed (if value is <i>no</i>). This has the same effect as choosing not to install the Universal Data Mover Server from the Custom Setup dialog (see Section 4.4 Universal Products for Windows – Installation).</p> <p>This optional parameter is required only if you do not want to install the Universal Data Mover Server.</p> <p>It is valid only when the /i switch is used. Otherwise, it is ignored.</p> <p>[Default is yes.]</p>
UELD	<p>Prevents Universal Event Log Dump from being installed (if value is <i>no</i>). This has the same effect as choosing not to install the Universal Event Log Dump from the Custom Setup dialog (see Section 4.4 Universal Products for Windows – Installation).</p> <p>This optional parameter is required only if you do not want to install the Universal Event Log Dump.</p> <p>It is valid only when the /i switch is used. Otherwise, it is ignored.</p> <p>[Default is yes.]</p>
UEMMGR	<p>Prevents Universal Event Monitor Manager from being installed (if value is <i>no</i>). This has the same effect as choosing not to install the Universal Event Monitor Manager from the Custom Setup dialog (see Section 4.4 Universal Products for Windows – Installation).</p> <p>This optional parameter is required only if you do not want to install the Universal Event Monitor Manager.</p> <p>It is valid only when the /i switch is used. Otherwise, it is ignored.</p> <p>[Default is yes.]</p>
UEMSRV	<p>Prevents Universal Event Monitor Server from being installed (if value is <i>no</i>). This has the same effect as choosing not to install the Universal Event Monitor Server from the Custom Setup dialog (see Section 4.4 Universal Products for Windows – Installation).</p> <p>This optional parameter is required only if you do not want to install the Universal Event Monitor Server.</p> <p>It is valid only when the /i switch is used. Otherwise, it is ignored.</p> <p>[Default is yes.]</p>
UQUERY	<p>Prevents Universal Query from being installed (if value is <i>no</i>). This has the same effect as choosing not to install Universal Query from the Custom Setup dialog (see Section 4.4 Universal Products for Windows – Installation).</p> <p>This optional parameter is required only if you do not want to install Universal Query.</p> <p>It is valid only when the /i switch is used. Otherwise, it is ignored.</p> <p>[Default is yes.]</p>

Parameter	Description
USPOOL	<p>Prevents Universal Spool utilities from being installed (if value is <i>no</i>). This has the same effect as choosing not to install the Universal Spool utilities from the Custom Setup dialog (see Section 4.4 Universal Products for Windows – Installation).</p> <p>This optional parameter is required only if you do not want to install the Universal Spool Utilities.</p> <p>It is valid only when the /i switch is used. Otherwise, it is ignored. [Default is yes.]</p>
/q	<p>Instructs Windows Installer to run without a graphical interface (silent install).</p> <p>If this optional parameter is omitted, it allows the Universal Products installation to be started from the command line, but run with a graphical interface. This is useful when an installation log file is desired.</p> <p>See Windows Installer Command Line Syntax, Windows Installer Installation Commands, and Detecting the Completion of a Silent Install in this section for additional information regarding silent installs.</p>
/L*v	<p>Instructs Windows Installer to create an installation log file with a name of <i>Logfilepath</i>. If <i>Logfilepath</i> contains spaces, you must use double (") quotation marks around the path name.</p> <p>The *v after the /L actually are flags used to specify the level of detail contained in the log file. To reduce the amount of output generated, these options can be omitted. However, if a problem occurs during installation, Stonebranch Inc. Customer Support may request a log file, in which case these options should be used.</p> <p>This is an optional parameter.</p>

Table 4.3 Universal Products for Windows - Windows Installer Command Line Parameters

Windows Installer Command Line Syntax

Figure 4.14, below, illustrates the command line syntax when using Windows Installer to install Universal Products from the command line.

```
msiexec.exe [/i|/fom|/x] SetupPath\Ucmd.msi
          [INSTALLDIR=InstallDir]
          [UCMDSRVWORKFOLDER=workFolderPath]
          [UCTLSRVWORKFOLDER=workFolderPath]
          [UDMSRVWORKFOLDER=workFolderPath]
          [UEMSRVWORKFOLDER=workFolderPath]
          [UCMDMGR={yes|no}] [UCMDSRV={yes|no}]
          [UCTLMGR={yes|no}] [UCTLSRV={yes|no}]
          [UDMMGR={yes|no}] [UDMSRV={yes|no}]
          [UELD={yes|no}]
          [UEMMGR={yes|no}] [UEMSRV={yes|no}]
          [UQUERY={yes|no}] [USPOOL={yes|no}]
          [/q] [/L*v LogfilePath]
```

Figure 4.14 Universal Products for Windows - Windows Installer Command Line Syntax

The order in which the parameters are specified is significant.

The **/i**, **/f**, or **/x** parameter must be first, followed by the name of the installation file, **Ucmd.msi**.

The parameters that set the installation directory, set the working folders, and control the installation of individual Universal Products components can be specified in any order. These parameters are used only when the install switch (**/i**) is used. However, because they are used to set properties within the installation script, they must be specified before the **/q** and **/L** parameters.

The **/q** and **/L** switches can be specified in any order, but, when used, must come after all other parameters.

Windows Installer Installation Commands

The following commands allow installation of Universal Products from the command line using Windows Installer (`msiexec.exe`).

To install all Universal Products components using the Windows Installer graphical interface, issue the following command:

```
msiexec.exe /i SetupPath\Ucmd.msi
```

To install all Universal Products components using the Windows Installer graphical interface, and instruct Windows Installer to write a log file to `C:\Temp\install.log` while performing the installation, issue the following command:

```
msiexec.exe /i SetupPath\Ucmd.msi /L C:\Temp\install.log
```

To install only the Universal Broker (which always is installed), Universal Command Manager, Universal Control Manager, and Universal Query components under `D:\Universal` (that is, a directory other than the one specified by the environment variable `PROGRAMFILES`) using the Windows Installer graphical interface, issue the following command:

```
msiexec.exe /i SetupPath\Ucmd.msi INSTALLDIR=D:\Universal UCMSRV=no  
UCTLSRV=no UELD=no USPOOL=no UDMGR=no UDMSRV=no UEMMGR=no UEMSRV=no /q
```

To install all Universal Products components using the Windows Installer without a graphical interface — that is, a silent install — issue the following command:

```
msiexec.exe /i SetupPath\Ucmd.msi /q
```

Detecting the Completion of a Silent Install

When the `/q` switch is used to perform a silent install, no graphical interface or user interaction is required. One drawback to this is that no feedback is provided indicating when the Windows Installer process (install, uninstall, or repair) finishes.

One method that can be used to detect when the Windows Installer process ends is to execute it using the system's `start` command. Using available command line switches, `start` can be used to initiate the Windows Installer process and then wait for it to finish. When `start` returns control to its calling process (for example, the command prompt), the process will have ended.

For example, from the command prompt, issue the following command to start the Universal Products installation and wait for it to finish.

```
start /b /wait msiexec.exe /i SetupPath\Ucmd.msi /q
```

- The `/b` switch prevents the `start` command from opening a new window.
- The `/wait` parameter causes the `start` command to start Windows Installer (`msiexec.exe`) and then wait for it to finish.

This syntax also can be used to execute the `start` command from within a script, such as a `.bat` file.

For more information on the `start` command, go to the Windows command prompt and enter: `start /?`.

Using the Distribution File for a Silent Install

To perform a silent install using the Universal Products for Windows distribution file, issue the following command:

```
unv-4.1.0.0-nt-i386.exe /s /a /s /w /v" /qn"
```

The switches (from left to right) are defined as:

- `/s` "Package for the Web" switch for silent install.
- `/a` "Package for the Web" switch to "add" commands (that is, send the next commands to `setup.exe`).
- `/s` `setup.exe` switch for silent install.
- `/w` Wait switch, for a Basic MSI project, forces `setup.exe` to wait until the installation is complete before exiting.
- `/v` `setup.exe` switch to pass arguments or options to the MSI package. Quotes must be in place if a space-separated command is passed to the MSI.
- `/qn` `msiexec.exe` switch for quiet install with no graphic display of progress.

4.4.4 Modifying a Universal Products Installation via the Windows Installer Command Line Interface

This section describes how to modify a Universal Products installation via the Windows Installer command line interface.

After Universal Products is installed, Windows Installer can be run as many times as needed to modify the installation by:

- [Adding or Removing Universal Products Components](#)
- [Repairing a Corrupted Universal Products Installation](#)
- [Removing a Universal Products Installation](#)

(For a description of the parameters used in these procedures, see [Table 4.3 Universal Products for Windows - Windows Installer Command Line Parameters](#).)

Adding or Removing Universal Products Components

Currently, it only is possible to add or remove individual Universal Products components using the Windows Installer graphical interface (see [Adding or Removing Universal Products Components](#) in Section 4.4.2 [Modifying a Universal Products Installation via the Windows Installer Graphical Interface](#)).

Repairing a Corrupted Universal Products Installation

To recover accidentally deleted files or registry entries required by Universal Products using the Windows Installer command line interface, use the `/f` switch together with the `om` parameters.

These are the same repair options set internally by the graphical interface installation. They cause Windows Installer to reinstall files that either are missing or older than the version contained in the Universal Products distribution file.

To repair a Universal Products installation from the command line, issue the following command:

```
msiexec.exe /fom SetupPath\Ucmd.msi /q
```

To repair a Universal Products installation using the Windows Installer graphical interface, and instruct Windows Installer to write a log file to `C:\Temp\repair.log` while running the repair, issue the following command:

```
msiexec.exe /fom SetupPath\Ucmd.msi /! C:\Temp\repair.log
```

Removing a Universal Products Installation

To uninstall a Universal Products installation using the Windows Installer command line interface, use the `/x` switch. Issue the following command:

```
msiexec.exe /x SetupPath\Ucmd.msi /q
```

4.5 Universal Products for Windows – Licensing

After Universal Products for Windows has been installed, the following Universal Products components must be configured with product licenses before they can be used; they will not run without valid licenses:

- Universal Command Manager
- Universal Data Mover Manager
- Universal Event Monitor Server

Product license information is provided, in a formatted text file, by your Stonebranch, Inc. account representative. The file contains the license parameter keywords and their values.

4.5.1 Licensing Methods

Universal Products for Windows components can be configured with product licenses either by:

- Entering the information into their configuration file. (The values are specified in the same syntax as all other configuration options.)
- Specifying the information on the Universal Configuration Manager License Information page for that component, either by.
 - Entering the information specified in the license file.
 - Importing the license file.

Note: Universal Configuration Manager is installed during the Universal Products for Windows installation.

4.5.2 Licensing via Universal Configuration Manager

To enter license information via the Universal Configuration Manager, perform the following steps:

1. On the Windows Control Panel, double-click the Universal Configuration Manager icon to display the Universal Configuration Manager screen.
2. In the Installed Components tree, click + next to the component that you want to configure with license information:
3. Click **License Information** to display the License Information page ([Figure 4.15](#), below).

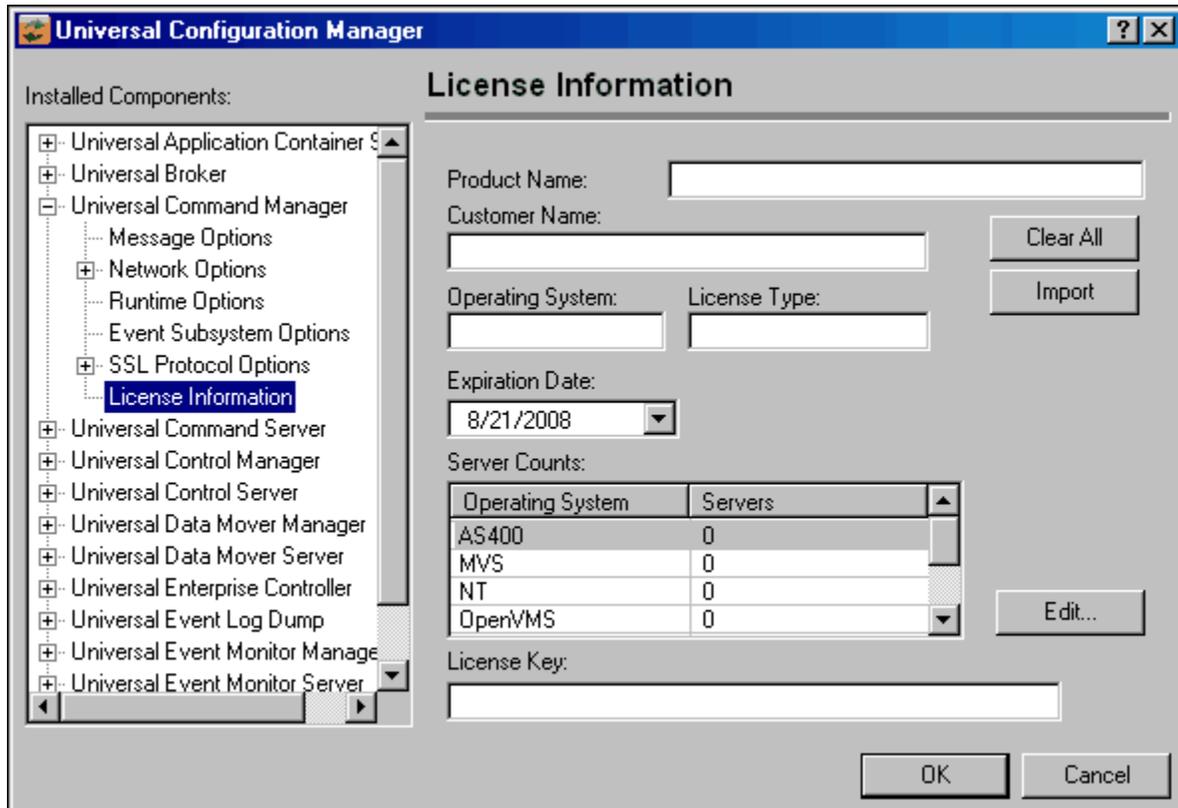


Figure 4.15 Universal Configuration Manager - License Information page

4. If you want to enter the license information on the page, make sure that you enter the information exactly as it is in the license information file. All information is case-sensitive; punctuation and spacing must be maintained.
5. If you want to import the license file:
 - a. Click the **Import** button.
 - b. On the Select License File dialog that displays, select the license file to import.
 - c. Click the **Open** button. The information is imported to the License information page.
6. Click the **OK** button to save the license information.

4.6 Universal Products for Windows – Installing on 64-bit Windows Editions

All Universal Products have been tested and verified on the 64-bit editions of the following Windows systems:

- Windows XP
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows Server 2008 R2

This section describes some modifications that may need to be made to the default installation options to ensure that the installed Universal Products function correctly.

4.6.1 Modifying the Working Folder for Universal Server Components

On 64-bit Windows editions, the default installation folder for 32-bit applications is `C:\Program Files (x86)`. Because all Windows applications developed by Stonebranch, Inc. are 32-bit applications, the default installation folder for the applications we deliver is `C:\Program Files (x86)\Universal`. The Universal Products installation package should have no problems installing its applications to this directory.

However, on 64-bit editions of Windows prior to XP SP2 or Server 2003 SP1, the error **'C:\Program' is not recognized as an internal or external command, operable program or batch file** may be issued in the following circumstances:

- Universal Command Manager is run using the `-script` option.
- Universal Event Monitor Server invokes an event handler that executes a script.

This error is a result of the way that scripts are prepared for execution and the way that the Windows command shell interpreter reads a quoted string containing special characters (that is, `& < > () @ ^ |`). Universal Command and Universal Event Monitor prepare scripts for execution by writing the script statements to a temporary `.bat` file in the application's working directory. A command line statement containing that working directory's path is then constructed to execute the `.bat` file. By default, this path includes the `Program Files (x86)` directory. Because the path includes one of the special characters listed above, the Windows command shell interpreter incorrectly parses the path, which results in the error above.

To resolve this issue, it is necessary to change the value (or location) of the Universal Command Server and/or Universal Event Monitor Server working directory.

Either of the following changes can be made:

- If short path names are permitted on your system, use the `dir /x` command from the command prompt to find the short name of the **Program Files (x86)** directory (for example, **Progra~2**). Use this value as part of the working directory's path name.
- Change the working directory to a location outside of the default installation path (that is, **C:\Program Files (x86)**). This new location can contain spaces, but it cannot contain any of the special characters listed above.

The changes can be made either of two ways:

1. During installation (see [Figure 4.16](#), below).

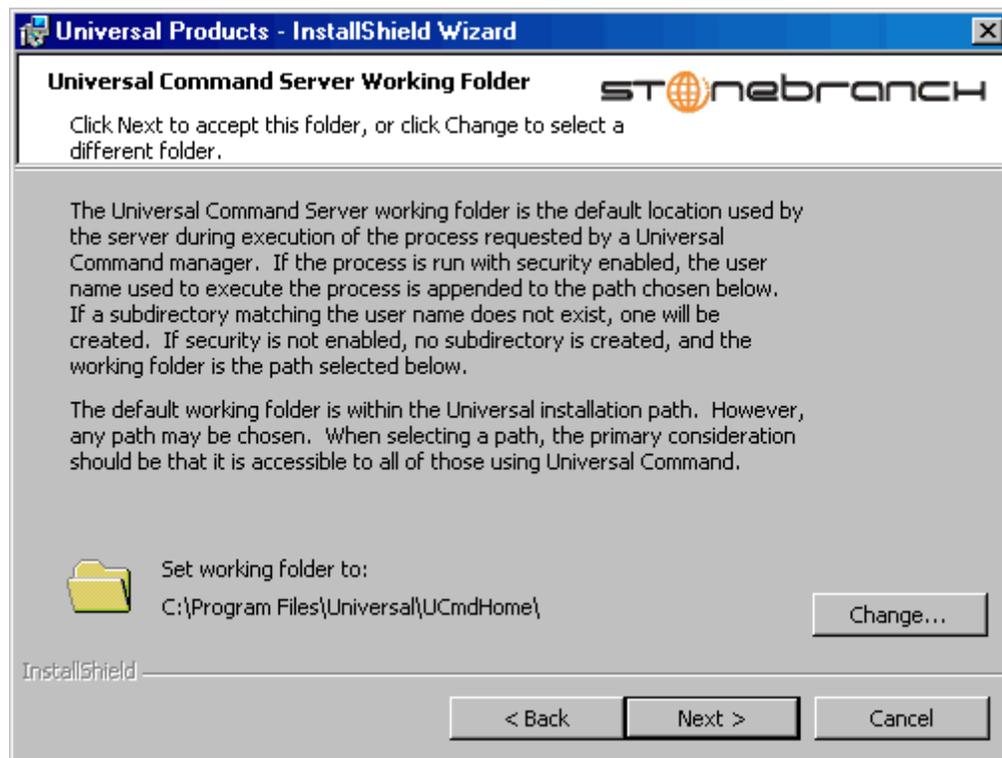


Figure 4.16 Universal Products for Windows - Setting the Working Folder during Installation

- Any time after installation via the Universal Configuration Manager, on the Component Definitions page for Universal Broker (see [Figure 4.17](#), below).

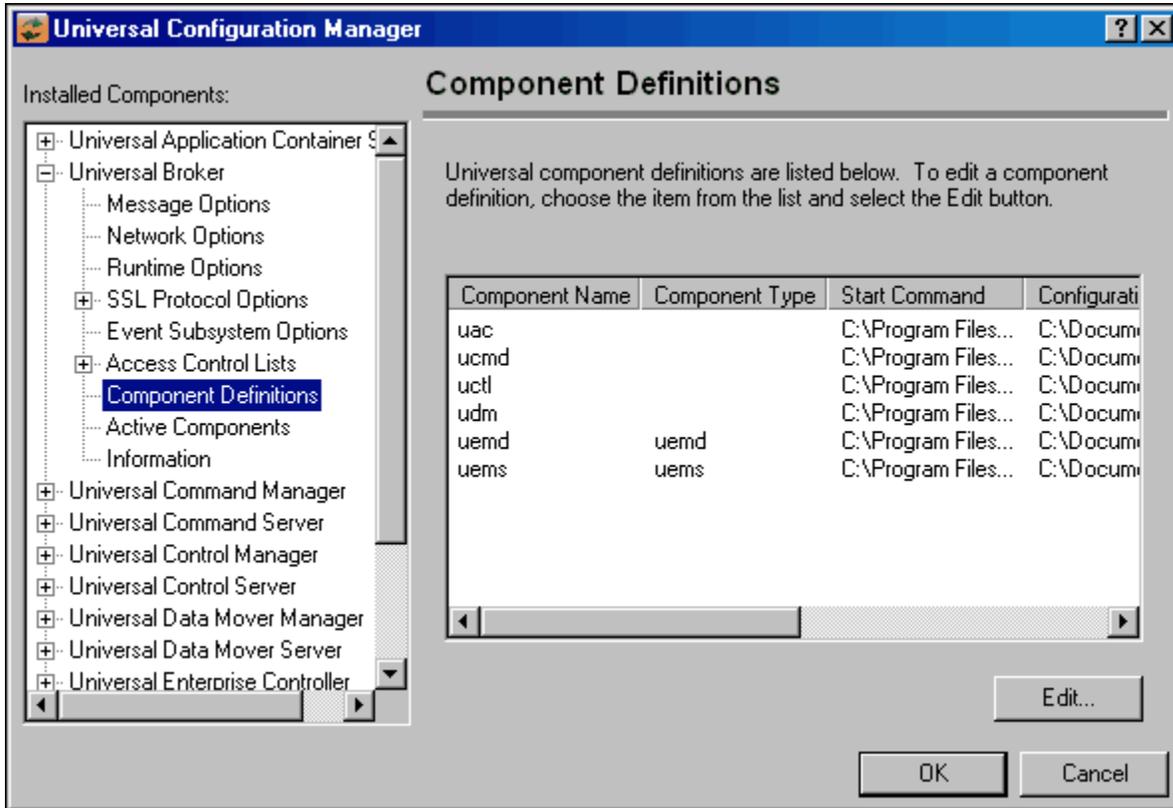


Figure 4.17 Universal Configuration Manager - Universal Broker / Component Definitions

4.6.2 Applications Installed in the Windows System Folder

The Universal Products package installs several command-line applications in the Windows system folder. The default system folder for 32-bit applications installed on 64-bit Windows editions is the %SystemRoot%\SysWOW64 directory (for example, C:\Windows\SysWOW64).

Table 4.4, below, identifies the affected Universal Products applications.

File Name	Description
ucert.exe	Universal Certificate
ucmd.exe	Universal Command Manager
ucopy.exe	Universal Copy
uctl.exe	Universal Control Manager
udm.exe	Universal Data Mover Manager
ueld.exe	Universal Event Log Dump Utility
uem.exe	Universal Event Monitor Manager
uemload.exe	Universal Event Monitor Load Utility
uencrypt.exe	Universal Encrypt Utility
umet.exe	Universal Message Translator
uquery.exe	Universal Query
urc.exe	Universal Return Code Utility

Table 4.4 Universal Products for Windows - Applications Installed in Windows System Folder

These applications can be executed using either the:

- 32-bit command shell (%SystemRoot%\SysWOW64\cmd.exe)
- Default 64-bit command shell (%SystemRoot%\System32\cmd.exe).

By default, the %SystemRoot%\SysWOW64 directory is not part of the system path. Therefore, to execute the above command line applications using the 32-bit command shell, either:

- Directory must be added to the PATH environment variable.
- Complete path to the application and/or the 32-bit command shell must be specified.

Example 1

To execute UCOPY in the default 64-bit command shell, issue the following command:

```
%SystemRoot%\SysWOW64\ucopy
```

Example 2

To execute UCOPY within the 32-bit command shell, use the following:

```
%SystemRoot%\SysWOW64\cmd.exe /C %SystemRoot%\SysWOW64\ucopy
```

4.7 Universal Products for Windows – File Inventory Lists

The Universal Products 4.1.0 for Windows package includes the files required for the following components / utilities:

- Universal Broker
- Universal Command Manager
- Universal Command Server
- Universal Configuration Manager
- Universal Control Manager
- Universal Control Server
- Universal Copy
- Universal Data Mover Manager
- Universal Data Mover Server
- Universal Encrypt
- Universal Event Log Dump
- Universal Event Monitor Manager
- Universal Event Monitor Server
- Universal Products Install Merge
- Universal Query
- Universal Spool Utilities

Universal Broker and Universal Configuration Manager always are installed. Other components are installed as desired. If any of the components already are installed, Windows Installer will upgrade them to the latest version.

This section lists the files installed with each Universal Products component. The file paths specified are relative to the root installation directory (for example, **C:\Program Files\Universal**) that was specified during the installation.

Items shown with a path of System32 are installed in the 32-bit system folder. The actual name of this directory depends on the Windows version:

- For Windows 2000, the path is **\Winnt\System32**.
- For all other supported 32-bit Windows editions, the path is **\Windows\System32**.
- For all supported 64-bit Windows editions, the path is **\Windows\SysWow64**.

Universal Broker

File	Description.
UBroker\bin\ubroker.exe	Console application program.
UBroker\bin\ubrsvc.exe	Windows service program.
UBroker\bin\ubrcfg.dll	Used by Universal Configuration Manager to manage Broker properties.
UBroker\bin\ubrbscfg.dll	Used by the Universal Configuration Manager to manage bootstrap options when the local Broker is running in managed mode.
UBroker\bin\ubrbscfg.hlp	Universal Broker bootstrap configuration help file.
UBroker\bin\ubrcfg.hlp	Universal Broker configuration help file.
UBroker\bin\ubrdbrec.bat	Recovers Universal Products database files.
UBroker\tml	XML template files used by I-Management Console for remotely configuring Universal Products.
nls*.utt	Code page files used for text translation between different operating systems and platforms.
nls\ussmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\conf\uac1.conf	Universal Access Control List (ACL) configuration file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.
%ALLUSERSPROFILE%\Application Data\Universal\conf\ubroker.conf	Universal Broker configuration file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.

Table 4.5 Universal Broker for Windows - File Inventory List

Universal Command Manager

File	Description
System32\ucmd.exe	Application program.
UCmdMgr\bin\ucmccfg.dll	Used by Universal Configuration Manager to manage Universal Command Manager properties.
UCmdMgr\bin\ucmccfg.hlp	Universal Command Manager configuration help file.
nls*.utt	Code page files used for text translation between different operating systems and platforms.
nls\ucmmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\conf\ucmd.conf	Universal Command Manager configuration file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.

Table 4.6 Universal Command Manager for Windows - File Inventory List

Universal Command Server

File	Description
UCmdSrv\bin\ucmsrv.exe	Application program.
UCmdSrv\bin\ucmscfg.dll	Used by Universal Configuration Manager to manage Universal Command Server properties.
UCmdSrv\bin\ucmscfg.hlp	Universal Command Server configuration help file.
System32\urc.exe	Universal Return Code utility.
nls*.utt	Code page files used for text translation between different operating systems and platforms.
nls\ucmmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\comp\ucmd	Universal Command Server component definition file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.
%ALLUSERSPROFILE%\Application Data\Universal\conf\ucmds.conf	Universal Command Server configuration file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.

Table 4.7 Universal Command Server for Windows - File Inventory List

Universal Control Manager

File	Description
System32\uct1.exe	Application program.
UCt1Mgr\bin\uctccfg.dll	Used by Universal Configuration Manager to manage Universal Control Manager properties.
UCt1Mgr\bin\uctccfg.hlp	Universal Control Manager configuration help file.
nls*.utt	Code page files used for text translation between different operating systems and platforms.
nls\uctmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\conf\uct1.conf	Universal Control Manager configuration file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.

Table 4.8 Universal Control Manager for Windows - File Inventory List

Universal Control Server

File	Description
UCt1Srv\bin\uctsrv.exe	Application program.
UCt1Srv\bin\uctscfg.dll	Used by Universal Configuration Manager to manage Universal Control Server properties.
UCt1Srv\bin\uctscfg.hlp	Universal Control Server configuration help file.
Nls*.utt	Code page files used for text translation between different operating systems and platforms.
Nls\uctmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\comp\uct1	Universal Control Server component definition file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.
%ALLUSERSPROFILE%\Application Data\Universal\conf\uct1s.conf	Universal Control Server configuration file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.

Table 4.9 Universal Control Server for Windows - File Inventory List

Universal Data Mover Manager

File	Description
System32\udm.exe	Application program.
UdmMgr\bin\udmccfg.dll	Used by Universal Configuration Manager to manage Universal Data Mover Manager properties.
UdmMgr\bin\udmccfg.hlp	Universal Data Mover Manager configuration help file.
nls*.utt	Code page files used for text translation between different operating systems and platforms.
nls\udmmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\conf\udm.conf	Universal Data Mover Manager configuration file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.

Table 4.10 Universal Data Mover Manager for Windows - File Inventory List

Universal Data Mover Server

File	Description
UdmSrv\bin\udmsrv.exe	Application program.
UdmSrv\bin\udmscfg.dll	Used by Universal Configuration Manager to manage Universal Data Mover Server properties.
UdmSrv\bin\udmscfg.hlp	Universal Data Mover Server configuration help file.
nls*.utt	Code page files used for text translation between different operating systems and platforms.
nls\udmmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\comp\udm	Universal Data Mover Server component definition file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.
%ALLUSERSPROFILE%\Application Data\Universal\conf\udms.conf	Universal Data Mover Server configuration file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.

Table 4.11 Universal Data Mover Server for Windows - File Inventory List

Universal Event Monitor Manager

File	Description
System32\uem.exe	Application program.
UemMgr\bin\uemccfg.dll	Used by Universal Configuration Manager to manage Universal Event Monitor Manager properties.
UemMgr\bin\uemccfg.hlp	Universal Event Monitor Manager configuration help file.
nls*.utt	Code page files used for text translation between different operating systems and platforms.
nls\uemmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\conf\uem.conf	Universal Event Monitor Manager configuration file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.

Table 4.12 Universal Event Monitor Manager for Windows - File Inventory List

Universal Event Monitor Server

File	Description
System32\uemload.exe	Event definition and event handler database load utility.
UemSrv\bin\uemsrv.exe	Application program.
UemSrv\bin\uemscfg.dll	Used by Universal Configuration Manager to manage Universal Event Monitor Server properties.
UemSrv\bin\uemscfg.hlp	Universal Event Monitor Server configuration help file.
nls*.utt	Code page files used for text translation between different operating systems and platforms.
nls\uemmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\comp\uemd	Component definition file for the Universal Event Monitor Demand-Driven Server. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.
%ALLUSERSPROFILE%\Application Data\Universal\comp\uems	Component definition file for the Universal Event Monitor Event-Driven Server. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.
%ALLUSERSPROFILE%\Application Data\Universal\conf\uems.conf	Universal Event Monitor Server configuration file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.

Table 4.13 Universal Event Monitor Server for Windows - File Inventory List

Universal Configuration Manager

File	Description
UCfgMgr\bin\ucfgmgr.cpl	Universal Configuration Manager control panel application.
UCfgMgr\bin\ucfgmgr.cpl	Universal Configuration Manager help file.

Table 4.14 Universal Configuration Manager for Windows - File Inventory List

Universal Copy

File	Description
System32\ucopy.exe	Utility used for binary file copies. Similar to the UNIX cat command. Installed only if Universal Command Server is installed.

Table 4.15 Universal Copy for Windows - File Inventory List

Universal Encrypt

File	Description
System32\uecrypt.exe	Application program. Installed only if Universal Command Manager is installed.

Table 4.16 Universal Encrypt for Windows - File Inventory List

Universal Event Log Dump

File	Description
System32\ue1d.exe	Application program.
UE1d\bin\ue1cfg.dll	Used by Universal Configuration Manager to manage Universal Event Log Dump properties.
UE1d\bin\ue1cfg.hlp	Universal Event Log Dump configuration help file.
N1s\ue1mceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\conf\ue1d.conf	Universal Event Log Dump configuration file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.

Table 4.17 Universal Event Log Dump for Windows - File Inventory List

Universal Message Translator

File	Description
System32\umet.exe	Application program, always installed. Used to map application-specific error messages to error codes.

Table 4.18 Universal Message Translator for Windows - File Inventory List

Universal Products Install Merge

File	Description
UPIMerge\bin\upimerge.exe	Application program, always installed. Provides command line access to the same functionality used by the Universal Products installation to merge options from a new configuration file into an existing file.

Table 4.19 Universal Products Install Merge for Windows - File Inventory List

Universal Query

File	Description
System32\uquery.exe	Application program.
UQuery\bin\uqrcfg.dll	Used by Universal Configuration Manager to manage Universal Query properties.
UQuery\bin\uqrcfg.hlp	Universal Query configuration help file.
Nls*.utt	Code page files used for text translation between different operating systems and platforms.
Nls\uqrmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\conf\uquery.conf	Universal Query configuration file. %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically C:\Documents and Settings\All Users.

Table 4.20 Universal Query for Windows - File Inventory List

Universal Spool Utilities

File	Description
USpool\bin\uslist.exe	Used to list the contents of Universal Spool files.
USpool\bin\uslrm.exe	Used to remove records from Universal Spool files.
USpool\bin\udb_archive.exe USpool\bin\udb_dump.exe USpool\bin\udb_load.exe USpool\bin\udb_recover.exe USpool\bin\udb_stat.exe USpool\bin\udb_upgrade.exe USpool\bin\udb_verify.exe	Miscellaneous spool file utilities. Should be used only for debugging purposes, and only at the request of Stonebranch, Inc. Customer Support .

Table 4.21 Universal Spool Utilities for Windows - File Inventory List

System Files

The following files will be installed only if they are newer than the existing file.

The directories shown in [Table 4.22](#), below, are relative to the %SYSTEMROOT% directory, where %SYSTEMROOT% is an environment variable that resolves to C:\WINNT on Windows 2000 and C:\Windows on all other Windows platforms.

File	Description
System32\asycfilt.dll	Version 2.40.4275.1. This DLL is one of the components of the Microsoft OLE library.
System32\comcat.dll	Version 4.71.1460.1 of the Microsoft Component Category Manager library.
Microsoft C-Runtime v8.0.50727.762 ¹	Version 8.0.50727.762 of the Microsoft C runtime side-by-side assembly.
Microsoft Foundation Classes v8.0.50727.762 ²	Version 8.0.50727.762 of the Microsoft Foundation Class (MFC) side-by-side assembly.
System32\msiexec.exe	Version 3.1.4000.1823 of the Microsoft Windows Installer (see Section 4.4 Universal Products for Windows – Installation for more information).
System32\oleaut32.dll	Version 2.40.4275.1. This DLL is one of the components of the Microsoft OLE library.
System32\olepro32.dll	Version 5.0.4275.1. This DLL is one of the components of the Microsoft OLE library.
System32\psapi.dll	Version 4.0.1371.1 of the Microsoft process status library
System32\stdole2.tlb	Version 2.40.4275.1. This file is one of the components of the Microsoft OLE library.
1	The Microsoft C-Runtime distribution consists of several files, which are subject to change. Refer to Microsoft documentation for a complete list of files delivered with the specified runtime version.
2	The Microsoft Foundation Classes (MFC) distribution consists of several files, which are subject to change. Refer to Microsoft documentation for a complete list of files delivered with the specified MFC version.

Table 4.22 Universal Products for Windows System Files - File Inventory List

4.8 Universal Enterprise Controller Package

The Universal Enterprise Controller (UEC) 4.1.0 for Windows package includes a single product component:

- Universal Enterprise Controller 4.1.0

Note: Installations of Universal Enterprise Controller v110-3 and earlier also included desktop application versions of the Universal Enterprise Controller Client Applications: I-Administrator, I-Activity Monitor, and I-Management Console.

As of v110-4, these client applications are contained in their own, separate installation package (see [Section 4.14 Universal Enterprise Controller Client Applications Package](#)).

4.8.1 Product Compatibility

[Table 4.23](#), below, identifies the compatibility of Universal Enterprise Controller 4.1.0 for Windows with previous product versions.

Product	Compatibility
Universal Enterprise Controller 4.1.0	<ul style="list-style-type: none"> • Universal Command Server 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0 • UEC Client Applications 4.1.0 and 3.2.0
<p>Note: Universal Enterprise Controller 3.1.1 versions are compatible only with 3.1.1 versions of the UEC Client Applications.</p>	

Table 4.23 Universal Enterprise Controller for Windows - Product Compatibility

4.9 Universal Enterprise Controller – Installation Requirements

To install the Universal Enterprise Controller, you must have one of the following versions of Windows:

- Windows 2000 Professional or Server with Service Pack 4 or higher
- Windows Server 2003
- Windows XP
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows Server 2008 R2

In addition, you must have:

- Approximately 35 megabytes of disk space for the installation.
- An account with administrative privileges.

4.10 Universal Enterprise Controller – Installation

There are two Windows Installer interfaces that can be used for installing Universal Enterprise Controller: graphical and command line.

This section provides the following procedures regarding these two interfaces:

- [Installing Universal Enterprise Controller via the Windows Installer Graphical Interface](#)
- [Modifying a Universal Enterprise Controller Installation via the Windows Installer Graphical Interface](#)
- [Installing Universal Enterprise Controller via the Windows Installer Command Line Interface](#)
- [Modifying a Universal Enterprise Controller Installation via the Windows Installer Command Line Interface](#)

4.10.1 Installing Universal Enterprise Controller via the Windows Installer Graphical Interface

To install Universal Enterprise Controller using the Windows Installer graphical interface, perform the following steps:

1. Download the Universal Enterprise Controller for Windows product distribution file, `UECt1r-4.1.0.<level>-nt-i386.exe`, to your work station.
2. Execute the distribution file to extract the files.

Note: If you already have extracted the files from the distribution file, but cancelled installation in order to separately install Windows Installer (see Section [4.1.3 Windows Installer](#)), you can simply double-click the extracted Universal Enterprise Controller installation file, `UECt1r.msi`, to begin the installation.

Installing over a Remote Desktop Session

Starting with Windows Server 2003, Remote Desktop provides distinct session environments for each logged-in user. This means that extraction may use an environment setting that is not available once the Remote Desktop session ends.

For example, the distribution file's default extraction location is based on the **TEMP** environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed.

To ensure that extracted files and other required resources are accessible after the initial install, extract the files to a well-known location that is not likely to change between Remote Desktop sessions.

Refer to the Microsoft documentation on the Remote Desktop feature for additional information.

The installation starts after the files are extracted.

It first will verify that your machine meets the minimum system requirements (see Section [4.9 Universal Enterprise Controller – Installation Requirements](#)). If the requirements are met, a Welcome dialog displays ([Figure 4.18](#)).

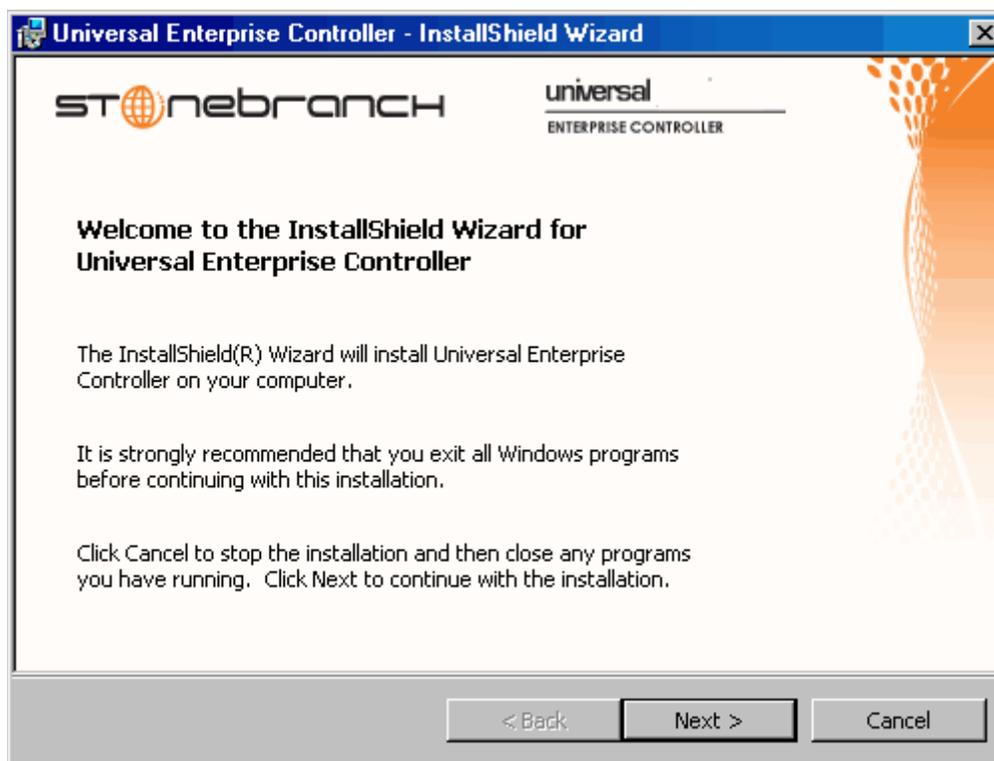


Figure 4.18 Universal Enterprise Controller - Welcome dialog

3. Click the **Next** button. A list of Universal Enterprise Controller components included in the installation package then displays (Figure 4.19). It is from this list that you can select which components to install.

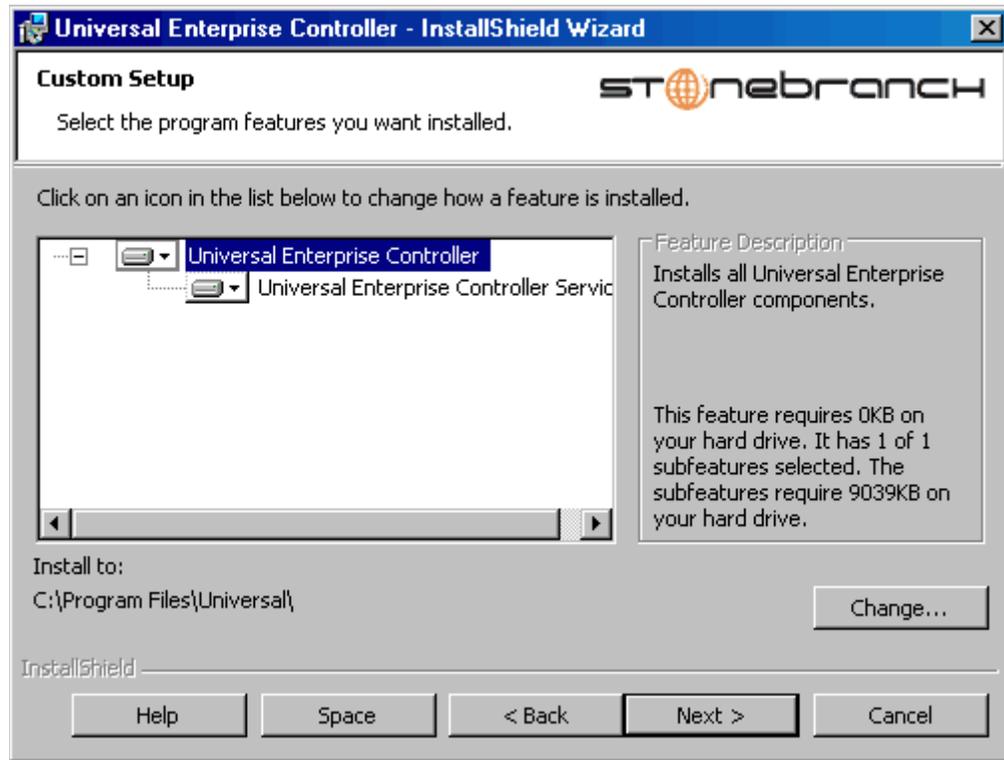


Figure 4.19 Universal Enterprise Controller - Custom Setup dialog

For a new installation, a drive icon  displays next to each item in the list, indicating that the component will be installed.

For an upgrade installation, either of the following icons displays next to an item:

- A drive icon  indicates that the component is either:
 - New to the installation and will be installed.
 - Currently is installed and will be upgraded.
- An **X** icon indicates that the component is either:
 - Currently not installed (but previously was available).
 - Previously installed but removed.



Stonemason's Tip

If the installation detects an existing Universal Enterprise Controller installation, currently installed components may be upgraded.

(Currently, there is no way to specify that the state of a currently installed component remain unchanged.)

If a component is selected for installation, and the version of the installed component is below the version of the component being installed, the installed component will be replaced by the component being installed.

If a component is not selected for installation (that is, the **X** icon is selected), and it currently is installed, the new installation will remove that component.

4. [Figure 4.19](#) shows that all Universal Enterprise Controller components will be installed in their respective directories under the C:\Program Files\Universal directory.
 - a. If you want to select a different location, click the **Change...** button.
 - b. If you want to check the amount of disk space required for the installation, and the amount of available disk space on the selected directory, click the **Space** button.
5. If you do not want to install a component:
 - a. Click the drive icon  next to that component name.
 - b. From the drop-down list that displays, select the **X** icon to mark the component as one not to be installed.

For example, [Figure 4.20](#), below, indicates that the Universal Enterprise Controller Service has been selected to not be installed.

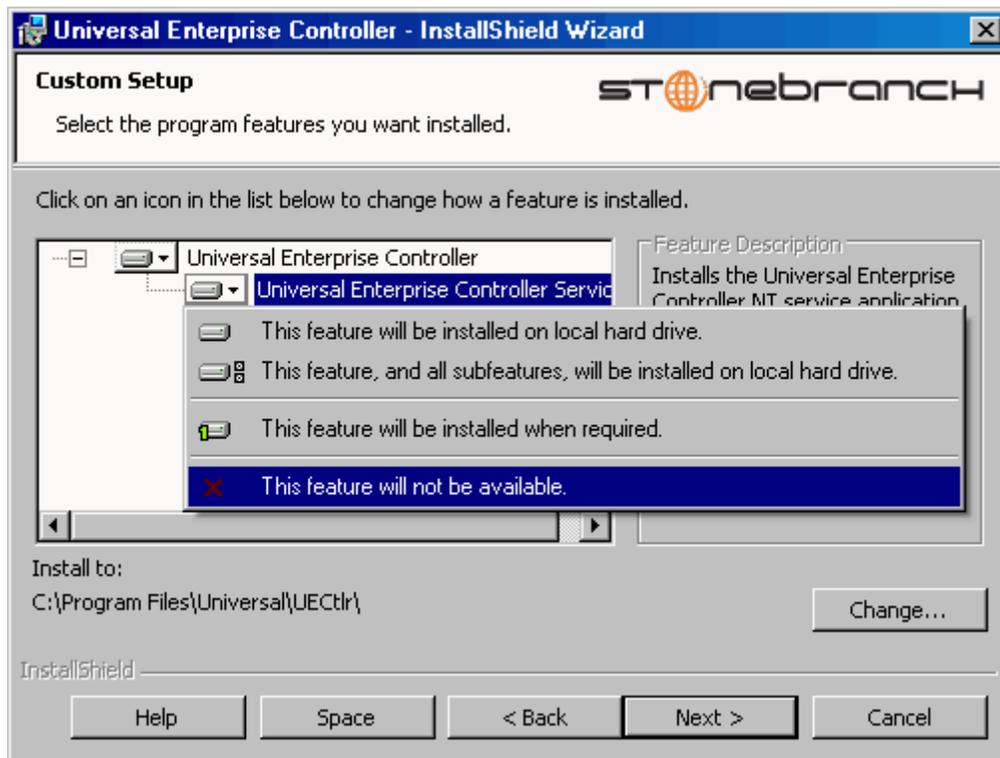


Figure 4.20 Universal Enterprise Controller - Custom Setup / Selecting to not Install a Component

6. When you have selected the components (and their installation destinations) that you want to install, click the **Next>** button to continue the installation process.

When the installation completes successfully, the Installation Complete dialog displays ([Figure 4.21](#), below).

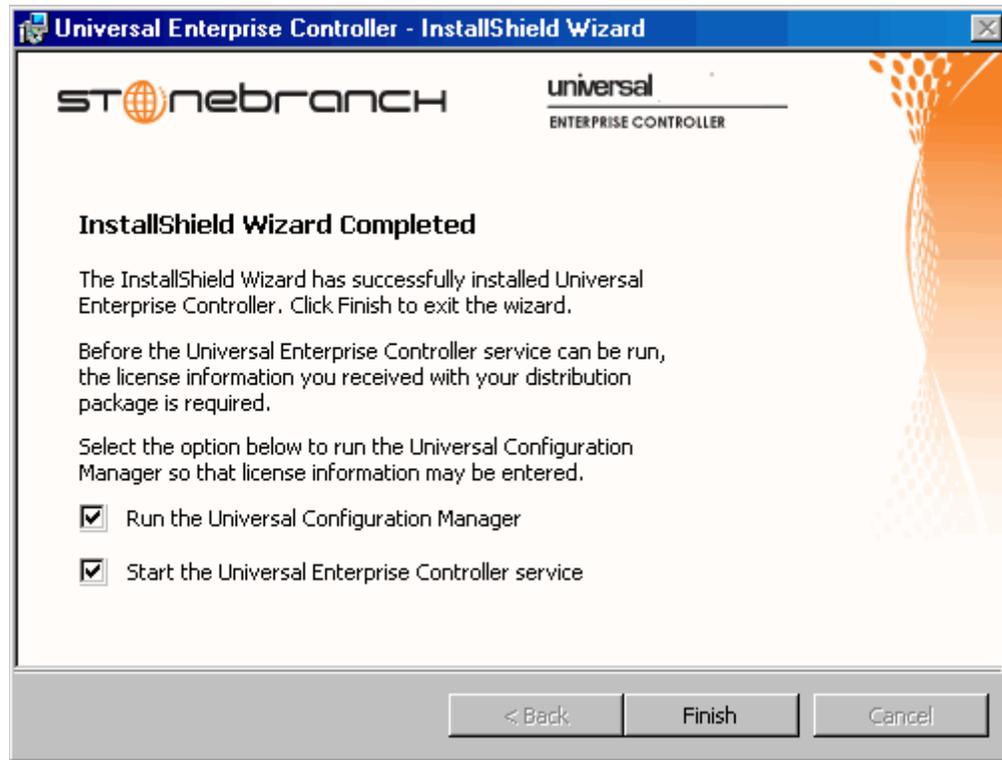


Figure 4.21 Universal Enterprise Controller - Installation Complete

7. If the Universal Enterprise Controller service was installed, the following options display on this dialog:
 - **Run the Universal Configuration Manager**
 - **Start the Universal Enterprise Controller service**

License information must be entered into the Universal Enterprise Controller service's configuration before the service application will run.

Select both of these options to enter the license information and then start the Universal Enterprise Controller service.

8. Click the **Finish** button to exit Windows Installation.

4.10.2 Modifying a Universal Enterprise Controller Installation via the Windows Installer Graphical Interface

After Universal Enterprise Controller is installed, Windows Installer can be run as many times as needed to modify the installation.

These installation modifications are:

- [Adding or Removing Universal Enterprise Controller Components](#)
- [Repairing a Corrupted Universal Enterprise Controller Installation](#)
- [Removing a Universal Enterprise Controller Installation](#)

Adding or Removing Universal Enterprise Controller Components

To add or remove components from a Universal Enterprise Controller installation, perform the following steps:

1. On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays (Figure 4.22, below).

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

The Programs and Features dialog (see Figure 4.7) replaces the Add or Remove Programs dialog.

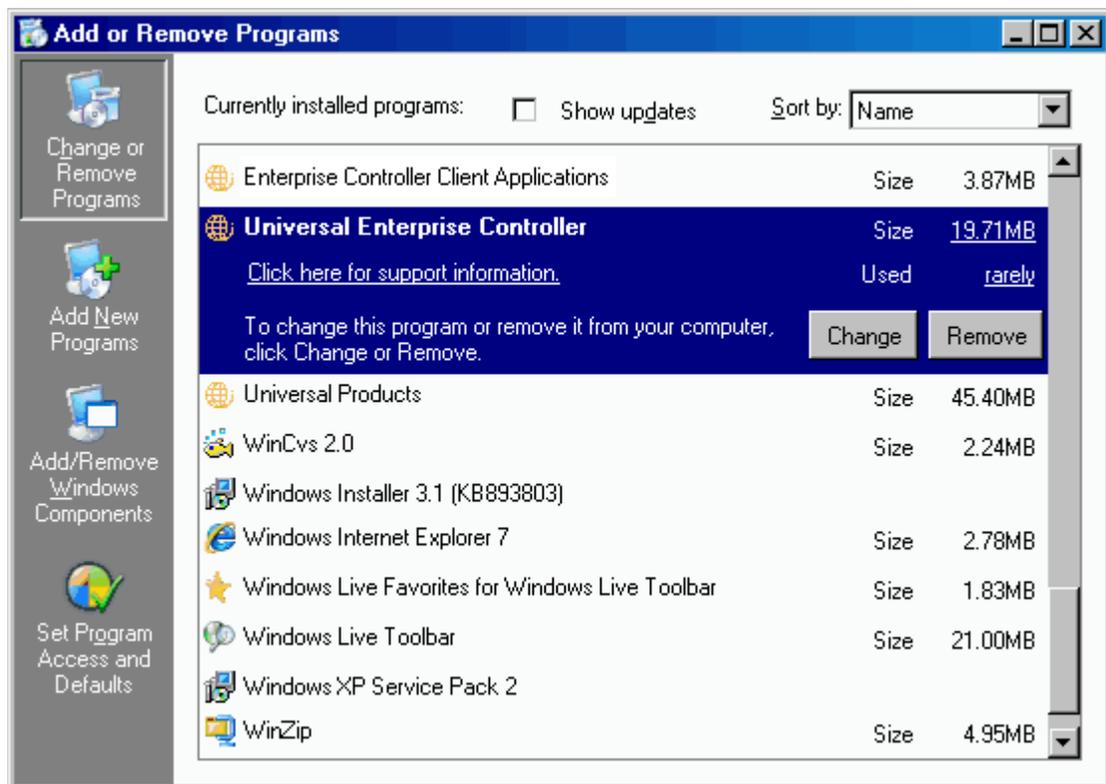


Figure 4.22 Universal Enterprise Controller - Add or Remove Programs dialog

2. From the list of installed programs, select **Universal Enterprise Controller**.
3. Click the **Change** button to start Windows Installer.
4. On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays (Figure 4.24).

Installing over a Remote Desktop Session

If Universal Enterprise Controller was installed via a Remote Desktop connection, the environment settings used during that session may no longer be available. Starting with Windows Server 2003, Remote Desktop provides distinct session environments for each logged-in user.

The distribution file's default extraction location is based on the **TEMP** environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed. Consequently, any attempts to modify the installation may fail because the Windows Installer can't locate the installation's source files (a dialog similar to the one shown in Figure 4.23, below, may be displayed).

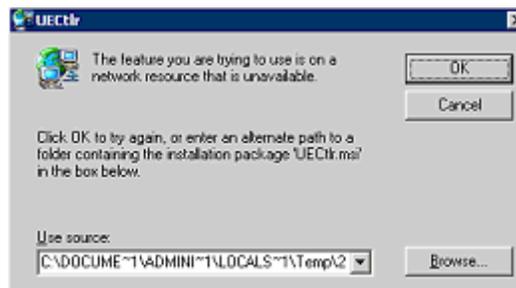


Figure 4.23 Failure to Access Install Files

To resolve this issue, re-extract the distribution files to a location that is independent of a Remote Desktop environment and specify that location in the dialog above. Keep in mind, however, that the extracted files must come from the same distribution package used to do the initial install. If matching distribution files can't be found, Universal Enterprise Controller must be uninstalled and then reinstalled with the desired modifications.

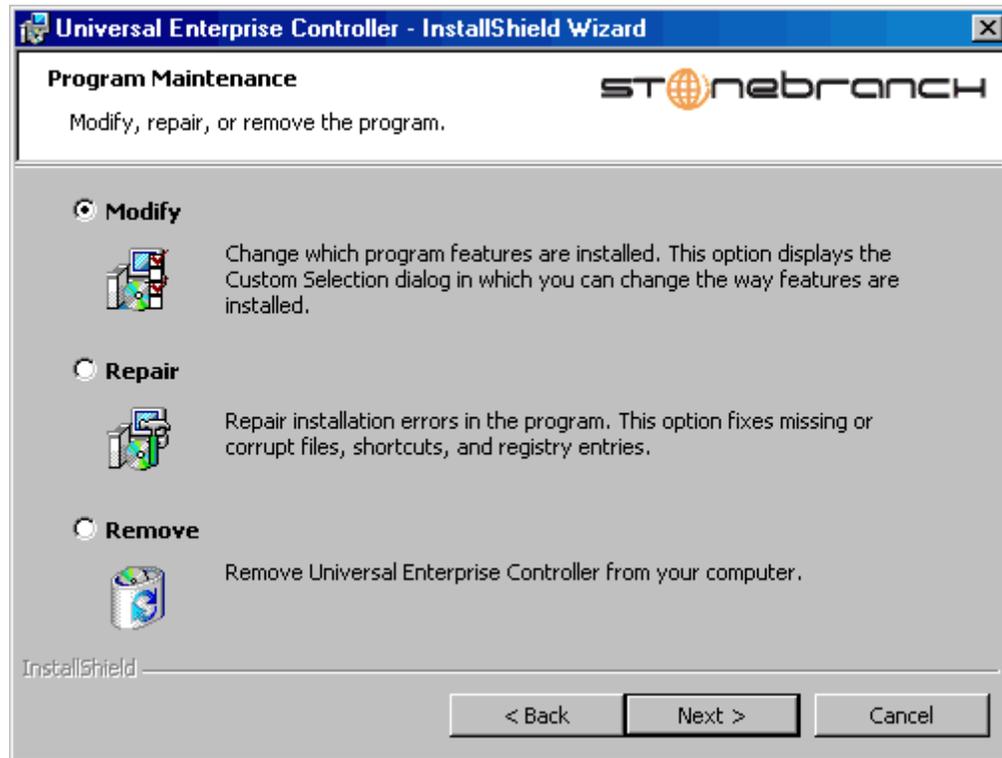


Figure 4.24 Universal Enterprise Controller - Program Maintenance / Modify dialog

5. Click the **Modify** radio button, and then the **Next>** button, to display the Custom Setup dialog (see [Figure 4.25](#)).

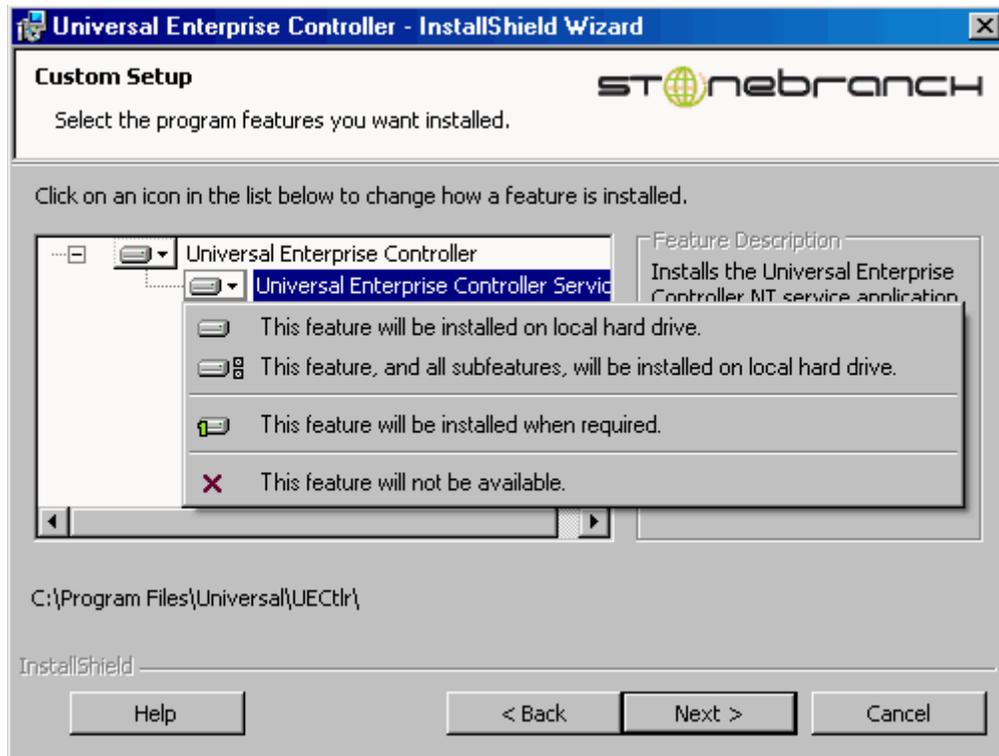


Figure 4.25 Universal Enterprise Controller - Custom Setup / Modifying an Installed Component

Currently installed components are identified by a drive icon .

Uninstalled components are identified by an **X** icon.

6. To remove a currently installed component:
 - a. Click the drive icon  next to that component.
 - b. Select the **X** icon from the drop-down list to mark the component for removal.
7. To add an uninstalled component:
 - a. Click the **X** icon next to that component.
 - b. Select the drive icon  from the drop-down list to mark the component for installation.
8. Click the **Next>** button to continue with the modification.

When the modifications are complete, the following actions will be taken:

- Components marked with a drive icon  will:
 - Remain installed if they already are installed.
 - Be installed if they are not already installed.
- Components marked with an **X** will:
 - Remain uninstalled if they are not currently installed
 - Be removed if they currently are installed.

Repairing a Corrupted Universal Enterprise Controller Installation

Windows Installer has the ability to recover accidentally deleted application files, configuration files, or registry entries required by Universal Enterprise Controller. This repair feature will re-install the missing items, making a complete re-install unnecessary.

During a repair, any options stored in the Universal Enterprise Controller configuration file is preserved. If the UEC configuration file was deleted, the installation will create a new configuration file with default values.

To repair an installation, perform the following steps:

1. On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays (see [Figure 4.22](#)).

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

The Programs and Features dialog (see [Figure 4.7](#)) replaces the Add or Remove Programs dialog.

2. From the list of installed programs, select **Universal Enterprise Controller**.
3. Click the **Change** button to start Windows Installer.
4. On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays ([Figure 4.26](#), below).

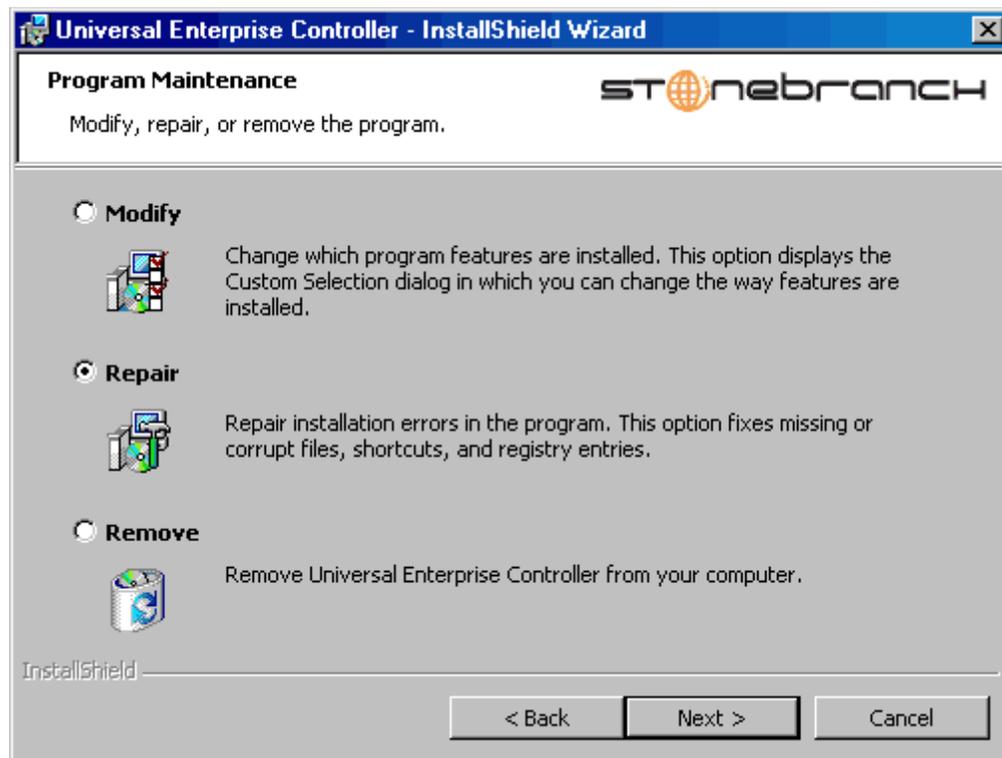


Figure 4.26 Universal Enterprise Controller - Program Maintenance / Repair dialog

5. Click the **Repair** radio button, then **Next>**, to display the Ready to Repair dialog.
6. Follow the instructions displayed in successive dialogs to complete the repair.

Removing a Universal Enterprise Controller Installation

To uninstall a Universal Enterprise Controller installation, perform the following steps:

1. On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays (see [Figure 4.22](#)).

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

The Programs and Features dialog (see [Figure 4.7](#)) replaces the Add or Remove Programs dialog.

2. From the list of installed programs, select **Universal Enterprise Controller**.

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

To skip the remaining steps, simply select the **Uninstall** button (see [Figure 4.7](#)).

3. Click the **Change** button to start Windows Installer.
4. On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays ([Figure 4.27](#), below).

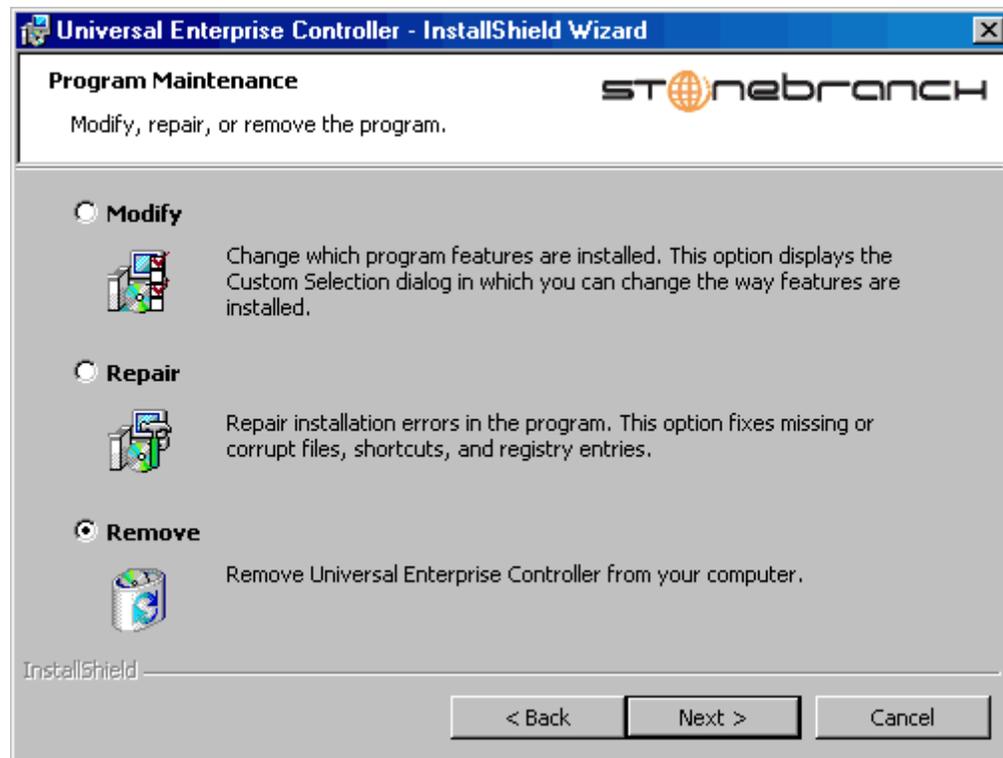


Figure 4.27 Universal Enterprise Controller - Program Maintenance / Remove dialog

5. Select **Remove** and click the **Next>** button.
6. On the subsequent dialog, click the **Remove** button to uninstall the installation.

Un-Installed Files

The un-install process will remove only those files created during the installation. Some files stored under the `.\Universal1` install directory by Universal Enterprise Controller, such as trace files, may be left behind after the uninstall. In this situation, those files and/or directories may simply be deleted.

Before deleting the entire `.\Universal1` directory, make sure that no other Stonebranch, Inc. products are installed there. (See Section [4.13 Universal Enterprise Controller – File Inventory Lists](#) for a list of files and directories installed with Universal Enterprise Controller.)

In addition to those files and directories created by the Universal Enterprise Controller installation, there may be some shared files left behind following an uninstall. These components will be removed when the last Stonebranch Inc. product that uses them is uninstalled.

4.10.3 Installing Universal Enterprise Controller via the Windows Installer Command Line Interface

This section describes how to install Universal Enterprise Controller using the Windows Installer command line interface.

The command line interface is useful in situations where:

1. Several Universal Enterprise Controller installations must be deployed.
2. It is not practical or convenient to perform the graphical interface installation.

To use the Windows Installer command line interface, the Universal Enterprise Controller installation files first must be extracted from the product distribution file. Since there is no command line option available to unpack the distribution file, this must be done via the graphical user interface:

1. Download the Universal Enterprise Controller for Windows product distribution file, `UECt1r-4.1.0.<level>-nt-i386.exe`, to your work station.
2. Execute the distribution file to extract the installation files.
3. On the Location to Save Files dialog (Figure 4.28, below), select a location in which to store the installation files and click the **Next>** button to extract the files.

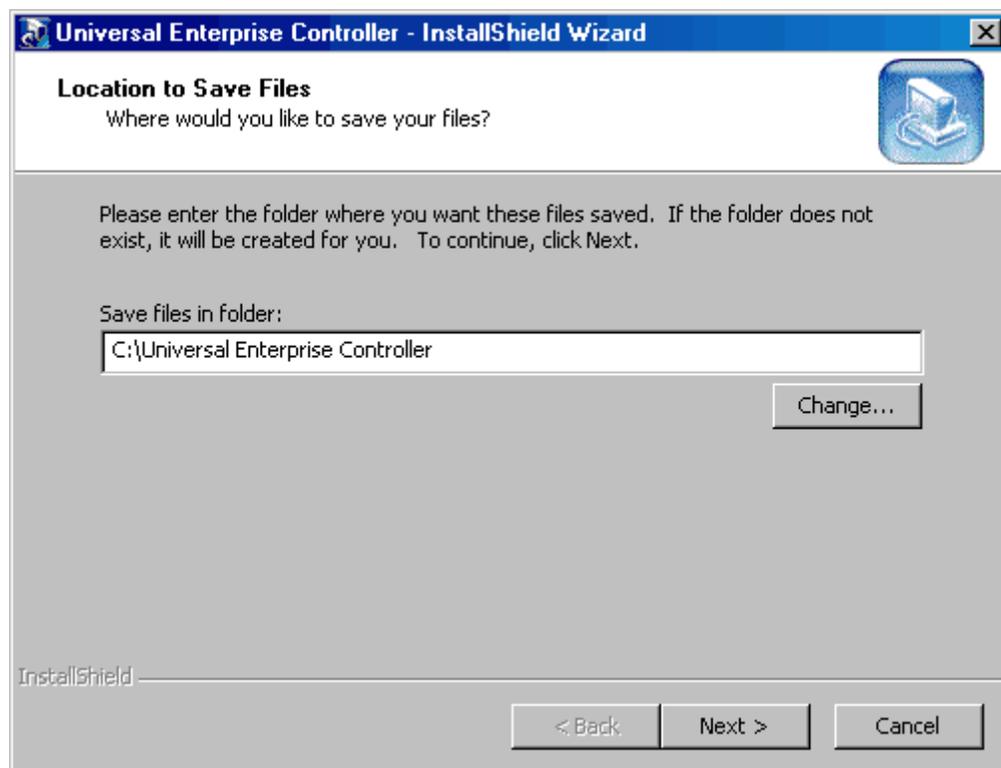


Figure 4.28 Universal Enterprise Controller - Location to Save Files dialog

4. On the Welcome dialog (see Figure 4.18), click the **Cancel** button.

At this point, all the installation files have been extracted into the location specified, but nothing has been installed. You may now complete the install from the command line, with no additional graphical interaction required.

Typically, to install an application that uses Windows Installer, a program such as **setup.exe** is executed. This program usually checks to make sure that a supported version of Windows Installer is installed. If one is not, the program will install it before proceeding. In fact, this is exactly what happens when Universal Enterprise Controller for Windows is installed from the product distribution file.

There is a command line interface available for **setup.exe**, but running it is not really ideal for unattended installations. If Windows Installer is installed or upgraded on the target system, a reboot may be required before the installation of Universal Enterprise Controller can continue (see [Determining if Windows Installer will be Installed or Upgraded](#) in Section 4.1.3 [Windows Installer](#)).

A better way to install Universal Enterprise Controller from the command line is to execute Windows Installer, **msiexec.exe** (see [Windows Installer Installation Commands](#)).

Note: A silent install also can be performed from the Universal Enterprise Controller for Windows distribution file (see [Using the Distribution File for a Silent Install](#)).

Windows Installer Parameters

[Table 4.24](#), below, describes the parameters available when using Windows Installer to install Universal Enterprise Controller from the command line.

Parameter	Description
/I	Installs Universal Enterprise Controller. This parameter cannot be used with the /f and /x parameters.
/fom	Repairs a Universal Enterprise Controller installation. The om after the /f are options actually used by the repair. There are other options available, but for behavior that matches the repair done from the graphical install, the om options must be used. This parameter cannot be used with the /I and /x parameters.
/x	Removes Universal Enterprise Controller. This parameter cannot be used with the /I and /f parameters.
SetupPath\UECtlr.msi	Specifies the path to the UECtlr.msi file. If the product distribution file is used, SetupPath will be whatever directory into which the files are extracted. Of course, the files can be copied from the original extraction directory into any location desired. In that case, SetupPath must point to the new location.
INSTALLDIR	Sets the root installation directory to Instal1Dir . Each component will be installed under this directory. This optional parameter is required only if you want to install UEC under a directory different from the one specified by the PROGRAMFILES environment variable (typically C:\Program Files\Universal). It is valid only when the /I switch is used. Otherwise, it is ignored. If the directory contains spaces, you must use double (") quotation marks around the path name.
UECTLR	Prevents the UEC service from being installed (if value is set to <i>no</i>). This has the same effect as choosing not to install the Universal Enterprise Controller from the Custom Setup dialog (see Section 4.10 Universal Enterprise Controller – Installation). This optional parameter is required only if you do not want to install the Universal Enterprise Controller service. It is valid only when the /I switch is used. Otherwise, it is ignored. [Default is yes.]
/q	Instructs Windows Installer to run without a graphical interface. If this optional parameter is omitted, it allows the UEC installation to be started from the command line, but run with a graphical interface. This is useful when an installation log file is desired. See Windows Installer Command Line Syntax , Windows Installer Installation Commands , and Detecting the Completion of Silent Installs in this section for additional information regarding silent installs.

Parameter	Description
/L *v	<p>Instructs Windows Installer to create an installation log file with a name of Logfilepath. If Logfilepath contains spaces, you must use double (") quotation marks around the path name.</p> <p>The *v after the /L actually are flags used to specify the level of detail contained in the log file. To reduce the amount of output generated, these options can be omitted. However, if a problem occurs during installation, Stonebranch Inc. Customer Support may request a log file, in which case these options should be used.</p> <p>This is an optional parameter.</p>

Table 4.24 Universal Enterprise Controller - Windows Installer Command Line Interface Parameters

Windows Installer Command Line Syntax

Figure 4.29, below, illustrates the command line syntax when using Windows Installer to install Universal Enterprise Controller from the command line.

```
msiexec.exe [/I|/fom|/x] SetupPath\UECtlr.msi  
[INSTALLDIR=InstallDir]  
[UECTLR={yes|no}]  
[/q] [/L*v Logfilepath]
```

Figure 4.29 Universal Enterprise Controller - Windows Installer Command Line Syntax

The order in which the parameters are specified is significant. The **/I**, **/f** or **/x** parameters must be first, followed by the name of the installation file, **UECtlr.msi**.

The parameters that set the installation directory, set the working folders, and control the installation of individual UEC components can be specified in any order. However, because they are used to set properties within the installation script, they must be specified before the **/q** and **/L** parameters. These parameters also are used only when the install switch (**/I**) is used.

The **/q** and **/L** switches can be specified in any order, but, when used, must come after all the others.

Windows Installer Installation Commands

The following commands allow installation of Universal Enterprise Controller from the command line using Windows Installer (**msiexec.exe**).

To install all Universal Enterprise Controller components using a graphical interface, issue the following command:

```
msiexec.exe /I SetupPath\UECt1r.msi
```

To install the Universal Enterprise Controller service under **D:\Universal** (that is, a directory other than the one specified by the environment variable **PROGRAMFILES**), issue the following command:

```
msiexec.exe /I SetupPath\UECt1r.msi INSTALLDIR=D:\Universal  
/q
```

To install all Universal Enterprise Controller components using the Windows Installer graphical interface, and instruct Windows Installer to write a log file to **C:\Temp\install.log** while performing the installation, issue the following command:

```
msiexec.exe /I SetupPath\UECt1r.msi /l C:\Temp\install.log
```

To install all Universal Enterprise Controller components without using a graphical interface — that is, a silent install — issue the following command:

```
msiexec.exe /I SetupPath\UECt1r.msi /q
```

Detecting the Completion of Silent Installs

When the `/q` switch is used to perform a silent install, no graphical interface or user interaction is required. One drawback to this is that no feedback is provided indicating when the Windows Installer process (install, uninstall, or repair) finishes.

One method that may be used to detect when the Windows Installer process ends is to execute it using the system's `start` command. Using available command line switches, the `start` command can be used to initiate the Windows Installer process and then wait for it to finish. When the `start` command returns control to its calling process (for example, the command prompt), the process will have ended.

For example, from the command prompt, issue the following command to start the Universal Enterprise Controller installation and wait for it to finish.

```
start /b /wait msiexec.exe /I SetupPath\UECtlr.msi /q
```

- The `/b` switch prevents the `start` command from opening a new window.
- The `/wait` parameter causes the `start` command to start the application, `msiexec.exe`, and then wait for it to finish.

The syntax above can also be used to execute the `start` command from within a script, such as a `.bat` file.

For more information on the `start` command, go to the Windows command prompt and enter: `start /?`.

Using the Distribution File for a Silent Install

To perform a silent install using the Universal Enterprise Controller for Windows distribution file, issue the following command:

```
UECtlr-4.1.0.0-nt-i386.exe /s /a /s /w /v" /qn"
```

The switches (from left to right) are defined as:

- `/s` "Package for the Web" switch for silent install.
- `/a` "Package for the Web" switch to "add" commands (that is, send the next commands to `setup.exe`).
- `/s` `setup.exe` switch for silent install.
- `/w` Wait switch, for a Basic MSI project, forces `setup.exe` to wait until the installation is complete before exiting.
- `/v` `setup.exe` switch to pass arguments or options to the MSI package. Quotes must be in place if a space-separated command is passed to the MSI.
- `/qn` `msiexec.exe` switch for quiet install with no graphic display of progress.

4.10.4 Modifying a Universal Enterprise Controller Installation via the Windows Installer Command Line Interface

This section describes how to modify a Universal Enterprise Controller installation via the Windows Installer command line interface.

After Universal Enterprise Controller is installed, Windows Installer can be run as many times as needed to modify the installation by:

- [Adding or Removing Universal Enterprise Controller Components](#)
- [Repairing a Corrupted Universal Enterprise Controller Installation](#)
- [Removing a Universal Enterprise Controller Installation](#)

Adding or Removing Universal Enterprise Controller Components

Currently, it only is possible to add or remove Universal Enterprise Controller components using the Windows Installer graphical interface (see [Adding or Removing Universal Products Components](#) in Section 4.10.2 [Modifying a Universal Enterprise Controller Installation via the Windows Installer Graphical Interface](#)).

Repairing a Corrupted Universal Enterprise Controller Installation

To recover accidentally deleted files or registry entries required by Universal Enterprise Controller using the Windows Installer command line interface, use the `/f` switch together with the `om` parameters.

These are the same repair options set internally by the graphical interface installation. They cause Windows Installer to reinstall files that are missing or older than the version contained in the Universal Enterprise Controller distribution file.

To repair a Universal Enterprise Controller installation from the command line, issue the following command:

```
msiexec.exe /fom SetupPath\UECt1r.msi /q
```

To repair a Universal Enterprise Controller installation using the Windows Installer graphical interface, and instruct Windows Installer to write a log file to `C:\Temp\repair.log` while running the repair, issue the following command:

```
msiexec.exe /fom SetupPath\UECt1r.msi /! C:\Temp\repair.log
```

Removing a Universal Enterprise Controller Installation

To uninstall Universal Enterprise Controller using the Windows Installer command line interface, use the `/x` switch. Issue the following command:

```
msiexec.exe /x SetupPath\UECt1r.msi /q
```

4.11 Universal Enterprise Controller – Installing on 64-bit Windows Editions

All Universal Products have been tested and verified on the 64-bit editions of the following Windows systems:

- Windows XP
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows Server 2008 R2

This section describes some modifications that may need to be made to the default installation options to ensure that the installed Universal Enterprise Controller components function correctly.

4.11.1 Applications Installed in the Windows System Folder

The Universal Enterprise Controller package installs several command-line applications in the Windows system folder. The default system folder for 32-bit applications installed on 64-bit Windows editions is the `%SystemRoot%\SysWOW64` directory (for example, `C:\Windows\SysWOW64`).

Table 4.25, below, identifies the affected Universal Products applications.

File Name	Description
ucert.exe	Universal Certificate
ueclod.exe	UEC Load
uencrypt.exe	Universal Encrypt

Table 4.25 Universal Enterprise Controller for Windows - Applications Installed in Windows System Folder

These applications can be executed using either the:

- 32-bit command shell (`%SystemRoot%\SysWOW64\cmd.exe`)
- Default 64-bit command shell (`%SystemRoot%\System32\cmd.exe`).

By default, the `%SystemRoot%\SysWOW64` directory is not part of the system path. Therefore, to execute `ueclod.exe` using the 32-bit command shell, either:

- Directory must be added to the `PATH` environment variable.
- Complete path to the application and/or the 32-bit command shell must be specified.

Example 1

To execute UECLoad in the default 64-bit command shell, issue the following command:

```
%SystemRoot%\SysWOW64\uecload
```

Example 2

To execute UECLoad within the 32-bit command shell, issue the following command:

```
%SystemRoot%\SysWOW64\cmd.exe /C %SystemRoot%\SysWOW64\uecload
```

4.12 Universal Enterprise Controller – Database Configuration

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

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

If it is not permissible, or there is no space to write these files in any of these directories, the following error can occur in UEC, and UEC shuts down:

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

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

1. Inside the UEC database directory, create a text file named **DB_CONFIG**.
2. Inside the **DB_CONFIG** file, add the following string:
set_tmp_dir *dbpath*
Where **dbpath** is the path to the location in which the database files reside.
3. Start / restart UEC.

4.13 Universal Enterprise Controller – File Inventory Lists

The Universal Enterprise Controller installation includes the files required for the following components / utilities:

- Universal Enterprise Controller
- Universal Configuration Manager
- Universal Products Install Merge

The Universal Configuration Manager is installed whenever Universal Enterprise is installed. It is used to maintain the configuration options for the Universal Enterprise Controller service. If any of the components already are installed, Windows Installer will upgrade them to the latest version.

The files installed with each Universal Enterprise Controller component / utility are listed in the following tables. The file paths specified are relative to the root installation directory (for example, `C:\Program Files\Universal`) that was specified during the installation.

Items shown with a path of `System32` are installed in the 32-bit system folder. The actual name of this directory depends on the Windows version:

- For Windows 2000, the path is `\Winnt\System32`.
- For all other supported 32-bit Windows editions, the path is `\Windows\System32`.
- For all supported 64-bit Windows editions, the path is `\Windows\SysWow64`.

Universal Enterprise Controller

File	Description
<code>nls*.utt</code>	Code page files used for text translation between different operating systems and platforms.
<code>nls\README.TXT</code>	Information about the contents of the <code>.nls</code> directory.
<code>nls\uecmceng.umc</code>	English message catalog.
<code>UCfgMgr\bin\ucfgmgr.cpl</code>	Universal Configuration Manager control panel application.
<code>UCfgMgr\bin\ucfgmgr.hlp</code>	Universal Configuration Manager help file.
<code>UECtlr\bin\acl.xml</code>	Used to store Access Control List entries for Universal Enterprise Controller.
<code>UECtlr\bin\brokers.xml</code>	Used to store a list of Universal Brokers that will be monitored by the Universal Enterprise Controller.
<code>UECtlr\bin\groups.xml</code>	Used to store defined groups of Universal Brokers.
<code>UECtlr\bin\librfc32.dll</code>	Support file for Universal Enterprise Controller.
<code>UECtlr\bin\ueccfg.dll</code>	Used by Universal Configuration Manager to manage Universal Enterprise Controller configuration options.
<code>UECtlr\bin\ueccfg.hlp</code>	Universal Enterprise Controller configuration help file.
<code>UECtlr\bin\uecdbrec.bat</code>	Recovers a Universal Enterprise Controller database, specified from the command line. The default database is <code>uec_evm.db</code> .
<code>UECtlr\bin\uecmsgnt.dll</code>	Used to write error messages to the Windows Application event log.

File	Description
UECt1r\bin\uect1rs.exe	Universal Enterprise Controller service program.
UECt1r\bin\users.xml	Used to store a list of user accounts authorized to use Universal Enterprise Controller and its associated Client Applications: I-Administrator, I-Activity Monitor, and I-Management Console.
UPIMerge\bin\upimerge.exe	Command line interface to configuration file merge routines).
USpool\bin\udb_dump.exe	UEC database utility; to be used only upon request of Stonebranch, Inc. Customer Support .
USpool\bin\udb_load.exe	UEC database utility; to be used only upon request of Stonebranch, Inc. Customer Support .
Installed under %ALLUSERSPROFILE%\ Application Data	
Universal\conf\ uec.conf	Universal Enterprise Controller configuration file.
%ALLUSERSPROFILE%\ Application Data, by default, resolves to:	
<ul style="list-style-type: none"> • 'C:\Documents and Settings\All Users\Application Data' on Windows 2000, 2003, and XP. • 'C:\ProgramData' on Vista and Windows 2008 Server. 	
Installed under %SystemRoot%	
ucert.exe	Universal Certificate utility, used to generate X.509 Certificates.
uecload.exe	Universal Enterprise Controller load utility.
uencrypt.exe	Universal Encrypt utility, used to encrypt sensitive Universal application command line options.
<ul style="list-style-type: none"> • On x86 platforms, these files are installed under %SystemRoot%\system32. • On x64 platforms, these files are installed under %SystemRoot%\SysWow64. 	
%SystemRoot% is an environment variable that, by default, resolves to:	
<ul style="list-style-type: none"> • C:\WINNT on Windows 2000. • C:\Windows on all other platforms. 	

Table 4.26 Universal Enterprise Controller for Windows - File Inventory List

Universal Products Install Merge Utility

File	Description
UPIMerge\bin\upimerge.exe	Application program, always installed. Provides command line access to the same functionality used by the Universal Products installation to merge options from a new configuration file into an existing file.

Table 4.27 Universal Products for Windows Install Merge Utility - File Inventory List

System Files

The following files will be installed only if they are newer than the existing file.

The directories shown in [Table 4.28](#), below, are relative to the %SYSTEMROOT% directory, where %SYSTEMROOT% is an environment variable that resolves to C:\WINNT on Windows 2000, and C:\Windows on all other Windows platforms.

File	Description
System32\asycfilt.dll	Version 2.40.4275.1. This DLL is one of the components of the Microsoft OLE library.
System32\comcat.dll	Version 4.71.1460.1 of the Microsoft Component Category Manager library.
Microsoft C-Runtime v8.0.50727.762 ¹	Version 8.0.50727.762 of the Microsoft C runtime side-by-side assembly.
System32\msiexec.exe	Version 3.1.4000.1823 of the Microsoft Windows Installer (see Section 4.1.3 Windows Installer for more information).
System32\oleaut32.dll	Version 2.40.4275.1. This DLL is one of the components of the Microsoft OLE library.
System32\olepro32.dll	Version 5.0.4275.1. This DLL is one of the components of the Microsoft OLE library.
System32\psapi.dll	Version 4.0.1371.1 of the Microsoft process status library
System32\stdole2.tlb	Version 2.40.4275.1. This file is one of the components of the Microsoft OLE library.
1	The Microsoft C-Runtime distribution consists of several files, which are subject to change. Refer to Microsoft documentation for a complete list of files delivered with the specified runtime version.

Table 4.28 Universal Enterprise Controller System Files - File Inventory List

4.14 Universal Enterprise Controller Client Applications Package

The Universal Enterprise Controller (UEC) 4.1.0 Client Applications for Windows package includes the desktop application versions of the following product components:

- I-Administrator 4.1.0
- I-Activity Monitor 4.1.0
- I-Management Console 4.1.0

4.15 UEC Client Applications – Installation Requirements

To install the UEC Client Applications, you must have one of the following versions of Windows:

- Windows 2000 Professional or Server with Service Pack 4 or higher
- Windows Server 2003 (Itanium not supported)
- Windows XP
- Windows Vista
- Windows Server 2008 (Itanium not supported)
- Windows 7
- Windows Server 2008 R2 (Itanium not supported)

Additionally, you also must have:

- About 5 megabytes of disk space. This value does not include space required for the Java Runtime Environment (JRE). See the JRE installation requirements, available from Sun, for more information.

An account with administrative privileges to install the UEC Client Applications is not required. However, the following conditions must be met before they can be successfully installed:

- Account used for the installation must have write access to the desired destination folder.
- No system policies (defined in Windows by your system administrator) may be in place that prohibit non-Administrative accounts from installing software.

To run the UEC Client Applications, you also must have the Java Runtime Environment (JRE) version 1.5 or above. You can download the latest JRE installation package directly from Sun's website, www.java.com.



Stoneman's Tip

Changes in Sun's versioning scheme for Java has created some confusion.

With version 1.2 of the Java environment, Sun began referring to Java as Java 2. The formal name was actually Java 2 with SDK 1.2.

As of January 2008, version 1.6 is latest version of the Java environment - a newer version than the original Java 2.

4.16 UEC Client Applications – Installation

There are two Windows Installer interfaces that can be used for installing UEC Client Applications: graphical and command line.

This section provides the following information regarding these two interfaces:

- [Installing UEC Client Applications via the Windows Installer Graphical Interface](#)
- [Modifying a UEC Client Applications Installation via the Windows Installer Graphical Interface](#)
- [Installing UEC Client Applications via the Windows Installer Command Line Interface](#)
- [Modifying a UEC Client Applications Installation via the Windows Installer Command Line Interface](#)

4.16.1 Installing UEC Client Applications via the Windows Installer Graphical Interface

To install UEC Client Applications for Windows using the Windows Installer graphical interface, perform the following steps:

1. Download the UEC Client Applications for Windows product distribution file, `UECClient-4.1.0.<level>-nt-i386.exe`, to your work station.
2. Execute the distribution file to extract the files.

Note: If you already have extracted the files from the distribution file, but cancelled installation in order to separately install Windows Installer (see [Section 4.1.3 Windows Installer](#)), you can simply double-click the extracted Client Applications installation file, `UECClient.msi`, to begin the installation.

Installing over a Remote Desktop Session

Starting with Windows Server 2003, Remote Desktop provides distinct session environments for each logged-in user. This means extraction may use an environment setting that is not available once the Remote Desktop session ends.

For example, the distribution file's default extraction location is based on the TEMP environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed.

To ensure that extracted files and other required resources are accessible after the initial install, extract the files to a well-known location that is not likely to change between Remote Desktop sessions.

Refer to the Microsoft documentation on the Remote Desktop feature for additional information.

The installation automatically will begin after the files are extracted.

It first will verify that your machine meets the minimum system requirements (see [Section 4.15 UEC Client Applications – Installation Requirements](#)). If the requirements are met, a Welcome dialog displays ([Figure 4.30](#)).



Figure 4.30 UEC Client Applications - Welcome dialog

3. Click the **Next** button.

- If the installation is being performed by a user account that is a member of the Administrators group, the Customer Information dialog displays (Figure 4.31, below).
- If the installation is being performed by a user account that is NOT a member of the Administrators group, the Custom Setup dialog displays (see Figure 4.32).

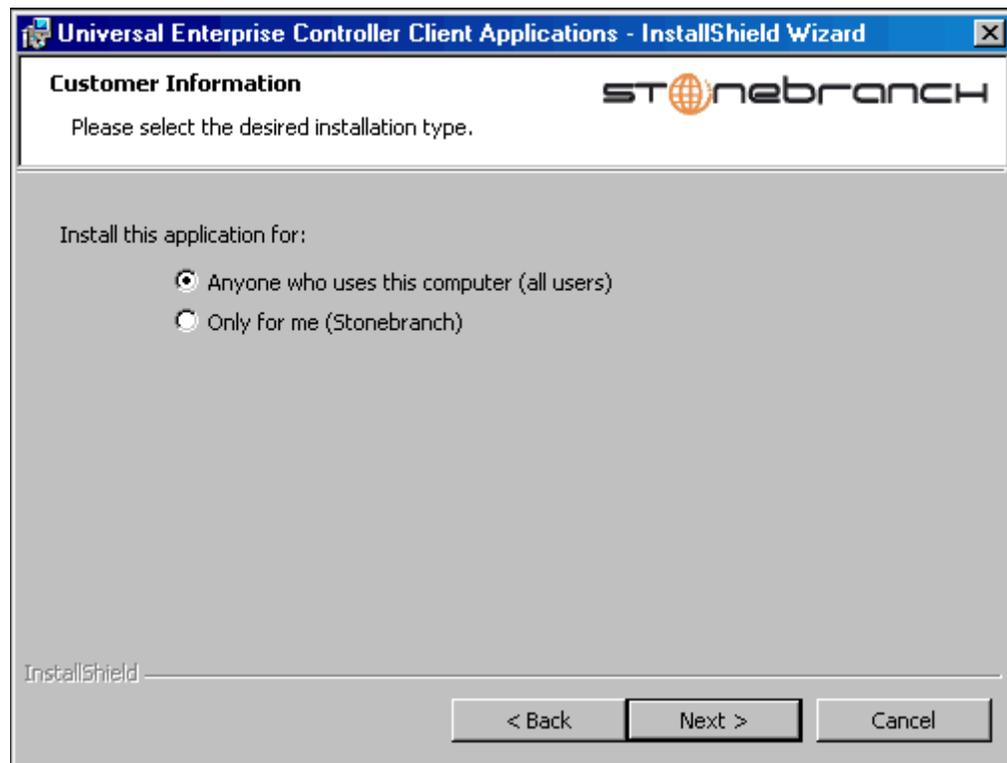


Figure 4.31 UEC Client Applications - Customer Information dialog (for Administrators accounts)

4. Click the appropriate radio button on this dialog:
 - If you want to perform an installation for all users with access to a given workstation, click **Anyone who uses this computer**.

This is referred to as a *per-machine* installation. When this type of installation is performed, shortcuts added to the **Start** menu, and any configuration settings stored in the Windows registry, are placed in a location accessible to anyone who logs on to that particular machine.

This type of installation also creates an entry for UEC Client Applications in the **Add or Remove Programs** list, accessible via the Windows Control Panel. However, only accounts with administrative privileges will be able to:

- Modify the installation (see [Section 4.16.2 Modifying a UEC Client Applications Installation via the Windows Installer Graphical Interface](#))
 - Uninstall the product (see [Removing a UEC Client Applications Installation](#)).
- If you want to perform an installation for the user identified on this dialog, click **Only for me**. (This is the account performing the installation.)

This is referred to as a *per-user* installation. When this type of installation is performed, the Start menu shortcuts, the configuration options stored in the Windows registry, and the application itself (depending on where it's installed) will be accessible only by this user.

This type of installation also creates an entry for UEC Client Applications in the **Add or Remove Programs** list, accessible via the Windows Control Panel. However, it will be visible only by this user.

In this case, the user also will be able to:

- Modify the installation (see Section [4.16.2 Modifying a UEC Client Applications Installation via the Windows Installer Graphical Interface](#)).
- Uninstall the product (see [Removing a UEC Client Applications Installation](#)).

For the UEC Client Applications to be available to another user on this machine, that user also must perform a *per-user* installation.

Note: It is possible for a *per-machine* and a *per-user* installations to be performed on the same machine, although there is no benefit in doing this. In that situation, when any user that has done a *per-user* installation is logged on, that installation takes precedence. If the user removes the Client Applications, the *per-machine* installation will remain in effect. Conversely, if the *per-machine* installation is removed, users that have performed a *per-user* installation will still have access to Client Applications.



For Non-Administrative Users

Because *per-machine* installations require access to certain system resources to which most non-Administrative accounts do not have access, all installations performed by non-Administrative users are *per-user* installations.

Stoneman's Tip

5. Click the **Next>** button. A list of UEC Client Applications components included in the installation package then displays ([Figure 4.32](#)). It is from this list that you can select which components to install.

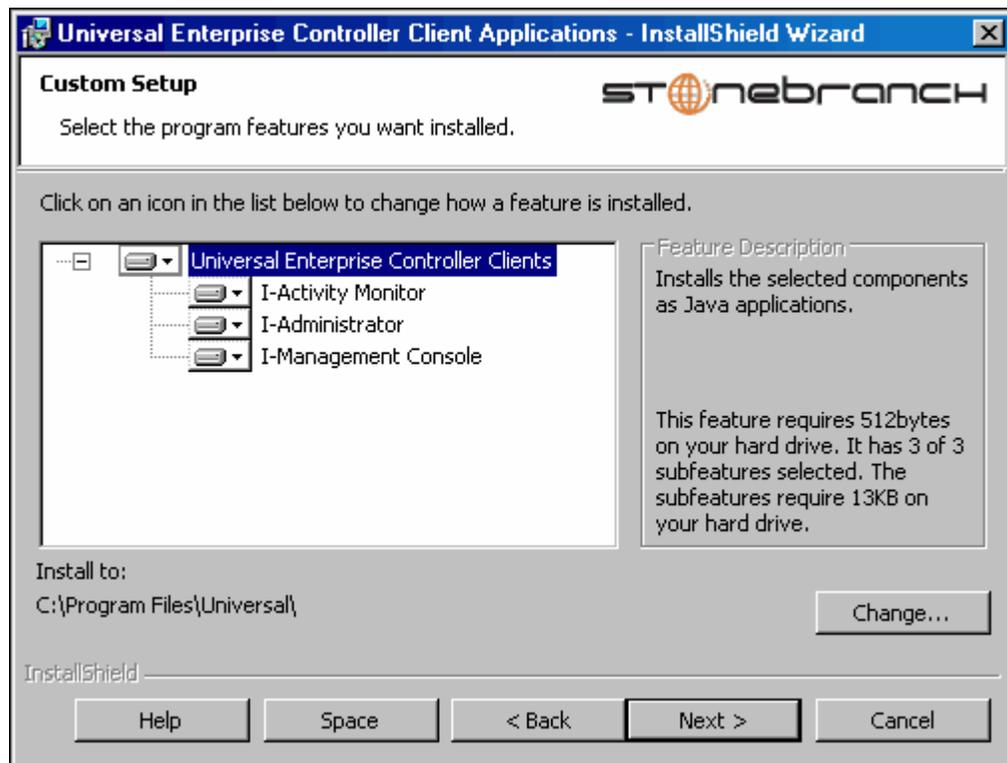


Figure 4.32 UEC Client Applications - Windows Installer / Custom Setup dialog

For a new installation, a drive icon  displays next to each item in the list, indicating that the component will be installed.

For an upgrade installation, either of the following icons displays next to an item:

- A drive icon  indicates that the component is either:
 - New to the installation and will be installed.
 - Currently is installed and will be upgraded.
- An **X** icon indicates that the component is either:
 - Currently not installed (but previously was available).
 - Previously installed but removed.



The directory identified in [Figure 4.32](#) is typical for a new, *per-machine* installation. Depending on the type of installation being performed, the directory may be different.

For a new, *per-user* installation, the dialog will identify a directory located within the user's profile directory (for example, `C: \Documents and Settings \username`).

For installation upgrades, whether *per-machine* or *per-user*, the UEC Client Applications' current location is displayed.

Stoneman's Tip

6. [Figure 4.32](#) shows that all UEC Client Applications will be installed in their respective directories under the `C:\Program Files\Universal` directory.
 - a. If you want to select a different location, click the **Change...** button.
 - b. If you want to check the amount of disk space required for the installation, and the amount of available disk space on the selected directory, click the **Space** button.
7. If you do not want to install a component:
 - a. Click the drive icon  next to that component name.
 - b. From the drop-down list that displays, select the **X** icon to mark the component as one not to be installed.

For example, [Figure 4.33](#), below, indicates that I-Management Console has been selected to not be installed.

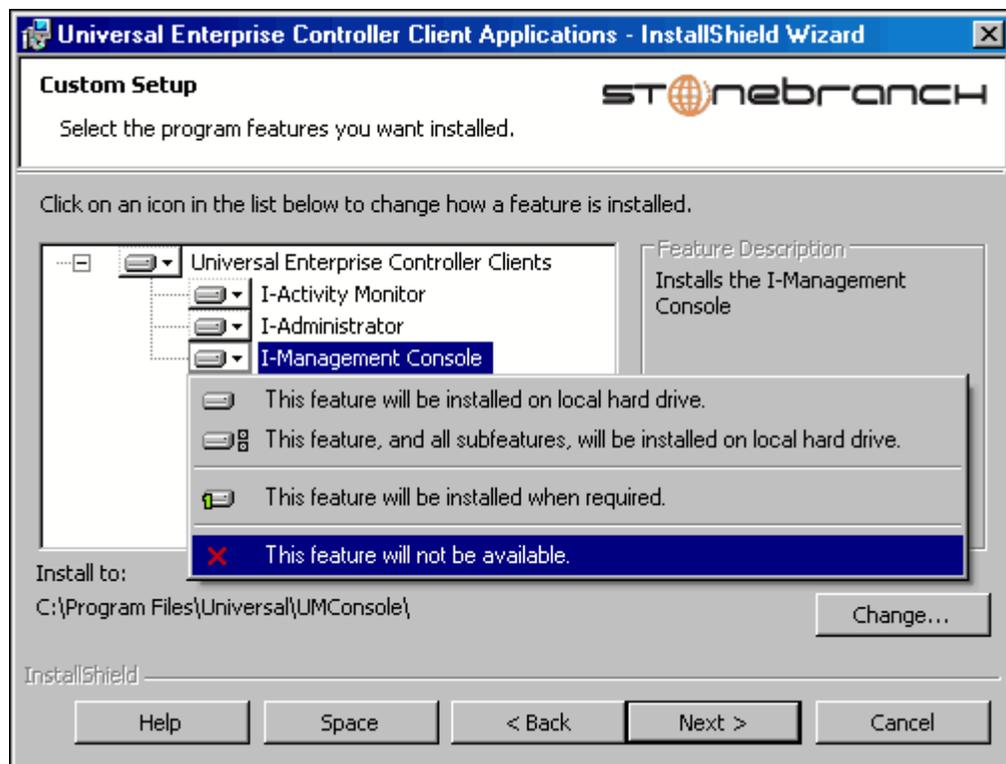


Figure 4.33 UEC Client Applications - Custom Setup / Selecting to not Install a Component

8. When you have selected the components (and their installation destinations) that you want to install, click the **Next** button to continue the installation process.

When the installation completes successfully, the Installation Complete dialog displays ([Figure 4.34](#)).

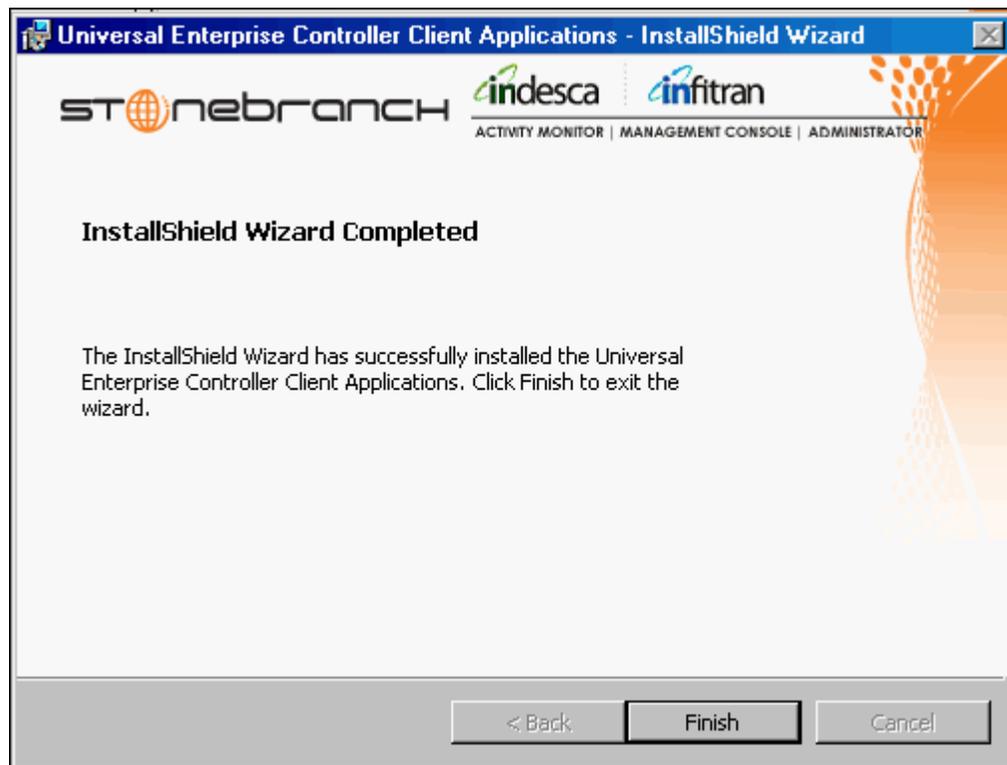


Figure 4.34 UEC Client Applications - Installation Complete

Note: During the install, you may have been notified that no Java Runtime Environment (JRE) was detected on your system. The UEC Client Applications require Java Runtime Environment (JRE) Version 1.5 or greater (see [Section 4.15 UEC Client Applications – Installation Requirements](#)).

9. Click the **Finish** button to exit Windows Installation.

4.16.2 Modifying a UEC Client Applications Installation via the Windows Installer Graphical Interface

After the UEC Client Applications are installed, Windows Installer can be run as many times as needed to modify the installation.

These installation modifications are:

- [Adding or Removing UEC Client Applications Components](#)
- [Repairing a Corrupted UEC Client Applications Installation](#)
- [Removing a UEC Client Applications Installation](#)

Adding or Removing UEC Client Applications Components

To add or remove components from a UEC Client Applications installation, perform the following steps:

1. On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays ([Figure 4.35](#), below).

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

The Programs and Features dialog (see [Figure 4.7](#)) replaces the Add or Remove Programs dialog.

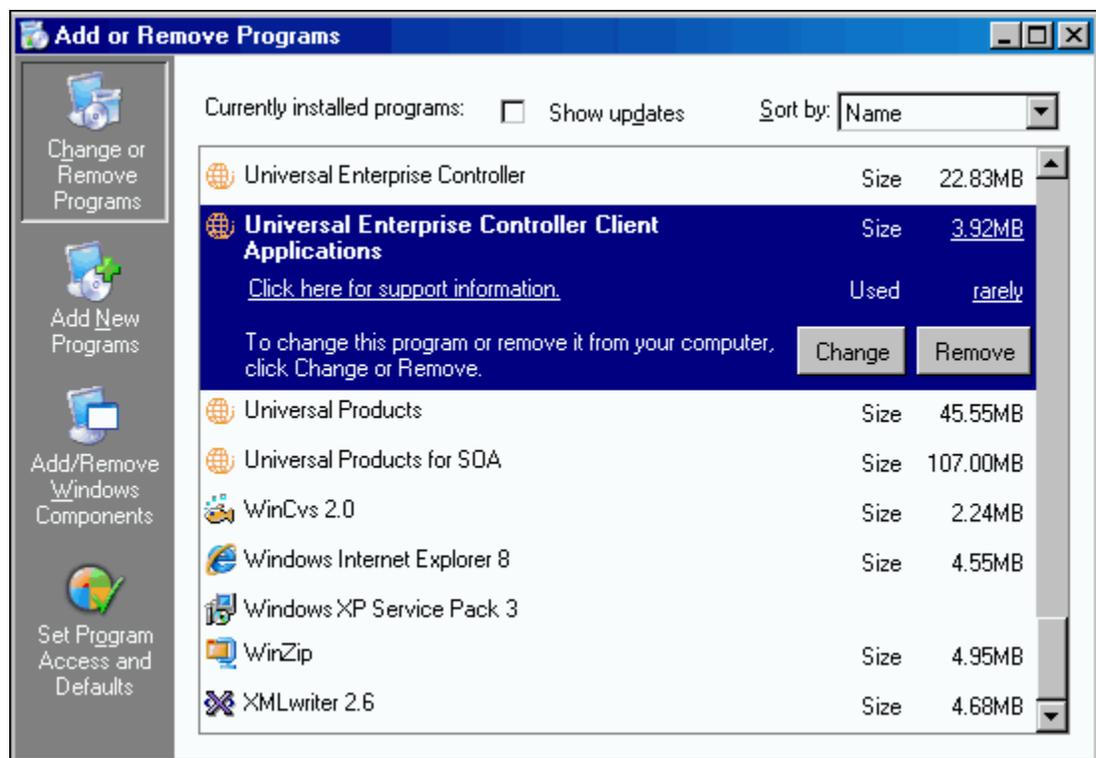


Figure 4.35 UEC Client Applications - Add or Remove Programs dialog

2. From the list of installed programs, select **Universal Enterprise Controller Client Applications**.
3. Click the **Change** button to start Windows Installer.
4. On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays (Figure 4.37).

Windows Server 2003

If the UEC Client Applications were installed via a Remote Desktop connection, the environment settings used during that session may no longer be available. Windows Server 2003 allows multiple Remote Desktop sessions for multiple users, and each session has its own environment. Depending on the way the Remote Desktop session for the UEC Client Applications installation was established, some problems may arise whenever an attempt is made to modify that installation.

The distribution file's default extraction location is based on the **TEMP** environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed. Consequently, any attempts to modify the installation may fail because the Windows Installer cannot locate the installation's source files (a dialog similar to the one shown in Figure 4.36, below may be displayed).

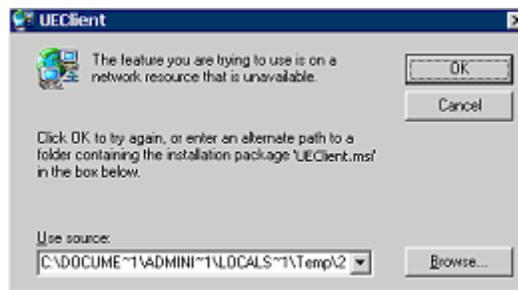


Figure 4.36 Failure to Access Install Files

To resolve this issue, re-extract the distribution files to a location that is independent of a Remote Desktop environment and specify that location in the dialog above. Keep in mind, however, that the extracted files must come from the same distribution package used to do the initial install. If matching distribution files can't be found, the UEC Client Applications must be uninstalled and then reinstalled with the desired modifications.

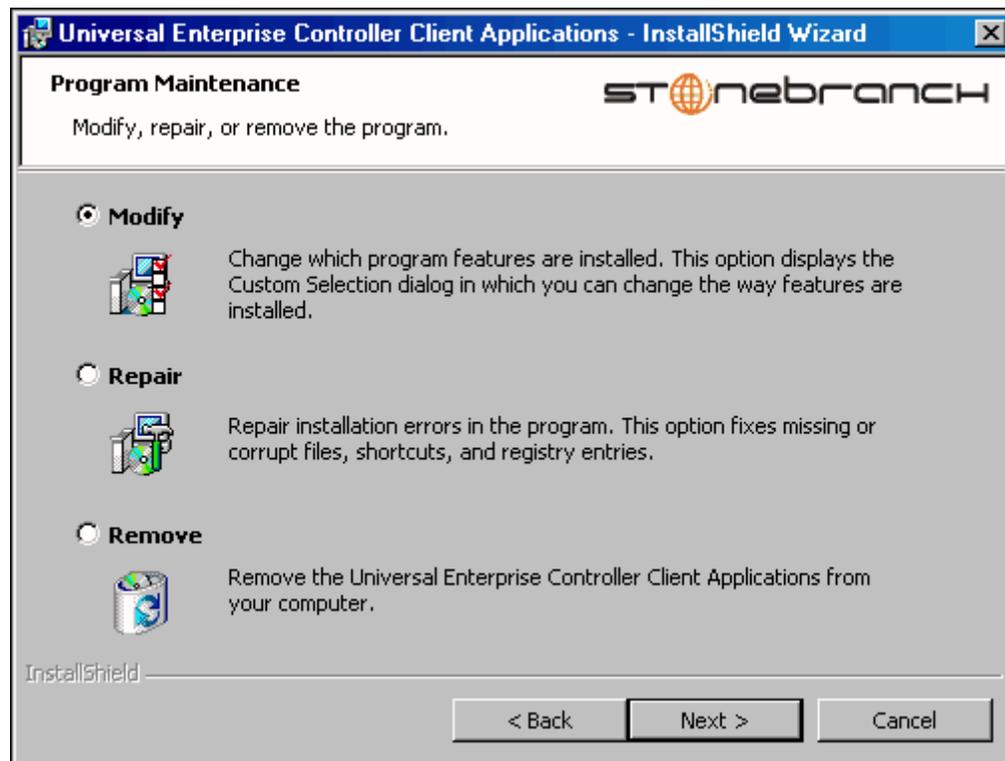


Figure 4.37 UEC Client Applications - Program Maintenance / Modify dialog

5. Click the **Modify** radio button, and then the **Next>** button, to display the Custom Setup dialog (see [Figure 4.38](#)).

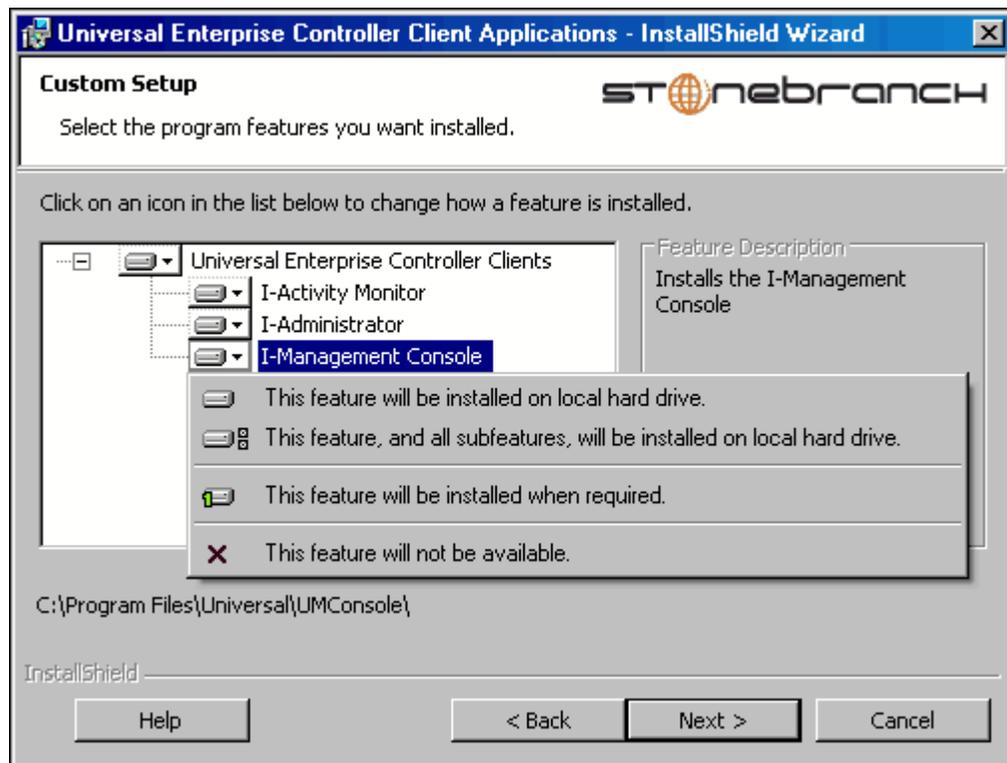


Figure 4.38 UEC Client Applications - Custom Setup / Modify dialog

Currently installed components are identified by a drive icon .

Uninstalled components are identified by an **X** icon.

6. To remove a currently installed component:
 - a. Click the drive icon  next to that component.
 - b. Select the **X** icon from the drop-down list to mark the component for removal.
7. To add an uninstalled component:
 - a. Click the **X** icon next to that component.
 - b. Select the drive icon  from the drop-down list to mark the component for installation.
8. Click the **Next>** button to continue with the modification.

When the modifications are complete, the following actions will be taken:

- Components marked with a drive icon  will:
 - Remain installed if they already are installed.
 - Be installed if they are not already installed.
- Components marked with an **X** will:
 - Remain uninstalled if they are not currently installed
 - Be removed if they currently are installed.

Repairing a Corrupted UEC Client Applications Installation

Windows Installer has the ability to recover accidentally deleted application files or registry entries required by the UEC Client Applications. This repair feature will re-install the missing items, making a complete re-install unnecessary.

To repair an installation, perform the following steps:

1. On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays (see [Figure 4.35](#)).

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

The Programs and Features dialog (see [Figure 4.7](#)) replaces the Add or Remove Programs dialog.

2. From the list of installed programs, select **Universal Enterprise Controller Client Applications**.
3. Click the **Change** button to start Windows Installer.
4. On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays ([Figure 4.26](#), below).

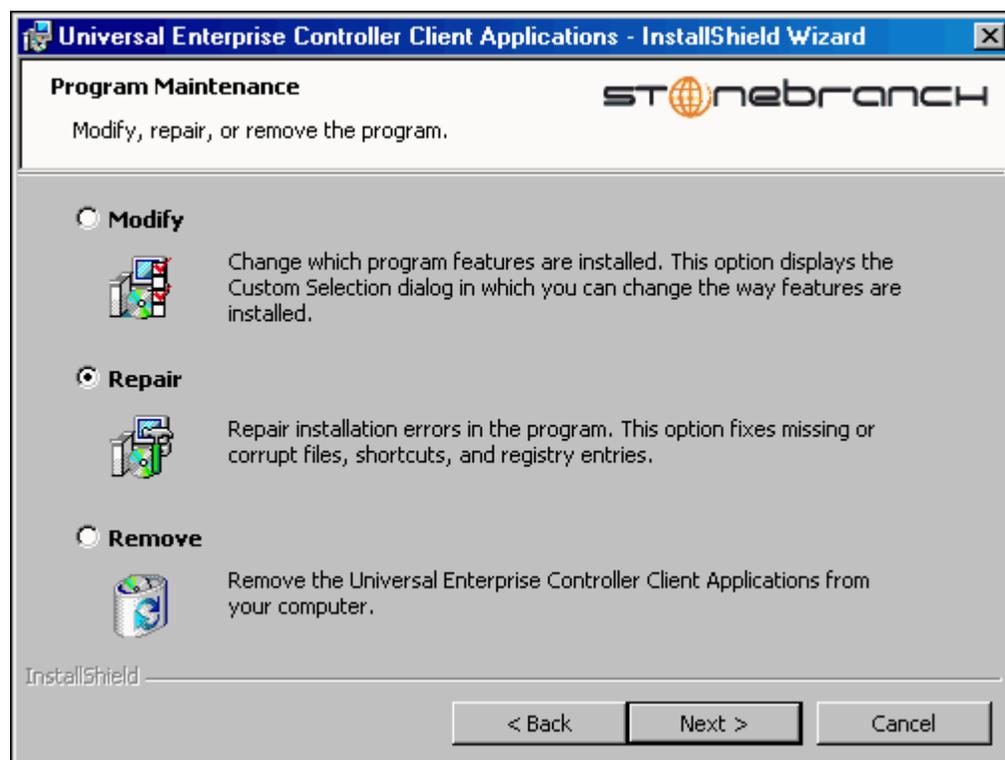


Figure 4.39 UEC Client Applications - Program Maintenance / Repair dialog

5. Click the **Repair** radio button, and then the **Next>** button, to display the Ready to Repair dialog.
6. Follow the instructions displayed in successive dialogs to complete the repair.

Removing a UEC Client Applications Installation

To uninstall a UEC Client Applications installation, perform the following steps:

1. On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays (see [Figure 4.35](#)).

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

The Programs and Features dialog (see [Figure 4.7](#)) replaces the Add or Remove Programs dialog.

2. From the list of installed programs, select **Universal Enterprise Controller Client Applications**.

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

To skip the remaining steps, simply select the **Uninstall** button (see [Figure 4.7](#)).

3. Click the **Change** button to start the installation program.
4. On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays ([Figure 4.40](#), below).

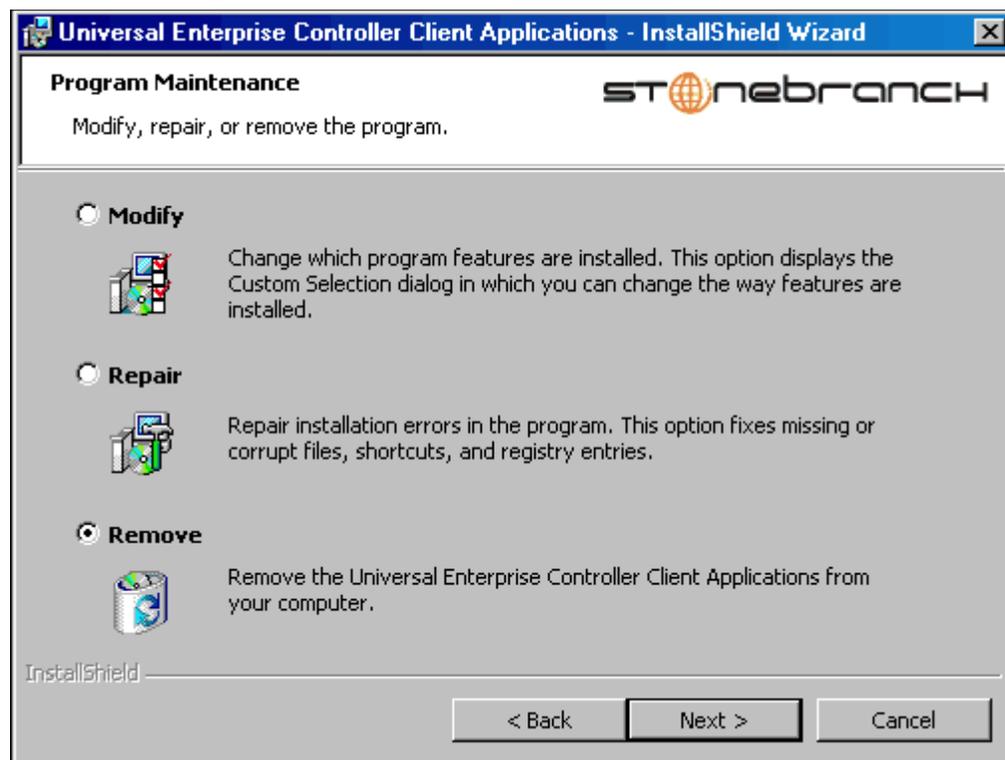


Figure 4.40 UEC Client Applications - Program Maintenance / Remove dialog

5. Select **Remove** and click the **Next>** button.
6. On the subsequent dialog, click the **Remove** button to uninstall the installation.

Un-Installed Files

The uninstall process will remove only those files created during the installation. Some files stored under the .\Universa1 install directory may be left behind after the uninstall. In this situation, those files and/or directories may simply be deleted.

Before deleting the entire .\Universa1 directory, make sure that no other Stonebranch, Inc. products are installed there. (See Section [4.18 UEC Client Applications – File Inventory List](#) for a list of files and directories installed with UEC Client Applications.)

4.16.3 Installing UEC Client Applications via the Windows Installer Command Line Interface

This section describes how to install UEC Client Applications using the Windows Installer command line interface.

The command line interface is useful in situations where:

1. Several UEC Client Applications installations must be deployed across many different systems.
2. It is not practical or convenient to perform the graphical interface installation.

To use the Windows Installer command line interface, the UEC Client Applications installation files first must be extracted from the product distribution file. Since there is no command line option available to unpack the distribution file, this must be done via the graphical user interface:

1. Download the UEC Client Applications for Windows product distribution file, **UECClient-4.1.0.<level>-nt-i386.exe**, to your work station.
2. Execute the distribution file to extract the installation files.
3. On the Location to Save Files dialog (Figure 4.41, below), select a location in which to store the installation files and click the **Next>** button to extract the files.

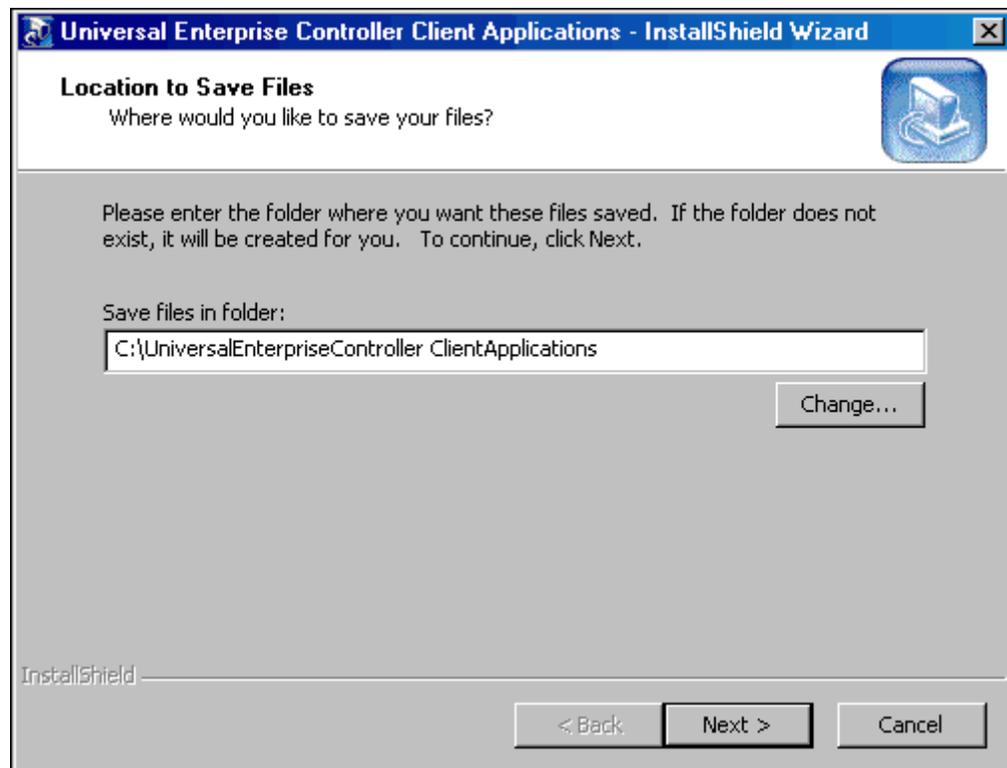


Figure 4.41 UEC Client Applications - Location to Save Files dialog

4. On the Welcome dialog (see [Figure 4.30](#)), click the **Cancel** button.

At this point, all the installation files have been extracted into the location specified, but nothing has been installed. You may now complete the install from the command line, with no additional graphical interaction required.

Typically, to install an application that uses Windows Installer, a program such as **setup.exe** is executed. This program usually checks to make sure that a supported version of Windows Installer is installed. If one is not, the program will install it before proceeding. In fact, this is exactly what happens when UEC Client Applications for Windows is installed from the product distribution file.

There is a command line interface available for **setup.exe**, but running it is not really ideal for unattended installations. If Windows Installer is installed or upgraded on the target system, a reboot may be required before the installation of Universal Products can continue (see [Determining if Windows Installer will be Installed or Upgraded](#) in Section [4.1.3 Windows Installer](#)).

A better way to install UEC Client Applications from the command line is to execute Windows Installer, **msiexec.exe** (see [Windows Installer Installation Commands](#)).

Note: A silent install also can be performed from the UEC Client Applications for Windows distribution file (see [Using the Distribution File for a Silent Install](#)).

Windows Installer Command Line Parameters

Table 4.29, below, describes the parameters available when using Windows Installer to install UEC Client Applications from the command line.

Parameter	Description
/I	Installs the UEC Client Applications. This parameter cannot be used with the /f and /x parameters.
/fom	Repairs a UEC Client Applications installation. The om after the /f actually are options used by the repair. There are other options available, but for behavior that matches the repair done from the graphical install, the om options must be used. This parameter cannot be used with the /I and /x parameters.
/x	Removes the UEC Client Applications. This parameter cannot be used with the /I and /f parameters.
SetupPath\UECClient.msi	Specifies the path to the UECClient.msi file. If the product distribution file is used is used, SetupPath will be whatever directory into which the files are extracted. Of course, the files can be copied from the original extraction directory into any location desired. In that case, SetupPath must point to the new location.
INSTALLTO	Sets the root installation directory to InstallDir . Each component will be installed under this directory. This optional parameter is required only if you want to install the UEC Client Applications in a directory that is different from the default. The default directory varies, depending on the type of installation being done. (For a discussion regarding the different installation types, see the PERUSER parameter in this table). For a per-machine installation, the installation directory will default to the one specified by the PROGRAMFILES environment variable (typically C:\Program Files\Universal). For a per-user installation, the install directory will default to a location under the directory specified by the USERPROFILE environment variable. For example, if the USERPROFILE directory is set to C:\Documents and Settings\username , the default target directory will be C:\Documents and Settings\username\Application Data\Universal . If the UEC Client Applications already are installed, the installation directory will default to its existing location. This parameter is valid only when the /I switch is used. Otherwise, it is ignored. If the directory contains spaces, you must use double (") quotation marks around the path name.

Parameter	Description
PERUSER	<p>Performs an installation for all users or a specific user account:</p> <ul style="list-style-type: none"> If the value is <i>0</i>, it performs an install for all users of a given machine. This means that any UEC Client Applications configuration options stored in the Windows registry, Start menu short cuts that launch the UEC Client Applications, and the applications files themselves will be visible to all accounts on the machine where the UEC Client Applications was installed. This is referred to as a per-machine installation. If the value is <i>1</i>, it performs an install that is specific to the user account doing the installation. This means that any UEC Client Applications configuration options stored in the Windows registry, Start menu short cuts that access the UEC Client Applications, and the application files themselves will be visible only to the user account with which the installation was performed. It will appear to all other users of a given machine as though the UEC Client Applications is not actually installed. This is referred to as a per-user installation. <p>This optional parameter is required only under the following situations:</p> <ul style="list-style-type: none"> For per-user installs, when the installation is being done with a Windows account that is a member of the Administrators group. For uninstalls, where a Windows account that is a member of the Administrators group is removing a per-user installation. <p>[Default is <i>0</i> if UEC Client Applications is installed using an Administrative account.]</p> <p>[Default is <i>1</i> if a regular user account (that is, a non-Administrative user) is executing the installation.]</p>
UECADMIN	<p>Prevents I-Administrator from being installed (if value is <i>no</i>). This has the same effect as choosing not to install the application version of the I-Administrator from the Custom Setup dialog (see Section 4.16 UEC Client Applications – Installation).</p> <p>This optional parameter is required only if you do not want to install the application version of I-Administrator.</p> <p>It is valid only when the <i>/I</i> switch is used. Otherwise, it is ignored.</p> <p>[Default is <i>yes</i>.]</p>
UAMONITOR	<p>Prevents I-Activity Monitor from being installed (if value is <i>no</i>). This has the same effect as choosing not to install the application version of the I-Activity Monitor utility from the Custom Setup dialog (see Section 4.16 UEC Client Applications – Installation).</p> <p>This optional parameter is required only if you do not want to install the application version of I-Activity Monitor.</p> <p>This parameter is valid only when the <i>/I</i> switch is used. Otherwise, it is ignored.</p> <p>[Default is <i>yes</i>.]</p>

Parameter	Description
UMGMTCON	<p>Prevents I-Management Console from being installed (is value is <i>no</i>). This has the same effect as choosing not to install the application version of the I-Management Console utility from the Custom Setup dialog (see 4.16 UEC Client Applications – Installation).</p> <p>This optional parameter is required only if you do not want to install the application version of I-Management Console.</p> <p>This parameter is valid only when the /I switch is used. Otherwise, it is ignored.</p> <p>[Default value is <i>yes</i>.]</p>
/q	<p>This optional parameter is required only if you want the Microsoft installer to run without a graphical interface.</p> <p>If this optional parameter is omitted, it allows the UEC Client Applications installation to be started from the command line, but run with a graphical interface. This is useful when an installation log file is desired.</p> <p>See Windows Installer Command Line Syntax, Windows Installer Installation Commands, and Detecting the Completion of Silent Installs in this section for additional information regarding silent installs.</p>
/L*v	<p>Instructs Windows Installer to create an installation log file with a name of Logfilepath. If Logfilepath contains spaces, you must use double (") quotation marks around the path name.</p> <p>The *v after the /L actually are flags used to specify the level of detail contained in the log file. To reduce the amount of output generated, these options can be omitted. However, if a problem occurs during installation, Stonebranch Inc. Customer Support may request a log file, in which case these options should be used.</p> <p>This is an optional parameter.</p>

Table 4.29 UEC Client Applications - Windows Installer Command Line Interface Parameters

Windows Installer Command Line Syntax

Figure 4.42, below, illustrates the command line syntax when using Windows Installer to install UEC Client Applications from the command line.

```
msiexec.exe [/I|/fom|/x] SetupPath\UEClient.msi  
[INSTALLTO=InstallDir] [PERUSER={0|1}]  
[UECADMIN={yes|no}] [UAMONITOR={yes|no}] [UMGMTCON={yes|no}]  
[/q] [/L*v Logfilepath]
```

Figure 4.42 UEC Client Applications - Windows Installer Command Line Syntax

The order in which the parameters are specified is significant. The **/I**, **/f** or **/x** parameters must be first, followed by the name of the installation file, **UEClient.msi**.

The parameters that set the installation directory, set the working folders, and control the installation of the I-Administrator, I-Activity Monitor, and I-Management Console components can be specified in any order. However, because they are used to set properties within the installation script, they must be specified before the **/q** and **/L** parameters. These parameters also are used only when the install switch (**/I**) is used.

The **/q** and **/L** switches can be specified in any order, but, when used, must come after all the others.

Windows Installer Installation Commands

The following commands allow installation of UEC Client Applications from the command line using Windows Installer (`msiexec.exe`).

To install all UEC Client Applications components using the Windows Installer graphical interface, issue the following command:

```
msiexec.exe /I SetupPath\UEClient.msi
```

To install all UEC Client Applications components as a *per-user* installation (that is, one in which registry entries, Start menu shortcuts, and so on are visible only to the user performing the installation), and to override the default installation directory, issue the following command:

```
msiexec.exe /I SetupPath\UEClient.msi  
INSTALLTO=C:\UEClientApps\Universal /q
```

To install all UEC Client Applications components using the Windows Installer graphical interface, and instruct Windows Installer to write a log file to `C:\Temp\install.log` while performing the installation, issue the following command:

```
msiexec.exe /I SetupPath\UEClient.msi /l C:\Temp\install.log
```

To install all UEC Client Applications components using Windows Installer without a graphical interface — that is, a silent install — issue the following command:

```
msiexec.exe /I SetupPath\UEClient.msi /q
```

Detecting the Completion of Silent Installs

When the `/q` switch is used to perform a silent install, no graphical interface or user interaction is required. One drawback to this is that no feedback is provided indicating when the Windows Installer process (install, uninstall, or repair) finishes.

One method that can be used to detect when the Installer process (`msiexec.exe`) ends is to execute it using the system's `start` command. Using available command line switches, the `start` command can be used to kick off the Installer process, and then wait for it to finish. When the `start` command returns control to its calling process (for example, the command prompt), the Installer process will have ended.

For example, from the command prompt, the following command can be issued to start the UEC Client Applications installation, and wait for it to finish.

```
start /b /wait msiexec.exe /I SetupPath\UEClient.msi /q
```

The `/b` switch prevents the `start` command from opening a new window.

The `/wait` parameter causes the `start` command to start the application, `msiexec.exe`, and then wait for it to finish.

The syntax above also can be used to execute the `start` command from within a script, such as a `.bat` file.

For more information on the `start` command, go to the Windows command prompt and enter: `start /?`.

Using the Distribution File for a Silent Install

To perform a silent install using the UEC Client Applications for Windows distribution file, issue the following command:

```
UEClient-4.1.0.0-nt-i386.exe /s /a /s /w /v" /qn"
```

The switches (from left to right) are defined as:

- `/s` "Package for the Web" switch for silent install.
- `/a` "Package for the Web" switch to "add" commands (that is, send the next commands to `setup.exe`).
- `/s` `setup.exe` switch for silent install.
- `/w` Wait switch, for a Basic MSI project, forces `setup.exe` to wait until the installation is complete before exiting.
- `/v` `setup.exe` switch to pass arguments or options to the MSI package. Quotes must be in place if a space-separated command is passed to the MSI.
- `/qn` `msiexec.exe` switch for quiet install with no graphic display of progress.

4.16.4 Modifying a UEC Client Applications Installation via the Windows Installer Command Line Interface

Adding or Removing Components

Currently, it only is possible to add or remove UEC Client Applications components using the Windows Installer graphical interface. (see [Adding or Removing Components in Section 4.16.2 Modifying a UEC Client Applications Installation via the Windows Installer Graphical Interface](#)).

Repairing a Corrupted UEC Client Applications Installation

To recover accidentally deleted files or registry entries required by the UEC Client Applications using the Windows Installer command line interface, use the `/f` switch together with the `om` parameters.

These are the same repair options set internally by the graphical interface installation. They cause Windows Installer to reinstall files that are missing or that are older than the version contained in the UEC Client Applications distribution file.

To repair a UEC Client Applications installation from the command line, use the following command:

```
msiexec.exe /fom SetupPath\UEClient.msi /q
```

To repair a UEC Client Applications installation using the Windows Installer graphical interface, and instruct Windows Installer to write a log file to `C:\Temp\repair.log` while running the repair, issue the following command:

```
msiexec.exe /fom SetupPath\UEClient.msi /l C:\Temp\repair.log
```

Removing UEC Client Applications from the Command Line

To uninstall UEC Client Applications using the Windows Installer command line interface, use the `/x` switch.

To uninstall the UEC Client Applications without using a graphical interface, issue the following command:

```
msiexec.exe /x SetupPath\UECClient.msi /q
```

To remove a *per-user* installation that was installed using an Administrator account, issue the following command:

```
msiexec.exe /x SetupPath\UECClient.msi PERUSER=1 /q
```



Stoneman's Tip

If you know that the UEC Client Applications were installed using an account that is a member of the Administrators group, but are not sure if a *per-user* or *per-machine* installation was done, it might be best to uninstall the Client Applications using the graphical interface (see [Removing a UEC Client Applications Installation](#)).

This will ensure complete removal of the UEC Client Applications.

If you are removing a UEC Client Applications installation that was installed using a non-administrative account, the **PERUSER** parameter is not necessary.

4.17 UEC Client Applications – Installing on 64-bit Windows Editions

All Universal Products have been tested and verified on the 64-bit editions of the following Windows systems:

- Windows XP
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows Server 2008 R2

The installation defaults for the UEC Client Applications should not require any modification when installing on 64-bit Windows editions.

4.18 UEC Client Applications – File Inventory List

The UEC Client Applications installation includes the files required for the following components:

- I-Administrator
- I-Activity Monitor
- I-Management Console

If any of the components already are installed, Windows Installer will upgrade them to the latest version.

This section lists the files installed with each UEC Client Applications component. The file paths specified are relative to the root installation directory that was specified during the installation.

Items shown with a path of System32 are installed in the 32-bit system folder. The actual name of this directory depends on the Windows version:

- For Windows 2000, the path is `\Winnt\System32`.
- For all other supported 32-bit Windows editions, the path is `\Windows\System32`.
- For all supported 64-bit Windows editions, the path is `\Windows\SysWow64`.

I-Administrator

File	Description
<code>UECAdmin\uecadmin.jar</code>	I-Administrator Java archive (JAR) file.
<code>UECAdmin\lib\ueccommon.jar</code>	Common routines shared between I-Administrator and I-Management Console.
<code>UECAdmin\lib\uecumc.jar</code>	I-Management Console resources used by I-Administrator.

Table 4.30 UEC Client Applications - I-Administrator File Inventory List

I-Activity Monitor

File	Description
<code>UAMonitor\uam.jar</code>	I-Activity Monitor Java archive (JAR) file.

Table 4.31 UEC Client Applications - I-Activity Monitor File Inventory List

I-Management Console

File	Description
UMConsole\uecumc.jar	I-Management Console Java archive (JAR) file.
UMConsole\lib\uecommon.jar	Common routines shared between I-Management Console and I-Administrator.

Table 4.32 UEC Client Applications - I-Management Console File Inventory List

System Files

The following files will be installed only if they are newer than the existing file.

File	Description
System32\msiexec.exe	Version 3.1.4000.1823 of the Microsoft Windows Installer (see Section 4.1.3 Windows Installer for more information).

Table 4.33 UEC Client Applications - System Files File Inventory List

4.19 Universal Products for SOA for Windows Package

The Universal Products for SOA 4.1.0 for Windows package includes the following products and their components:

- Universal Products for SOA
 - Universal Application Container Server
 - Universal Application Container
 - Universal Application Interface

4.20 Universal Products for SOA for Windows – Installation Requirements

To install Universal Products for SOA for Windows, you must have one of the following versions of Windows:

- Windows 2000 Professional or Server, with Service Pack 4 or higher
- Windows Server 2003
- Windows XP
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows Server 2008 R2

In addition, you must have:

- Administrator access.
- Possible reboot. A reboot is required if the Windows Installer service is not installed, a version of the Windows Installer prior to 3.1.4000.1283 is installed, or required files are in use at the time of the installation.
- TCP/IP.
- About 110 megabytes of disk space.
- Universal Products 3.2.0.0 or later.

4.21 Universal Products for SOA for Windows – Pre-Installation / Upgrade Backups

Note: The Universal Products for SOA 3.2.0.2 package will upgrade a Universal Command Agent for SOA 3.2.0.0 installation. Subsequent releases of Universal Products for SOA will upgrade all prior releases of Universal Products for SOA (in addition to a Universal Command Agent for SOA 3.2.0.0 installation).

Before upgrading to the latest release of Universal Products for SOA, we recommend stopping any active Universal Application Container (UAC) Server instances (via the Universal Control utility or by stopping the local Universal Broker). If the UAC Server is active during the upgrade, the Windows Installer will require a reboot of the system. Also, stopping the UAC Server before backing up the files listed below will ensure the latest copies of each are captured.

The installation process overwrites the current files, effectively removing your modifications. Backing up these files will optimize the time it takes you to get up and running after installing or upgrading.

The following list identifies the files – and their locations – that should be backed up or copied before you install a new release or upgrade a current release.

UAC.xml File

`%ALLUSERSPROFILE%\Application Data\Universal\uac`

Log4jConfiguration.xml File

`Program Files\Universal\uac (UAC)`

`Program Files\Universal\uai (UAI)`

JMS Provider Client Jar Files

`Program Files\Universal\uac\container\webapps\axis2\WEB-INF\lib`

Note: The names of the jar files differ depending on which JMS Provider you are using.

JMS Provider Client Properties Files

`Program Files\Universal\uai\xml`

Note: These are suggested locations; you can place these files anywhere on the files system. If you have located these files under the `uai` directory, you should back them up.

Payload Files

Normally, payload files should be located on the host system where Universal Command Manager is installed.

If you have chosen to store them elsewhere, the suggested location is:

Program Files\Universal\uai\xml

Note: You can place these files anywhere on the files system. If you have located these files under the **uai** directory, you should back them up.

4.22 Universal Products for SOA for Windows – Installation

Windows Installer provides two interfaces that you can use to install Universal Products for SOA: graphical and command line.

This section provides the following procedures regarding these two interfaces:

- [Installing Universal Products for SOA via the Windows Installer Graphical Interface](#)
- [Modifying a Universal Products for SOA Installation via the Windows Installer Graphical Interface](#)
- [Installing Universal Products for SOA via the Windows Installer Command Line Interface](#)
- [Modifying a Universal Products for SOA Installation via the Windows Installer Command Line Interface](#)

4.22.1 Installing Universal Products for SOA via the Windows Installer Graphical Interface

To install Universal Products for SOA for Windows using the Windows Installer graphical interface, perform the following steps:

1. Download the Universal Products for SOA for Windows product distribution file, `ups-3.2.0.<level>-nt-i386.exe`, to your work station.
2. Execute the distribution file to extract the files.

Note: If you already have extracted the files from the distribution file, but cancelled the installation process in order to install Windows Installer separately (see Section 4.1.3 [Windows Installer](#)), you can simply double-click the extracted Universal Products for SOA installation file, `UPforSOA.msi`, to begin the installation.

Installing Over a Remote Desktop Session

Starting with Windows Server 2003, Remote Desktop provides distinct session environments for each logged-in user. This means the file extraction location may rely on an environment setting that is not available after the Remote Desktop session ends.

For example, the distribution file's default extraction location is based on the **TEMP** environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed.

To ensure that extracted files and other required resources are accessible after the initial install, extract the files to a well-known location that is not likely to change between Remote Desktop sessions.

(Refer to the Microsoft documentation on the Remote Desktop feature for additional information.)

The installation automatically begins after the file extraction completes.

It first will verify that your machine meets the minimum system requirements (see Section 4.20 [Universal Products for SOA for Windows – Installation Requirements](#)). If the requirements are met, a Welcome dialog displays ([Figure 4.43](#)).



Figure 4.43 Universal Products for SOA - Welcome dialog

3. Click the **Next** button. A list of Universal Products for SOA components included in the installation package then displays ([Figure 4.44](#)). It is from this list that you can select which components to install.

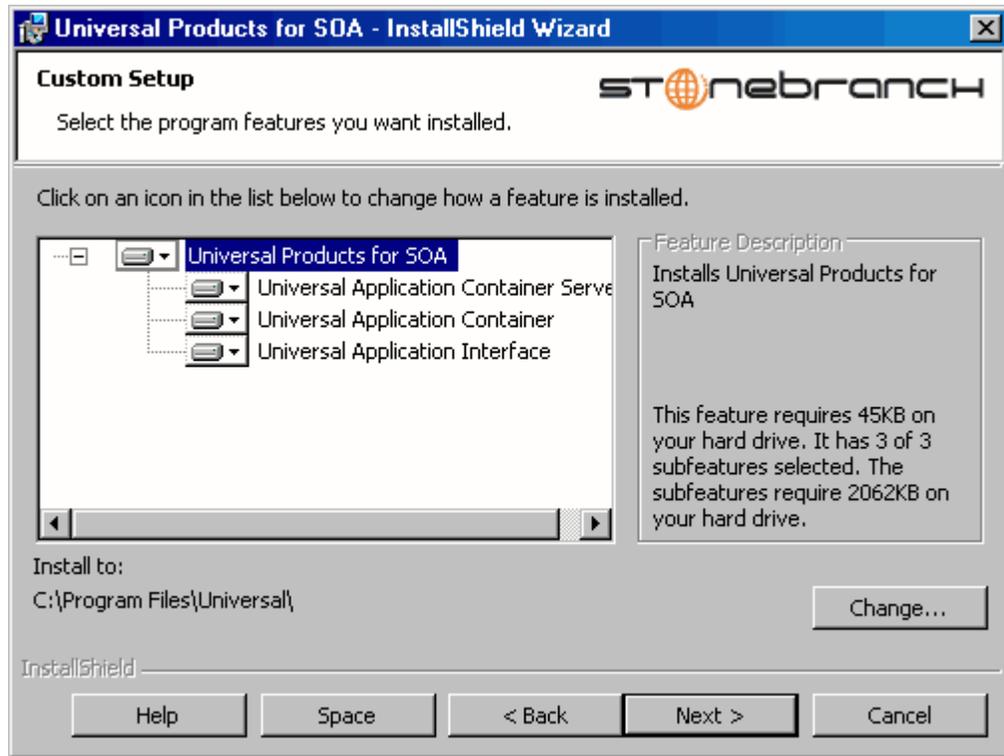


Figure 4.44 Universal Products for SOA - Custom Setup Dialog

For a new installation, a drive icon  displays next to each item in the list, indicating that the component will be installed.

For an upgrade installation, the drive icon  next to each item indicates that the component is either:

- New to the installation and will be installed.
- Currently installed and will be upgraded.



Stonemanch's Tip

If the installation detects an existing Universal Products for SOA installation, currently installed components may be upgraded.

(Currently, there is no way to specify that the state of a currently installed component remain unchanged.)

If a component is selected for installation, and the version of the installed component is below the version of the component being installed, the installed component will be replaced by the component being installed.

4. [Figure 4.44](#) shows that all Universal Products for SOA components will be installed in their respective directories under the `C:\Program Files\Universal` directory.
 - a. If you want to select a different location, click the **Change...** button.
 - b. If you want to check the amount of disk space required for the installation, and the amount of available disk space on the selected directory, click the **Space** button.
5. After verifying the install location, click the **Next>** button to continue the installation process. Follow the directions to complete the installation.

When the installation completes successfully, the Installation Complete dialog ([Figure 4.45](#)) displays.

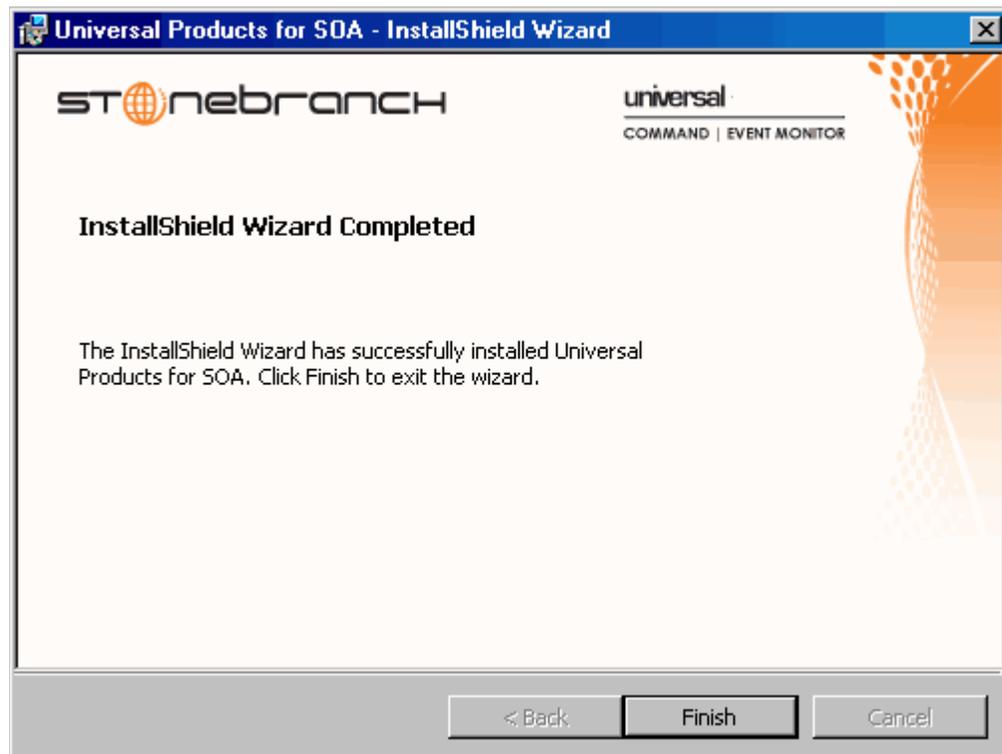


Figure 4.45 Universal Products for SOA - Installation Complete dialog

6. Click the **Finish** button to exit the Windows installation.

4.22.2 Modifying a Universal Products for SOA Installation via the Windows Installer Graphical Interface

This section describes how to modify a Universal Products for SOA installation via the Windows Installer graphical interface.

After installing the Universal Products for SOA, run the installation programs as needed to modify the installation by:

- [Repairing a Corrupted Universal Products for SOA Installation](#).
- [Removing a Universal Products Installation](#).

Note: Although Windows Installer provides a Modify selection for Universal Products for SOA (see [Figure 4.47](#)), it cannot be modified; that is, individual components cannot be added or removed.

Repairing a Corrupted Universal Products for SOA Installation

The installation program has the ability to recover accidentally deleted application files, configuration and component definition files, or registry entries required by Universal Products for SOA. This repair feature will re-install the missing items, making a complete re-install unnecessary.

During a repair, any options stored in configuration and component definition files are preserved. If a configuration or component definition file was deleted, the installation will create a new file with default values.

To repair an installation, perform the following steps:

1. On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog ([Figure 4.46](#)) displays.

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

The Programs and Features dialog has replaced the Add or Remove Programs dialog.

2. From the list of installed programs, select **Universal Products for SOA**.

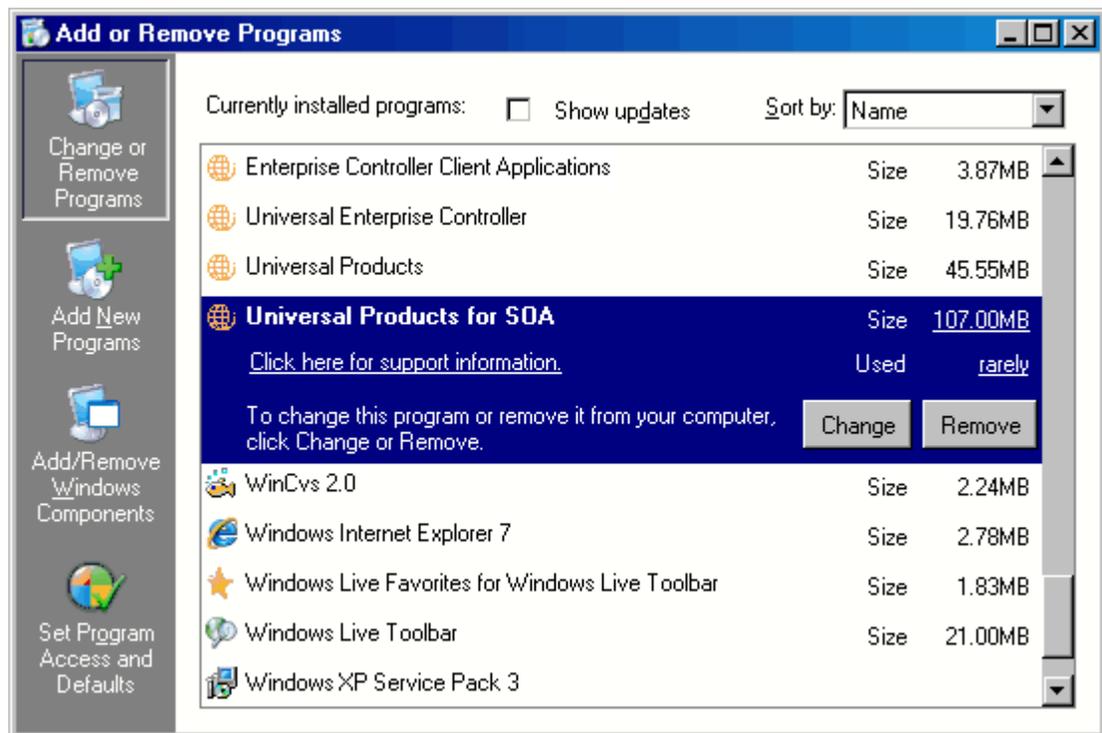


Figure 4.46 Universal Products for SOA - Add or Remove Programs Dialog

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

The Programs and Features dialog (see [Figure 4.7](#)) replaces the Add or Remove Programs dialog.

3. Click the **Change** button to start the installation program.

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

To skip the remaining steps, simply select the **Repair** button (see [Figure 4.7](#)).

4. On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays ([Figure 4.47](#), below).

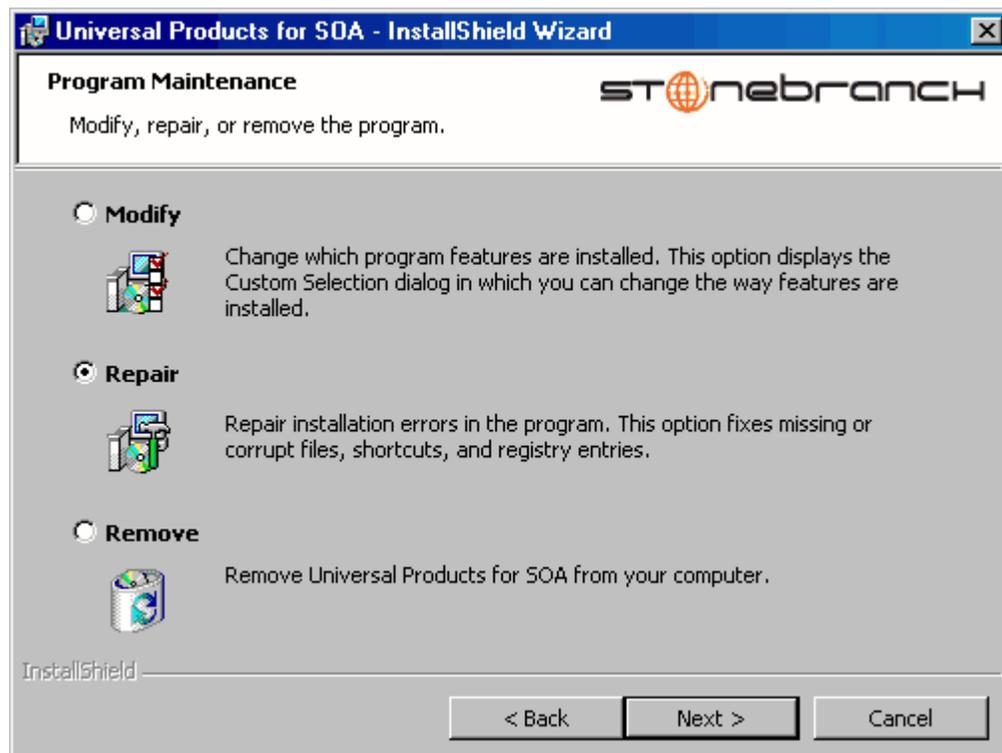


Figure 4.47 Universal Products for SOA - Program Maintenance / Repair dialog

5. Select **Repair** and click the **Next>** button to display the Ready to Repair dialog.
6. Follow the instructions in successive dialogs to complete the repair.

Removing a Universal Products for SOA Installation

To uninstall a Universal Products for SOA installation, perform the following steps:

1. On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays (see [Figure 4.46](#)).

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

The Programs and Features dialog (see [Figure 4.7](#)) replaces the Add or Remove Programs dialog.

2. From the list of installed programs, select Universal Products for SOA.
3. Click the **Change** button to start the installation program.
(To skip the remaining steps, simply select the **Remove** button.)

Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

To skip the remaining steps, simply select the **Uninstall** button.

4. On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays ([Figure 4.9](#), below).

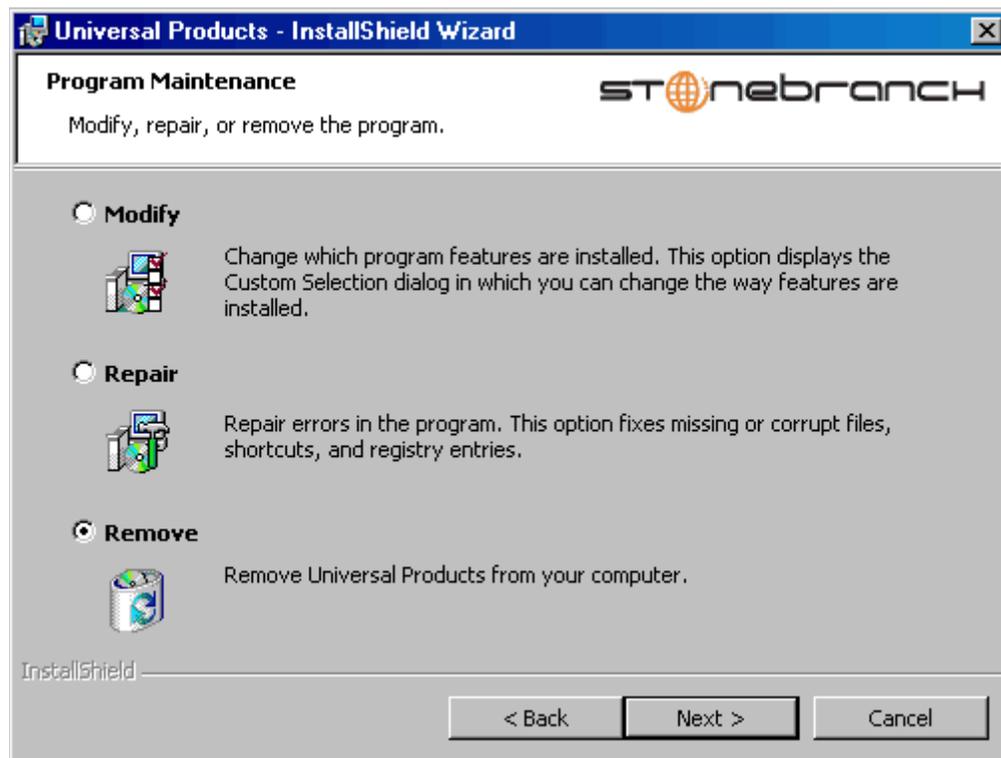


Figure 4.48 Universal Products for SOA - Program Maintenance / Remove dialog

5. Select **Remove** and click the **Next>** button.

6. On the subsequent dialog, click the **Remove** button to remove the Universal Products for SOA installation.

Un-Installed Files

The uninstall process will remove only those files created during the installation. Some files stored under the `.\Universal` install directory by Universal Products for SOA, such as trace files, may be left behind after the uninstall. In this situation, simply delete the uninstalled files and/or directories.

Before deleting the entire `.\Universal` directory, make sure it does not contain application files for any other installed Stonebranch, Inc. product. (See Section [4.25 Universal Products for SOA for Windows – File Inventory Lists](#) for a list of files and directories installed with Universal Products for SOA.)

4.22.3 Installing Universal Products for SOA via the Windows Installer Command Line Interface

This section describes how to install Universal Products for SOA using the Windows Installer command line interface.

The command line interface is useful in situations where:

- Several Universal Products for SOA installations must be deployed across many different systems.
- It is not practical or convenient to perform a graphical interface installation.

To use the Windows Installer command line interface, the Universal Products for SOA installation files first must be extracted from the product distribution file. Since there is no command line option available to unpack the distribution file, this must be done via the graphical user interface:

1. Download the Universal Products for SOA for Windows product distribution file, `ups-4.1.0.<level>-nt-i386.exe`, to your work station.
2. Execute the distribution file to extract the installation files.
3. On the Location to Save Files dialog (Figure 4.49, below), select a location in which to store the installation files and click the **Next>** button to extract the files.

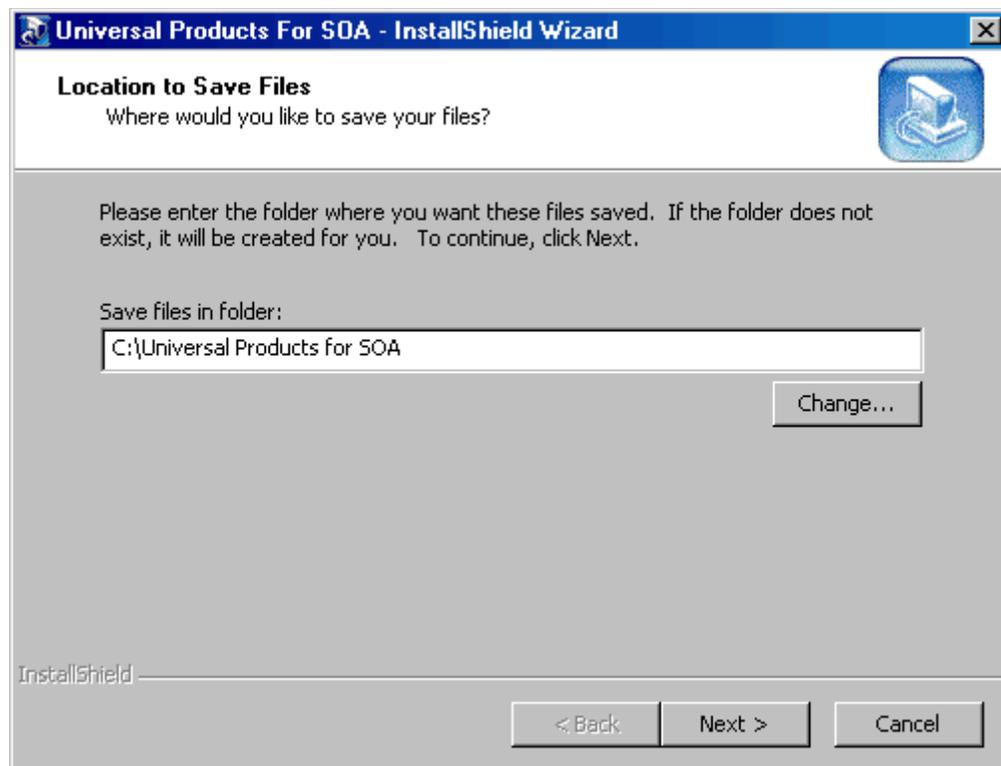


Figure 4.49 Universal Products for SOA - Location to Save Files dialog

4. On the Welcome dialog (see [Figure 4.43](#)), click the **Cancel** button.

At this point, all the installation files have been extracted into the location specified, but nothing has been installed. You may now complete the install from the command line, with no additional graphical interaction required.

Typically, to install an application that uses Windows Installer, a program such as **setup.exe** is executed. This program usually checks to make sure that a supported version of Windows Installer is installed. If one is not, the program will install it before proceeding. In fact, this is exactly what happens when Universal Products for SOA for Windows is installed from the product distribution file.

There is a command line interface available for **setup.exe**, but running it is not really ideal for unattended installations. If Windows Installer is installed or upgraded on the target system, a reboot may be required before the installation of Universal Products for SOA can continue (see [Determining if Windows Installer will be Installed or Upgraded](#) in [Section 4.1.3 Windows Installer](#)).

A better way to install Universal Products for SOA from the command line is to execute Windows Installer, **msiexec.exe** (see [Windows Installer Installation Commands](#) in this section).

Note: A silent install also can be performed from the Universal Products for SOA for Windows distribution file (see [Using the Distribution File for a Silent Install](#) in this section).

Windows Installer Command Line Parameters

Table 4.34, below, describes the parameters available when using Windows Installer to install Universal Products for SOA from the command line.

Parameter	Description
/i	Installs Universal Products for SOA. This parameter cannot be used with either the /f (repair) or /x (remove) parameter.
/fom	Repairs a Universal Products for SOA installation. The om after the /f are actually options used by the repair. There are other options available, but the om options result in behavior that matches a Repair initiated via the graphical install. This parameter cannot be used with either the /i (install) or /x (remove) parameter.
/x	Removes Universal Products for SOA. This parameter cannot be used with either the /i (install) or /f (repair) parameter.
SetupPath\ UPforSOA.msi	Specifies the path to the UPforSOA.msi file. SetupPath is the directory in which files extracted from the downloaded distribution package reside.
INSTALLDIR	Sets the root installation directory to InstallDir. All components are installed in this directory. This optional parameter is required only if you want to install Universal Products for SOA under a directory different from the one specified by the PROGRAMFILES environment variable (typically C:\Program Files\Universal). This parameter is valid only when the /i switch is used. It is ignored for other options. If the directory contains spaces, you must use double (") quotation marks around the path name.
/q	Instructs Windows Installer to run without a graphical interface (silent install). If this optional parameter is omitted, it allows the Universal Products for SOA installation to be started from the command line, but run with a graphical interface. This is useful when an installation log file is desired. See Windows Installer Command Line Syntax , Windows Installer Installation Commands , and Detecting the Completion of a Silent Install in this section for additional information regarding silent installs.
/L*v	Instructs Windows Installer to create an installation log file with a name of Logfilepath. If Logfilepath contains spaces, you must use double (") quotation marks around the path name. The *v after the /L actually are flags used to specify the level of detail contained in the log file. To reduce the amount of output generated, omit these options. However, using these options is good practice, and can assist Stonebranch Customer Support with problem determination should any errors occur during installation. This is an optional parameter.

Table 4.34 Universal Products for SOA for Windows - Windows Installer Command Line Parameters

Windows Installer Command Line Syntax

<Insert new figure/>, below, illustrates the command line syntax when using Windows Installer to install Universal Products for SOA from the command line.

```
msiexec.exe [/i|/fom|/x] SetupPath\UPforSOA.msi  
[INSTALLDIR=InstallDir]  
[/q] [/L*v Logfilepath]
```

Figure 4.50 Universal Products for SOA for Windows – Windows Installer Command Line Syntax

The order in which the parameters are specified is significant.

The **/i**, **/f**, or **/x** parameter must be first, followed by the name of the installation file, **UPforSOA.msi**.

The **/q** and **/L** switches can be specified in any order, but, when used, must come after all other parameters.

Windows Installer Installation Commands

The following commands allow installation of Universal Products for SOA from the command line using Windows Installer (**msiexec.exe**).

To execute an interactive Universal Products for SOA installation, issue the following command:

```
msiexec.exe /i SetupPath\UPforSOA.msi
```

To generate a log file named **C:\Temp\install.log** while executing an interactive Universal Products for SOA installation, issue the following command:

```
msiexec.exe /i SetupPath\UPforSOA.msi /L C:\Temp\install.log
```

To install Universal Products for SOA silently to a non-default location, issue the following command:

```
msiexec.exe /i SetupPath\UPforSOA.msi INSTALLDIR=D:\Universal /q
```

To install Universal Products for SOA silently, accepting all defaults, issue the following command:

```
msiexec.exe /i SetupPath\UPforSOA.msi /q
```

Detecting the Completion of a Silent Install

When the `/q` switch is used to perform a silent install, no graphical interface or user interaction is required. One drawback to this is that no feedback is provided indicating when the Windows Installer process (install, uninstall, or repair) finishes.

One method that can be used to detect when the Windows Installer process ends is to execute it using the system's `start` command. Using available command line switches, `start` can initiate the Windows Installer process and then wait for it to finish. When `start` returns control to its calling process (for example, the command prompt), the process will have ended.

For example, from the command prompt, issue the following command to start the Universal Products for SOA installation and wait for it to finish.

```
start /b /wait msiexec.exe /i SetupPath\UPforSOA.msi /q
```

- The `/b` switch prevents the `start` command from opening a new window.
- The `/wait` parameter causes the `start` command to start Windows Installer (`msiexec.exe`) and then wait for it to finish.

You also can use the syntax above to execute the `start` command from within a script, such as a `.bat` file.

For more information on the `start` command, open a Windows command prompt and enter: `start /?`.

Using the Distribution File for a Silent Install

To perform a silent install using the Universal Products for SOA for Windows distribution file, issue the following command:

```
ups-4.1.0.<level>-nt-i386.exe /s /a /s /w /v" /qn"
```

The switches (from left to right) are as follows:

- `/s` "Package for the Web" switch for silent install.
- `/a` "Package for the Web" switch to "add" commands (that is, send the next commands to `setup.exe`).
- `/s` `setup.exe` switch for silent install.
- `/w` Wait switch, for a Basic MSI project, forces `setup.exe` to wait until the installation is complete before exiting.
- `/v` `setup.exe` switch to pass arguments or options to the MSI package. Quotes must be in place if a space-separated command is passed to the MSI.
- `/qn` `msiexec.exe` switch for quiet install with no graphic display of progress.

4.22.4 Modifying a Universal Products for SOA Installation via the Windows Installer Command Line Interface

This section describes how to modify a Universal Products for SOA installation via the Windows Installer command line interface.

After installing Universal Products for SOA, you may rerun the installation program as needed in order to:

- Repair a Corrupted Universal Products for SOA Installation.
- Remove a Universal Products for SOA Installation.

(For a description of the parameters used in these procedures, see [Windows Installer Command Line Parameters](#).)

Repairing a Corrupted Universal Products for SOA Installation

To recover accidentally deleted application files, configuration and component definition files, or registry entries required by Universal Products for SOA using the Windows Installer command line interface, use the `/f` switch together with the `om` parameters.

These are the same repair options set internally by the graphical interface installation. They cause the installation program to reinstall files that are missing or are older than the version contained in the Universal Products for SOA distribution file.

To repair a Universal Products for SOA installation from the command line, issue the following command:

```
msiexec.exe /fom SetupPath\UPforSOA.msi /q
```

To repair a Universal Products for SOA installation interactively while writing a log file to `C:\Temp\repair.log`, issue the following command:

```
msiexec.exe /fom SetupPath\UPforSOA.msi /! C:\Temp\repair.log
```

Removing a Universal Products for SOA Installation

To uninstall a Universal Products for SOA installation using the Windows Installer command line interface, use the `/x` switch. Issue the following command:

```
msiexec.exe /x SetupPath\UPforSOA.msi /q
```

4.23 Universal Products for SOA for Windows – Configuring and Starting the Universal Application Container Server

After installing Universal Products for SOA for Windows, perform the following steps before attempting to configure and start the Universal Application Container Server.

(For more information, see the UAC Server section in the Universal Products for SOA 4.1.0 User Guide.)

1. Refresh the local Universal Broker's cached configuration.

Starting with version 3.2.0.0, Universal Broker caches configuration and component definition information for all Universal Product applications. Before UAC Server will run, its configuration and component definition information must reside in the local Broker's cache.

To refresh the local Broker's cache without stopping and restarting the Broker, issue a REFRESH command via Universal Control (For more information, see the Universal Control chapter in the Universal Products Utilities 4.1.0 User Guide.)

2. Add Universal Application Container (UAC) Server license information to the UAC Server configuration file.

Stonebranch, Inc. provides UAC Server license information in a formatted text file. This file contains the required license parameter keywords and values. You can simply append this license information to the UAC Server configuration file, `uacs.conf`, or import it using the Universal Configuration Manager.

To import license information using the Universal Configuration Manager.

- a. Open the Windows Control Panel to start the Universal Configuration Manager.
- b. In the Installed Components tree, click on the + symbol next to Universal Application Container Server to display its property pages.
- c. Select License Information to display the License Information property page (see [Figure 4.51](#), below).

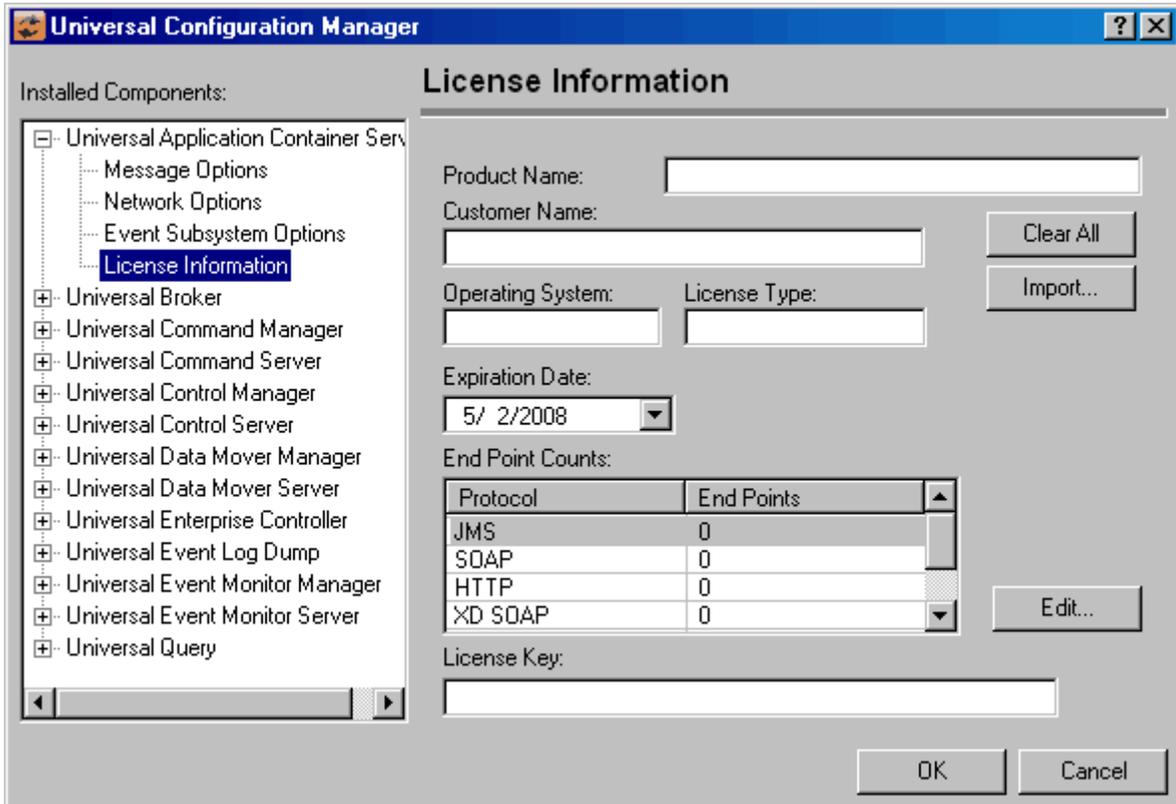


Figure 4.51 Universal Configuration Manager - Universal Application Container Server - License Information

- d. If you want to enter the license information manually on this page, make sure that you enter the information exactly as it appears in the license information file.
 - All information is case-sensitive.
 - You must enter all punctuation and spacing as provided in the file.
 Instead of entering the information manually, you can import the license file by clicking the **Import** button, and then opening the file from the Select License File dialog.
- e. After entering or importing the license information, click the **OK** button to save the information in the UAC Server configuration file.

3. Recycle the local Universal Broker to start the UAC Server.

By default, the UAC Server is set to start automatically whenever its local Universal Broker starts. Stopping and restarting the Universal Broker will cause UAC Server to start.

4.24 Universal Products for SOA for Windows – Installing on 64-bit Windows Editions

All Universal Products for SOA components have been tested and verified on the 64-bit editions of the following Windows systems:

- Windows XP
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows Server 2008 R2

The installation defaults for Universal Products for SOA should not require any modification when installing on 64-bit Windows editions.

4.25 Universal Products for SOA for Windows – File Inventory Lists

The Universal Products for SOA 4.1.0 for Windows package includes the files required for the following components / utilities:

- Universal Application Container Server
- Universal Application Container
- Universal Application Interface
- Sun Java Runtime Environment, 1.5.0_14

This section lists the files installed with each Universal Products for SOA component. The file paths specified are relative to the root installation directory (for example, **C:\Program Files\Universal**) that was specified during the installation.

Universal Application Container Server

File	Description
UAC\startUAC.bat	Starts the Universal Application Container.
UAC\shutdownUAC.bat	Stops the Universal Application Container.
UAC\UAC.xsd	File used by Universal Application Container to validate content of the UAC.xml inbound configuration file. It requires no user interaction.
UAC\derby.log	Universal Application Container database log file.
UAC\log4jConfiguration.xml	Universal Application Container logging configuration file that you may need to modify if you want to change the logging level of the UAC component.
UAC\INTEventLogAppender.dll	Universal Application Container Windows logging DLL.
UAC\uacValidateInbound.bat	Script that validates the contents of the UAC.xml file using the schema definition contained in UAC.xsd .
UAC\bin\uacsrv.exe	Universal Application Container Server program file.
UAC\bin\uacscfg.dll	Used by Universal Configuration Manager to manage Universal Application Container Server configuration options.
UAC\bin\uacscfg.hlp	Universal Application Container Server configuration help file.
UAC\lib\uacsrv.jar	Universal Application Container Server class file.
UBroker\tml\uaccfg	Universal Application Container Server configuration template file.
UBroker\tml\uaccmp	Universal Application Container Server component definition template file.
nls\uacmceng.umc	Universal Application Container Server message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\conf\UAC.xml	Inbound operation configuration file that you may need to modify to support your inbound workload operations.

File	Description
%ALLUSERSPROFILE%\Application Data\Universal\confluacs.conf	Universal Application Container Server configuration file.
%ALLUSERSPROFILE%\Application Data\Universal\comp\uac	Universal Application Container Server component definition file.
%ALLUSERSPROFILE%\ Application Data, by default, resolves to: <ul style="list-style-type: none"> • 'C:\Documents and Settings\All Users\Application Data' on Windows 2000, 2003, and XP. • 'C:\ProgramData' on Windows Vista / Windows 2008 Server and later. 	

Universal Application Container

File	Description
UAC\container	Subdirectory containing the libraries and other deployable objects needed for UAC operation.

Universal Application Interface

File	Description
UAI\lib	Subdirectory containing the uacsrv.jar file needed for communication between the uacsrv and uac components. It requires no user interaction.
UAI\keystore	File that contains the certificates for secure communication between UAC and UAI.
UAI\log4jConfiguration.xml	Universal Application Interface logging configuration file.
UAI\uai.bat	Universal Application Interface start-up script.

Sun Java Runtime Environment, 1.5.0_14

File	Description
jre*.*	Files required for redistribution of the 1.5.0_14 JRE.

System Files

The following files will be installed only if they are newer than the existing file.

The directories shown in [Table 4.22](#), below, are relative to the %SYSTEMROOT% directory, where %SYSTEMROOT% is an environment variable that resolves to C:\WINNT on Windows 2000 and C:\Windows on all other Windows platforms.

File	Description
System32\asycfilt.dll	Version 2.40.4275.1. This DLL is one of the components of the Microsoft OLE library.
System32\comcat.dll	Version 4.71.1460.1 of the Microsoft Component Category Manager library.
Microsoft C-Runtime v8.0.50727.762 ¹	Version 8.0.50727.762 of the Microsoft C runtime side-by-side assembly.
Microsoft Foundation Classes v8.0.50727.762 ²	Version 8.0.50727.762 of the Microsoft Foundation Class (MFC) side-by-side assembly.
System32\msiexec.exe	Version 3.1.4000.1823 of the Microsoft Windows Installer (see Section 4.4 Universal Products for Windows – Installation for more information).
System32\oleaut32.dll	Version 2.40.4275.1. This DLL is one of the components of the Microsoft OLE library.
System32\olepro32.dll	Version 5.0.4275.1. This DLL is one of the components of the Microsoft OLE library.
System32\psapi.dll	Version 4.0.1371.1 of the Microsoft process status library
System32\stdole2.tlb	Version 2.40.4275.1. This file is one of the components of the Microsoft OLE library.
<p>1 The Microsoft C-Runtime distribution consists of several files, which are subject to change. Refer to Microsoft documentation for a complete list of files delivered with the specified runtime version.</p> <p>2 The Microsoft Foundation Classes (MFC) distribution consists of several files, which are subject to change. Refer to Microsoft documentation for a complete list of files delivered with the specified MFC version.</p>	

Table 4.35 Universal Products for SOA for Windows System Files – File Inventory List

Chapter 5

UNIX Installation

5.1 Overview

This chapter provides information on the installation of Stonebranch, Inc. Universal Products 4.1.0 on UNIX operating systems. Unless otherwise specified, all references to Universal Products for UNIX in this chapter refer to version 4.1.0.

Note: This chapter does not include installation on z/OS UNIX System Services (USS). See [Chapter 3 z/OS USS Installation](#) for z/OS USS installation instructions.

Universal Products 4.1.0 for UNIX are provided in two separate easily installed, one-time installation packages:

- [Universal Products for UNIX Package](#)
- [Universal Products for SOA for UNIX Package](#)

Note: Starting with the 3.2.0 release of Universal Products, a Universal Broker must run on all systems on which a Universal Product component is running, including manager components. The Broker maintains product configuration data for all components that have a configuration file.

5.1.1 Installation Methods

The installation method depends on which type of UNIX operating system the package is being installed. Vendor-recommended packaging of the vendor's own recommended installation methods is used for the following supported UNIX operating systems:

- AIX 5.2 and above
- HP-UX 11.11 and above
- Solaris 8 and above
- Linux
 - 2.4 kernel for Intel-based (x86) and IBMS/390 (and zSeries) systems
 - 2.6 kernel for Itanium (ia64) systems
 - Redhat Package Manager (RPM)

Universal Products for UNIX for all other types of UNIX operating systems are installed with the generic Universal Products for UNIX installation script.

5.2 Universal Products for UNIX Package

The Universal Products for UNIX package includes the following product components:

- Universal Broker
- Universal Certificate 4.1.0
- Universal Command Manager and Server 4.1.0
- Universal Connector 4.1.0 (for AIX, HP-UX, and Solaris)
- Universal Control Manager and Server 4.1.0
- Universal Data Mover Manager and Server 4.1.0
- Universal Encrypt 4.1.0
- Universal Event Monitor Manager and Server 4.1.0
- Universal Message to Exit Code Translator 4.1.0
- Universal Query 4.1.0

5.2.1 Product Compatibility

[Table 5.1](#), below, identifies the compatibility of Universal Products 4.1.0 for UNIX with previous product versions.

Product	Compatibility
Universal Command 4.1.0	Universal Command 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Control 4.1.0	Universal Control 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0.
Universal Data Mover 4.1.0	Universal Data Mover 3.2.0, 3.1.1 and 3.1.0.
Universal Encrypt 4.1.0	Universal Encrypt 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0.
Universal Query 4.1.0	Universal Broker 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0.

Table 5.1 Universal Products 4.1.0 for UNIX - Product Compatibility

The product references pertain to all supported UNIX operating systems for that version.

5.3 Universal Products for UNIX – Installation Requirements

Universal Products for UNIX installation requirements are:

- TCP/IP Socket implementation.
- Approximately 50 megabytes of disk space for the installation. More disk space is required for variable files, such as log files, spool files, and trace files.
- Bourne shell or compatible.
- Superuser (root) access.

Note: In order to install these products, you must be able to write to the directory from which the installation is launched.

5.3.1 Directories and Files

All product files that are written to during product execution are stored in the `/var/opt/universa1` directory by default. This section documents the estimated amount of space required on the file system for all directories (and their sub-directories), required security access, and mount requirements.

Command Reference Directory

Universal Command Server can execute commands of type `cmdref`. A command reference is a predefined command or script to which the Universal Command Manager refers by its file name.

The command reference directory is `/var/opt/universa1/cmdref`.

Space

The amount of space required is solely dependent on the number of command reference files you define. No command reference files are included in the installation.

Security

Universal Command Server requires read access to the `cmdref` directory. The product administrator requires read/write access in order to maintain the command reference files. No general user access is required.

Log Directory

Universal Broker can be configured to write its messages to a log file.

The current log file and previous log file generations are stored in the `/var/opt/universal/log` directory.

Space

A log file size grows to about 150,000 bytes and is then rolled over to a generation file. Five generations of log files are saved. The oldest generation log file is deleted. The amount of space required for the five generations and the current log file is about 900,000 bytes.

Security

Universal Broker requires read/write access to the log directory and read/write access to all files in the log directory. No other Universal Products use the log directory at this time. No general user access is required.

Trace Directory

Universal Broker and its server components (for example, Universal Command Server) create product trace files when configured to do so. A trace file is used by Stonebranch, Inc. [Customer Support](#) to resolve product problems.

Space

Trace files can grow to significant size depending on how long the trace is active and how much work the program is doing during the tracing period. A trace file size of about 10MB is not unusual.

Trace file sizes can be limited by setting the maximum number of lines to write to a file with the `MAX_TRACE_LINES` configuration option. Once the maximum is reached, the trace file will wrap to the beginning.

Under normal operation, no space is required for trace files. Trace files are requested by Stonebranch, Inc. [Customer Support](#) only for problem resolution. When trace files are required, at least 20MB of disk space should be available.

Security

Universal Broker and the Broker components (Universal Command Server and Universal Control Server) require read/write access to the trace directory. No other Universal Products access the trace directory. No general user access is required.

Spool Directory

The spool directory is used to store the following types of information:

- Execution information for Universal components started by Universal Broker.
- Event definitions and event handlers managed by Universal Broker and used by Universal Event Monitor.
- Results of events tracked by Universal Event Monitor.
- Redirected standard I/O files (stdin, stdout, and stderr) captured by Universal Command when run with manager fault tolerance enabled.

The default location for the spooled standard I/O files and other database files is the `/var/opt/universa1/spoo1` directory.

Space

Spool files are located, by default, in directory `/var/opt/universa1/spoo1`. The location of various product files can be configured via the product configuration files.

The amount of disk space required for the spool directory depends on:

1. Number of spooling user processes that will be executing simultaneously. A user process is created for each command requested by a Universal Command Manager. The default maximum number is 50.

When a user process ends and a Manager has received all the spool files, the spool files themselves are deleted.

2. Average size of the user processes standard input, output, and error files.



Stoneman's Tip

Keep in mind that spooling is not a feature for file transfer purposes. File transfer-related processes should execute without spooling enabled.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

$\text{MAX-PROCESSES} \times \text{AVERAGE-STDIO-SIZE} \times 2 = \text{required disk space.}$

For example, if the maximum number of simultaneous user processes is estimated at 20 and the average size of processes standard I/O files is 100,000 bytes, the average amount of required disk space is 4MB (20 x 100000 x 2).

The Universal Command Server is configured with spooling disabled to prevent unintentional disk utilization. The feature must be turned on through the `ALLOW_SPOOLING` configuration settings.

Further details on the Manager Fault Tolerant feature and the spooling of redirected standard I/O files can be found in the Universal Command 4.1.0 User Guide.

Security

Universal Broker, Universal Command Server and Universal Event Monitor require read/write access to the spool directory. No other Universal Products access the spool directory. No general user access is required.

Mount

The spool directory must be mounted on a local device. It cannot be mounted on a network device, such as an NFS or SAMBA mount.

5.4 Universal Products for UNIX – Installation Upgrades

This section describes changes in the product installation that have occurred with new versions of the product.

If a particular version is absent from the list, either:

- No change occurred.
- Version is no longer supported.

5.4.1 Universal Command 2.2.0

Installation Method

Starting with Universal Command 2.2.0, the generic Universal Products for UNIX installation script is no longer used for AIX, HP-UX, Solaris, and Linux installs. The native operating system vendor-recommended packaging methods are used on these operating systems. All other UNIX operating systems continue to use the generic installation script.

Product Directories

All Universal Products now use the Filesystem Hierarchy Standard (FHS) 2.2 for product directory names and locations. This conforms to most vendor directory implementations. Consequently, the installation directory is `/opt/universa1`, the variable files directory is `/var/opt/universa1`, and the configuration files directory is `/etc/universa1`.

Executable files from versions prior to Universal Command 2.2.0 (**ucmd**, **uct1**, **uquery**, **umet**, **uencrypt**) require manual removal from their default installation directory, `/usr/local/bin`. Previous releases of Universal Command copied executable programs to the `/usr/local/bin` directory by default.

Configuration Files

The installation methods obtain the location where the configuration files for each of the Universal Products are stored. Configuration files will be moved from their current location to the `/etc/universa1` directory.

5.4.2 All Universal Product Releases

All configuration files (for example, `ucmd.conf` and `ubroker.conf`) are updated and not replaced. Local modifications to the configuration files are preserved.

Note: With the generic Universal Products for UNIX installation script, the previous installed directories (default or user-specified) are not removed by the installation of Universal Products 3.1.0 or above.

With the AIX, HP-UX, Solaris, and Linux installs, the previous installation directories are removed when the native installer has detected that additional directories or files have not been added to the original installation directories. If they have been modified, the directories will remain; they can be reviewed and removed when desired by your Administrator.

5.5 Universal Products for UNIX – Product Distribution File

Stonebranch, Inc. provides different Universal Products 4.1.0 for UNIX packages for different types of UNIX operating systems.

To obtain the Universal Products 4.1.0 for UNIX package for your type of UNIX operating system, you must download the corresponding product distribution file from the Support / Current Products area of the Stonebranch, Inc. website: <http://www.stonebranch.com>.

Note: A customer user ID and password — provided by Stonebranch, Inc. — are required to access this area.

After a distribution file has been downloaded, the installation files contained in that file must be extracted before the product can be installed (see [5.6 Universal Products for UNIX – Installation](#)).

5.5.1 Distribution File Format

The name of each Universal Products for UNIX distribution file has the following format:

```
unv-Version.Release.Modification Level.Maintenance Level-operating  
system-version(.release) (-platform) .tar.Z
```

For example: `unv-4.1.0.1-linux-2.6-ia64.tar.Z`

In this format:

- **V**ersion is the current version of Universal Products.
- **R**elease is the current release of Universal Products.
- **M**odification **L**evel is the current Universal Products feature set.
- **M**aintenance **L**evel is the Universal Products build level.
- **o**perating **s**ystem is the name of the operating system (for example, AIX or Linux).
- **v**ersion(**.r**elease) is the supported version and, optionally, the release of the operating system.
- **p**latform is the targeted hardware platform (for example, RS6000 or i386). It is included in the file name only if there is more than one platform available for the specified operating system.

5.6 Universal Products for UNIX – Installation

This section provides the following UNIX operating systems installation procedures:

- [Universal Products for AIX Installation](#)
- [Universal Products for HP-UX Installation](#)
- [Universal Products for Solaris Installation](#)
 - [Intel-Based Systems](#)
 - [SPARC-Based Systems](#)
- [Universal Products for Linux Installation](#)
 - [Intel-Based Systems](#)
 - [IBM S/390 and zSeries Systems](#)
 - [Itanium Systems](#)
- [Universal Products for UNIX Installation \(Generic\)](#)

Note: This installation procedure uses the generic Universal Products for UNIX installation script to install Universal Products on all UNIX operating systems other than AIX, HP-UX, Solaris, and Linux.

5.6.1 Universal Products for AIX Installation

Installation of Universal Products for AIX is a three-step process:

1. Download product distribution file (see Section [5.5 Universal Products for UNIX – Product Distribution File](#)).
2. Extract the installation files from the distribution file.
3. Install the extracted files.

Extracting the Universal Products for AIX Installation Files

The Universal Products for AIX product distribution file is in a compressed `tar` format.

To uncompress and extract the installation files from the distribution file, issue the following command:

```
zcat unv-4.1.0.0-aix-5.2.tar.Z | tar xvf -
```

This command assumes the following:

- Name of the distribution file is `unv-4.1.0.0-aix-5.2.tar.Z`.
- File is located in the current working directory.

The distribution file contains multiple files, including packages in the AIX backup file format, using extension `.bff`. The actual base name of the `.bff` file depends on the AIX version for which the distribution file is intended. (See Section [5.5.1 Distribution File Format](#) for distribution file naming conventions.)

[Table 5.2](#), below, identifies the files contained in the distribution file.

File	Description
<code>Readme.unv</code>	Summary of the installation procedure.
<code>ubr-4.1.0.0-aix-5.2.bff</code>	AIX backup file format package for the base UBROKER product.
<code>ucmd-4.1.0.0-aix-5.2.bff</code>	AIX backup file format package for the base UCMD product.
<code>ucom-4.1.0.0-aix-5.2.bff</code>	AIX backup file format package for the base COMMON package.
<code>ucr-4.1.0.0-aix-5.2.bff</code>	AIX backup file format package for the base UCERT package.
<code>udm-4.1.0.0-aix-5.2.bff</code>	AIX backup file format package for the base UDM product.
<code>uem-4.1.0.0-aix-5.2.bff</code>	AIX backup file format package for the base UEM product.
<code>unvinst</code>	Installation script.
<code>upimerge.sh</code>	Script that uses the Universal Installation Merge (UIM) module.
<code>usap-4.1.0.0-aix-5.2.bff</code>	AIX backup file format package for the base USAP product.
<code>utl-4.1.0.0-aix-5.2.bff</code>	AIX backup file format package for the base UTILITIES package.

Table 5.2 Universal Products for AIX - Product Distribution File Contents

If your AIX `tar` file does not contain these files, contact Stonebranch, Inc. [Customer Support](#) to obtain the correct distribution file.

AIX Packages

Table 5.3, below, identifies the filesets contained in the AIX packages.

Package	Fileset Name	Description
UNVcom	UNVcom.UNVcom.msg.en_US	Universal Products common files, such as message catalogs and translation tables.
UNVubr	UNVubr.UNVct1.rte UNVubr.UNVcts.rte UNVubr.UNVqry.rte UNVubr.UNVsp1.rte UNVubr.UNVubr.rte	Universal Control Manager Universal Control Server Universal Query Manager Universal spool maintenance utilities Universal Products Broker
UNVucom	UNVucom.UNVucom.rte UNVucom.UNVuucs.rte	Universal Command Manager Universal Command Server
UNVuocr	UNVuocr.UNVuocr.rte	Universal Certificate
UNVudm	UNVudm.UNVdmgr.rte UNVudm.UNVdsrv.rte	Universal Data Mover Manager Universal Data Mover Server
UNVuem	UNVuem.UNVuem.rte UNVuem.UNVues.rte	Universal Event Monitor Manager Universal Event Monitor Server
UNVusp	UNVusp.UNVusp.rte	Universal Connector
UNVutil	UNVut1.UNVutil.rte	Universal Utilities

Table 5.3 Universal Products AIX - Package Filesets

Installing Universal Products for AIX

Universal Products for AIX is installed with the `unvinst` script, which executes the `installp` command. The command to start the script must be executed with the superuser ID.

Note: Stonebranch, Inc. strongly recommends the use of the `unvinst` script for the AIX installation above any other method.

To install all of the AIX packages, issue the following command:

```
sh ./unvinst
```

This is a silent install. The output from `unvinst` is written to file `install.log` in the current directory.

An entry is added to the system initialization table, `/etc/inittab`, to start the `ubrokerd` daemon when the system boots to runlevel 2. The `inittab` entry is similar to the following:

```
ubroker:2:once:/opt/universal/ubroker-4.1.0/ubrokerd start
```

Note: With the Solaris, HP, and AIX installs, the previous installation directories are removed when the native installer has detected that additional directories or files have not been added to the original installation directories. If they have been modified, the directories will remain and can be reviewed and removed, as desired, by your Administrator.

Removing Universal Products for AIX

(Before removing Universal Products for AIX, stop the `ubrokerd` daemon.)

Universal Products for AIX is removed with the `installp` command. The command must be executed with the superuser ID.

To remove all filesets of AIX, issue the following command:

```
installp -u UNVusp UNVucr UNVut1 UNVuem UNVudm UNVucm UNVubr UNVcom
```

All entries in the `/etc/inittab` file that reference `ubroker` are removed.

Listing Universal Products for AIX Information

On AIX, information on an installed product is listed with the `ls1pp` command. The command must be executed with the superuser ID.

Issue the following command to list information for Universal Products for AIX:

```
ls1pp -La 'UNV*'
```

5.6.2 Universal Products for HP-UX Installation

Installation of Universal Products for HP-UX is a three-step process:

1. Download product distribution file (see Section [5.5 Universal Products for UNIX – Product Distribution File](#)).
2. Extract the installation files from the distribution file.
3. Install the extracted files.

Pre-Requisite to HP-UX Installation

In order to install and run Universal Products for HP-UX, you first must install the following patches (these versions or newer):

- **GOLDQPK11i_B.11.11.0612.459**, which contains these two patch sets:
 - **GOLDAPPS11iB.11.11.0612.459**
 - **GOLDBASE11iB.11.11.0612.459**

Extracting the Universal Products for HP-UX Installation Files

The Universal Products for HP-UX product distribution file is in a compressed **tar** format.

To uncompress and extract the installation files from the distribution file, issue the following command:

```
zcat unv-4.1.0.0-hpux-11.11.tar.Z | tar xvf -
```

This command assumes the following:

- Name of the distribution file is **unv-4.1.0.0-hpux-11.11.tar.Z**.
- File is located in the current working directory.

The distribution file contains multiple files, including a package in the HP-UX file format (extension **.depot**). The actual base name of the **.depot** file depends on the HP-UX version for which the distribution file is intended. (See Section [5.5.1 Distribution File Format](#) for distribution file naming conventions.)

[Table 5.4](#), below, identifies the files contained in the distribution file.

File	Description
Readme.unv	Summary of the installation procedure.
unv-4.1.0.0-hpux-11.11.depot	HP-UX .depot file-format package.
unvinst	Installation script.
upimerge.sh	Script that uses the Universal Installation Merge (UIM) module.

Table 5.4 Universal Products for HP-UX - Distribution File Contents

If your Universal Products for HP-UX distribution file does not contain these files, contact Stonebranch, Inc. [Customer Support](#) to obtain a correct distribution file.

HP-UX Package

The Universal Products for HP-UX is packaged as a depot file (extension `.depot`). The Universal Products depot contains product UCM.

[Table 5.5](#), below, identifies the sub-products contained in the HP-UX package.

Subproduct Name	Description
UNV.UNVcom	Universal Products common files, such as message catalogs and translation tables.
UNV.UNVct1	Universal Control Manager
UNV.UNVcts	Universal Control Server
UNV.UNVucm	Universal Command Manager
UNV.UNVucs	Universal Command Server
UNV.UNVqry	Universal Query
UNV.UNVsp1	Universal Spool
UNV.UNVubr	Universal Broker
UNV.UNVut1	Universal Utilities
UNV.UNVudm	Universal Data Mover Manager
UNV.UNVuds	Universal Data Mover Server
UNV.UNVusp	Universal Connector
UNV.UNVuem	Universal Event Monitor Manager
UNV.UNVues	Universal Event Monitor Server
UNV.UNVucr	Universal Certificate

Table 5.5 Universal Products for HP-UX - Package Subproducts

Installing Universal Products for HP-UX

Universal Products for HP-UX is installed with the `unvinst` script, which executes the `swinstall` command. The command to start the script must be executed with the superuser ID.

Note: Stonebranch, Inc. strongly recommends the use of the `unvinst` script for the HP-UX installation above any other method.

To install all of HP-UX packages, issue the following command:

```
sh ./unvinst
```

This is a silent install. The output from `unvinst` is written to file `install.log` in the current directory.

System initialization files `/sbin/init.d/ubrokerd` and `/sbin/rc3.d/S850ubrokerd` are created to start the `ubrokerd` daemon when the system boots to runlevel 3.

Note: With the Solaris, HP, and AIX installs, the previous installation directories are removed when the native installer has detected that additional directories or files have not been added to the original installation directories. If they have been modified, the directories will remain and can be reviewed and removed, as desired, by your Administrator.

Removing Universal Products for HP-UX

(Before removing Universal Products for HP-UX, stop the `ubrokerd` daemon.)

Universal Products for HP-UX is removed with the `swremove` command. The command must be executed with the superuser ID.

To remove all HP-UX subproducts, issue the following command:

```
swremove UNV
```

System initialization files `/sbin/init.d/ubrokerd` and `/sbin/rc3.d/S850ubrokerd` are removed.

Listing Universal Products for HP-UX Information

On HP-UX, information on an installed product and subproducts is listed with the `swlist` command. The command must be executed with the superuser ID.

To list information for Universal Products for HP-UX, issue the following command:

```
swlist -l subproduct UNV
```

5.6.3 Universal Products for Solaris Installation

Installation of Universal Products for Solaris is a three-step process:

1. Download product distribution file (see Section [5.5 Universal Products for UNIX – Product Distribution File](#)).
2. Extract the installation files from the distribution file.
3. Install the extracted files.

Extracting the Universal Products for Solaris Installation Files

Stonebranch, Inc. provides a separate product distribution file for the following Solaris systems:

- Intel-based systems
- SPARC-based systems

Universal Products for Solaris product distribution files are in a compressed `tar` format.

Intel-Based Systems

To uncompress and extract the installation files from the distribution file, issue the following command:

```
zcat unv-4.1.0.0-solaris-10-x64.tar.Z | tar xvf -
```

This command assumes that:

- Name of the distribution file is `unv-4.1.0.0-solaris-10-x64.tar.Z`.
- File is located in the current working directory.

[Table 5.9](#), below, identifies the files contained in the distribution file.

File	Description
<code>Readme.unv</code>	Summary of the installation procedure.
<code>unv-4.1.0.0-solaris-10-x64.pkg</code>	Solaris Intel-based file format package.
<code>unvinst</code>	Installation script.
<code>upimerge.sh</code>	Script that uses the Universal Installation Merge (UIM) module.

Table 5.6 Universal Products for Solaris (Intel-Based Systems) - Distribution File Contents

If your Universal Products for Solaris (Intel-based systems) distribution file does not contain these files, contact Stonebranch, Inc. [Customer Support](#) to obtain a correct distribution file.

SPARC-Based Systems

To uncompress and extract the installation files from the distribution file, issue the following command:

```
zcat unv-4.1.0.0-solaris-8-sparc.tar.Z | tar xvf -
```

This command assumes that:

- Name of the distribution file is `unv-4.1.0.0-solaris-8-sparc.tar.Z`.
- File is located in the current working directory.

The actual base name of the `.pkg` file depends on the Solaris version for which the distribution file is intended. (See Section 5.5.1 [Distribution File Format](#) for distribution file naming conventions.)

Table 5.7, below, identifies the files contained in the distribution file.

File	Description
<code>Readme.unv</code>	Summary of the installation procedure.
<code>unv-4.1.0.0-solaris-8-sparc.pkg</code>	Solaris SPARC-based file format package.
<code>unvinst</code>	Installation script.
<code>upimerge.sh</code>	Script that uses the Universal Installation Merge (UIM) module.

Table 5.7 Universal Products for Solaris (SPARC-Based Systems) - Product Distribution File

Note: If your Universal Products for Solaris (SPARC-based systems) distribution file does not contain these files, contact Stonebranch, Inc. [Customer Support](#) to obtain a correct distribution file.

Solaris Package

Universal Products for Solaris is packaged as a set of Solaris packages (extension .pkg).

Table 5.8, below, identifies the Universal Products for Solaris package names.

Package Name	Description
UNVcom	Universal Products common files, such as, message catalogs and translation tables.
UNVct1	Universal Control Manager
UNVcts	Universal Control Server
UNVucom	Universal Command Manager
UNVuCs	Universal Command Server
UNVqry	Universal Query
UNVsp1	Universal Spool
UNVubr	Universal Broker
UNVut1	Universal Utilities
UNVudm	Universal Data Mover Manager
UNVuds	Universal Data Mover Server
UNVuem	Universal Event Monitor Manager
UNVues	Universal Event Monitor Server
UNVusp *	Universal Connector
UNVucr	Universal Certificate
* UNVusp is not provided with the Universal Products for Solaris Intel-based file format package.	

Table 5.8 Universal Products for Solaris - Package Names

Installing Universal Products for Solaris

Universal Products for Solaris is installed with the `unvinst` script, which executes the `pkgadd` command. The command to start the script must be executed with the superuser ID.

Note: Stonebranch, Inc. strongly recommends the use of the `unvinst` script for the Solaris installation above any other method.

To install all Solaris packages, issue the following command:

```
sh ./unvinst
```

This is a silent install. The output from `unvinst` is written to file `install.log` in the current directory.

System initialization files `/etc/init.d/ubrokerd` and `/etc/rc3.d/S85ubrokerd` are created to start the `ubrokerd` daemon when the system boots to runlevel 3.

Note: With the Solaris, HP, and AIX installs, the previous installation directories are removed when the native installer has detected that additional directories or files have not been added to the original installation directories. If they have been modified, the directories will remain and can be reviewed and removed, as desired, by your Administrator.

Removing Universal Products for Solaris

(Before removing Universal Products for Solaris, stop the `ubrokerd` daemon.)

Universal Products for Solaris is removed with the `pkgrm` command. The command must be executed with the superuser ID.

To remove all Solaris packages, issue the following command:

```
pkgrm UNVucr UNVusp UNVut1 UNVues UNVuem UNVuds UNVudm UNVucs UNVucm  
UNVqry UNVct1 UNVcts UNVubr UNVsp1 UNVcom
```

System initialization files `/etc/init.d/ubrokerd` and `/etc/rc3.d/S85ubrokerd` are removed.

Note: `UNVusp` is not provided with the Universal Products for Solaris Intel-based file format package.

Listing Universal Products for Solaris Information

Information on an installed packages is listed with the `pkginfo` command. The command must be executed with the superuser ID.

To list information for Universal Products for Solaris, issue the following command:

```
pkginfo UNVucr UNVusp UNVut1 UNVues UNVuem UNVuds UNVudm UNVucs UNVucm  
UNVqry UNVct1 UNVcts UNVubr UNVsp1 UNVcom
```

Note: `UNVusp` is not provided with the Universal Products for Solaris Intel-based file format package.

5.6.4 Universal Products for Linux Installation

Installation of Universal Products for Linux is a three-step process:

1. Download product distribution file (see Section [5.5 Universal Products for UNIX – Product Distribution File](#)).
2. Extract the installation files from the distribution file.
3. Install the extracted files.

Extracting the Universal Products for Linux Installation Files

Stonebranch, Inc. provides separate product distribution files for the following Linux systems:

- Intel-based systems
- IBM S/390 and zSeries systems
- Itanium systems

Universal Products for Linux product distribution files are in a compressed `tar` format.

Intel-Based Systems

(The RPM package for Intel-based systems contains the `i386` qualifier.)

To uncompress and extract the installation files from the product distribution file, issue the following command:

```
zcat unv-4.1.0.0-linux-2.4-i386.tar.Z | tar xvf -
```

This command assumes that the name of the distribution file is `unv-4.1.0.0-linux-2.4-i386.tar.Z`.

[Table 5.9](#), below, identifies the files contained in the distribution file.

File	Description
<code>Readme.unv</code>	Summary of the installation procedure.
<code>unv-4.1.0.0-linux-2.4-i386.rpm</code>	Linux RPM file format package.
<code>unvinst</code>	Installation script.
<code>upimerge.sh</code>	Script that uses the Universal Installation Merge (UIM) module.

Table 5.9 Universal Products for Linux (Intel-Based Systems) - Product Distribution File

Note: If your Universal Products for Linux (Intel-based systems) distribution file does not contain these files, contact Stonebranch, Inc. [Customer Support](#) to obtain a correct distribution file.

IBM S/390 and zSeries Systems

(The RPM package for IBM S/390 and zArchitecture systems contains the **s390** qualifier.)

To uncompress and extract the installation files from the product distribution file, issue the following command:

```
zcat unv-4.1.0.0-linux-2.4-s390.tar.Z | tar xvf -
```

This command assumes that the name of the distribution file is **unv-4.1.0.0-linux-2.4-s390.tar.Z**.

[Table 5.10](#), below, identifies the files contained in the distribution file.

File	Description
Readme.unv	Summary the installation procedure.
unv-4.1.0.0-linux-2.4-s390.rpm	Linux RPM file format package.
unvinst	Installation script.
upimerge.sh	Script that uses the Universal Installation Merge (UIM) module.

Table 5.10 Universal Products for Linux (IBM S/390 and zSeries Systems) - Product Distribution File

Note: If your Universal Products for Linux (IBM S/390 and zSeries systems) distribution file does not contain these files, contact Stonebranch, Inc. [Customer Support](#) to obtain a correct distribution file.

Itanium Systems

(The RPM package for Intel-based systems contains the **ia64** qualifier.)

To uncompress and extract the installation files from the product distribution file, issue the following command:

```
zcat unv-4.1.0.0-linux-2.6-ia64.tar.Z | tar xvf -
```

This command assumes that the name of the distribution file is **unv-4.1.0.0-linux-2.6-ia64.tar.Z**.

[Table 5.11](#), below, identifies the files contained in the distribution file.

File	Description
Readme.unv	Summary of the installation procedure.
unv-4.1.0.0-linux-2.6-ia64.rpm	Linux RPM file format package.
unvinst	Installation script.
upimerge.sh	Script that uses the Universal Installation Merge (UIM) module.

Table 5.11 Universal Products for Linux (Itanium Systems) - Product Distribution File

Note: If your Universal Products for Linux (Itanium systems) distribution file does not contain the these files, contact Stonebranch, Inc. [Customer Support](#) to obtain a correct distribution file.

Installing Universal Products for Linux

Universal Products for Linux is installed with the `unvinst` script, which executes the `rpm` command. The command to start the script must be executed with the superuser ID.

Note: Stonebranch, Inc. strongly recommends the use of the `unvinst` script for the Linux installation above any other method.

To start the installation script, issue the following command:

```
sh ./unvinst
```

This is a silent install. The output from `unvinst` is written to file `install.log` in the current directory. The products are installed into directory `/opt`.

Note: For this release of Linux RPM, the previous non-RPM version will not be uninstalled. The Administrator can remove the remaining files/directories as needed.

Removing Universal Products for Linux

(Before removing Universal Products for SOA for Linux, stop the `ubrokerd` daemon.)

Universal Products for Linux is removed with the `rpm` command. The command must be executed with the superuser ID.

To remove all packages, issue the following command:

```
rpm -e unv
```

Listing Universal Products for Linux Information

Information on installed packages is listed with the `rpm` command. The command must be executed with the superuser ID.

To list information for the Universal Products for Linux, issue the following command:

```
rpm -qi unv
```

Linux PAM Customization

Linux installations utilize Pluggable Authentication Modules (PAM) for user authentication. Many of the Universal Products servers, such as Universal Command (UCMD), Universal Data Mover (UDM), and Universal Control (UCTL), must authenticate user accounts and passwords. Proper PAM configuration is essential for product operation.

There are many organizations and companies that package and distribute the Linux operating system. Most have consistent PAM implementations, but there are exceptions.

All supported Linux installations – except for SuSE 9 and below – require the sample PAM configuration file delivered with Universal Products to be copied to directory `/etc/pam.d`:

```
cp /opt/universal/ucmdsrv/samp/ucmd.pam /etc/pam.d/ucmd
```

All Universal Products utilize the same PAM configuration file. Its contents are:

- `auth required /lib/security/pam_pwdb.so shadow nullok`
- `auth required /lib/security/pam_nologin.so`
- `account required /lib/security/pam_pwdb.so`

Your Administrator must modify this sample PAM file to meet your local configuration.

Examples of Customized PAM Configuration Files

Universal Products for Redhat systems 5.0 and greater

- `auth include system-auth`
- `account required pam_nologin.so`
- `account include system-auth`

Universal Products for SUSE-based systems 10.0 and greater

- `auth required pam_unix2.so nullock`
- `auth required pam_nologin.so`
- `account include common-account`

Alternative Universal Products for SUSE 10.1

- `auth required pam_unix2.so nullock`
- `auth required pam_nologin.so`
- `account include common-account`

5.6.5 Universal Products for UNIX Installation (Generic)

Installation of Universal Products 4.1.0 for UNIX on operating systems other than AIX, HP-UX, Solaris, and Linux are performed with the generic Universal Products for UNIX installation script.

Installation of these Universal Products for UNIX is a three-step process:

1. Download product distribution file (see Section [5.5 Universal Products for UNIX – Product Distribution File](#)).
2. Extract the installation files from the distribution file.
3. Install the extracted files.

Extracting the Universal Products for UNIX Installation Files

The Universal Products for UNIX product distribution file is in a compressed **tar** format. It must be uncompressed and extracted in the user's home directory.

The products are installed in directory **/opt**.

To uncompress and extract the installation files from the distribution file, issue the following command:

```
zcat unv-4.1.0.0-mpras-3.02-i386.tar.Z | tar xvf -
```

(This command assumes that the name of the distribution file is **unv-4.1.0.0-mpras-3.02-i386.tar.Z**.)

[Table 5.12](#), below, identifies the files contained in this distribution file.

File	Description
Readme.unv	Summary of the installation procedure.
unv.tar	tar file format package.
unvinst	Installation script.
upimerge.sh	Script that uses the Universal Installation Merge (UIM) module.

Table 5.12 Universal Products for UNIX - Product Distribution File Contents

If your UNIX distribution file does not contain the above files, contact Stonebranch, Inc. [Customer Support](#) to obtain a correct distribution file.

Installing Universal Products for UNIX

Universal Products for UNIX (generic) is installed with the generic installation script, `unvinst`. The command to start the script must be executed with the superuser ID.

Note: Stonebranch, Inc. strongly recommends the use of the `unvinst` script for this UNIX installation above any other method.

To start the installation script, issue the following command:

```
sh ./unvinst
```

An installation log file is written to `./install.log`.

Note: With installation of Universal Products for UNIX using the generic installation script, the previous installation directories — default or user-specified — are not removed.

Installation Environment Variables

Environment variables can be defined to set default values for certain installation options.

Table 5.13, below, identifies the environment variables used and their install-script defaults.

Variable Name	Default Value	Notes
UNV_INSTLOG	<code>./install.log</code>	Installation log file.
UNV_CFGDIR	<code>/etc/universal</code>	Default configuration directory.
UNV_START_DAEMON	Y	Default response to start the Universal Broker daemon or not.
UNV_USRBIN	<code>/opt/universal/bin</code>	Default directory in which symbolic links are created to user programs.

Table 5.13 Universal Products for UNIX - Environment Variables

Removing Universal Products for UNIX

(Before removing Universal Products for UNIX, stop the `ubrokerd` daemon.)

Universal Products for UNIX (generic) is removed simply by deleting the relevant directories:

- `/opt/universal`
- `/var/opt/universal`
- `/etc/universal`

5.7 Universal Products for UNIX – Customization

This section provides the following information for the customization of Universal Products components:

- Configuration
- Product License
- System initialization (Universal Broker)

5.7.1 Universal Broker Customization

Configuration

Configuration options for Universal Broker are stored in configuration file, `ubroker.conf`, in directory `/etc/universal` by default.

See the Universal Broker 4.1.0 User Guide for details on configuring Universal Broker.

System Initialization

A Broker daemon start-up script is provided in file `/opt/universal/ubroker/ubrokd`. A single command line argument — either `start`, `stop`, `status`, or `restart` — instructs the script on the action to take.

See the Universal Broker 4.1.0 User Guide for details on the Broker daemon script.

To start the Broker daemon automatically each time that the system is started, add this script to the system initialization process. This is performed by the AIX, HP-UX, Solaris, and Linux (Red Hat- and SUSE-style systems) installation process, but not the generic UNIX installation script.

Unless the Broker daemon is started by the system initialization process, it will inherit the environment of the user that starts it. In some cases, there may be environmental variables that should not be inherited. For those variables, the start-up script `/opt/universal/ubroker-4.1.0/ubrokd` can be modified to unset the undesired environmental variables.

The format of the unset is as follows:

```
unset Variable1
unset MAIL
unset Variable2
```

This above modifications would cause `Variable1`, `MAIL`, and `Variable2` to become UNSET within the environment of the Broker Daemon.

5.7.2 Universal Command Manager Customization

Configuration

Configuration options for Universal Command Manager are stored in configuration file, `ucmd.conf`, in directory `/etc/universa1` by default.

See the Universal Command 4.1.0 User Guide for details on configuring Universal Command Manager.

The product executable files intended for command line use are located in the default directory `/opt/universa1/bin`. This directory must be added to the PATH environment variable for intended users of the executable files.

Product License

Universal Command Manager must be configured with a Universal Command product license before it is used. It will not run without a valid license.

The product license contains the license parameter keywords and their values, which must be entered into the Universal Command Manager configuration file, `ucmd.conf`. The values are specified in the same syntax as all other configuration options.

Product licenses are provided by your Stonebranch, Inc. account representative.

5.7.3 Universal Command Server Customization

Configuration

Configuration options for Universal Command Server are stored in configuration file, `ucmds.conf`, in directory `/etc/universa1` by default.

See the Universal Command 4.1.0 User Guide for details on configuring Universal Command Server.

Universal Command Server runs as a component managed by Universal Broker. Universal Command Server provides a component definition file, `ucmd`, that Universal Broker uses to start the Server and establish its runtime environment. `ucmd` is located in directory `/etc/universa1/comp` by default

The product executable files intended for command line use are located in the default directory `/opt/universa1/bin`. This directory must be added to the PATH environment variable for intended users of the executable files.

Universal Command Server uses the Universal Access Control List (UACL) configuration file as a level of product security. How Universal Command Server utilizes the UACL file is described in the Universal Command 4.1.0 User Guide.

5.7.4 Universal Connector Customization

Configuration

Configuration options for Universal Connector are stored in configuration file, **usap.conf**, in directory **/etc/universal** by default.

See the Universal Connector 4.1.0 User Guide for details on configuring USAP.

The product executable files intended for command line use are located in the default directory **/opt/universal/bin**. This directory must be added to the **PATH** environment variable for intended users of the executable files.

SAP RFC Configuration

Universal Connector utilizes SAP's RFC interface. The RFC interface uses configuration file **saprfc.ini** to store information required for connecting to SAP systems. The **saprfc.ini** file is an SAP resource that can be shared by multiple external tools using the SAP RFC interface. The **saprfc.ini** file groups connection information into **destinations**. Each destination contains the connection information required to establish a connection to a particular SAP system.

The connections defined in the **saprfc.ini** file must be configured to meet your local SAP environment. A sample **saprfc.ini** file is installed with Universal Connector to directory **/opt/universal/usap/samp**. The file is provided by SAP and contains complete documentation on its configuration.

In order for Universal Connector to find the configured **saprfc.ini** file, it must be placed in the Universal Connector executable directory, or environment variable **RFC_INI** must be set to its full path name.

Product License

Universal Connector must be configured with a Universal Connector product license before it is used. It will not run without a valid license.

The product license contains the license parameter keywords and their values, which must be entered into the Universal Connector configuration file, **usap.conf**. The values are specified in the same syntax as all other configuration options.

Product licenses are provided by your Stonebranch, Inc. account representative.

5.7.5 Universal Control Manager Customization

Configuration

Configuration options for Universal Control Manager are stored in configuration file, `uct1.conf`, in directory `/etc/universa1` by default.

See the Universal Products Utilities 4.1.0 User Guide for details on configuring Universal Control Manager.

The product executable files intended for command line use are located in the default directory `/opt/universa1/bin`. This directory must be added to the PATH environment variable for intended users of the executable files.

5.7.6 Universal Control Server Customization

Configuration

Configuration options for Universal Control Server are stored in configuration file, `uct1s.conf`, in directory `/etc/universa1` by default.

See the Universal Products Utilities 4.1.0 User Guide for details on configuring Universal Control Server.

Universal Control Server runs as a component managed by Universal Broker. Universal Control Server provides a component definition file, `uct1`, that Universal Broker uses to start the Server and establish its runtime environment. `uct1` is located in directory `/etc/universa1/comp` by default

The product executable files intended for command line use are located in the default directory `/opt/universa1/bin`. This directory must be added to the PATH environment variable for intended users of the executable files.

Universal Control Server uses the Universal Access Control List (UACL) configuration file as a level of product security. How Universal Control Server utilizes the UACL file is described in the Universal Products Utilities 4.1.0 User Guide.

5.7.7 Universal Data Mover Manager Customization

Configuration

Configuration options for UDM Manager are stored in configuration file, **udm.conf**, in directory **/etc/universa1** by default.

See the Universal Data Mover 4.1.0 User Guide for details on configuring UDM Manager.

The product executable files intended for command line use are located in the default directory **/opt/universa1/bin**. This directory must be added to the PATH environment variable for intended users of the executable files.

Product License

UDM Manager must be configured with a Universal Data Mover product license before it is used. It will not run without a valid license.

The product license contains the license parameter keywords and their values, which must be entered into the UDM Manager configuration file, **udm.conf**. The values are specified in the same syntax as all other configuration options.

Product licenses are provided by your Stonebranch, Inc. account representative.

5.7.8 Universal Data Mover Server Customization

Configuration

Configuration options for UDM Server are stored in configuration file, **udms.conf**, in directory **/etc/universa1** by default.

UDM Server runs as a component managed by Universal Broker. UDM Server provides a component definition file, **udm**, that Universal Broker uses to start the Server and establish its runtime environment. **udm** is located in directory **/etc/universa1/comp** by default

The product executable files intended for command line use are located in the default directory **/opt/universa1/bin**. This directory must be added to the PATH environment variable for intended users of the executable files.

UDM Server uses the Universal Access Control List (UACL) configuration file as a level of product security.

See the Universal Data Mover 4.1.0 User Guide for details on configuring UDM Server and information on how UDM Server utilizes the UACL file.

5.7.9 Universal Event Monitor Manager Customization

Configuration

Configuration options for UEM Manager are stored in configuration file `uem.conf`, in directory `/etc/universal` by default. See the Universal Event Monitor 4.1.0 User Guide for details on configuring UEM Manager.

The product executable files intended for command line use are located in the default directory `/opt/universal/bin`. This directory must be added to the PATH environment variable for intended users of the executable files.

5.7.10 Universal Event Monitor Server Customization

Configuration

Configuration options for UEM Server are stored in configuration file, `uems.conf`, in directory `/etc/universal` by default.

UEM Server runs as a component managed by Universal Broker. UEM Server provides two component definition files, located in the default directory `/etc/universal/comp`, that Universal Broker uses to start the Server and establish its runtime environment:

- `uems` is used to start an event-driven UEM Server.
- `uemd` is used to start a demand-driven UEM Server.

The product executable files intended for command line use are located in the default directory `/opt/universal/bin`:

- `uem`
- `uemload`

This directory must be added to the PATH environment variable for intended users of the executable files.

UEM Server uses the Universal Access Control List (UACL) configuration file as a level of product security. See the Universal Event Monitor 4.1.0 User Guide for details on configuring UEM Server and information on how UEM Server utilizes the UACL file.

Product License

A UEM server (whether event-driven or demand-driven) requires a valid Universal Event Monitor product license to be configured in the UEM Server configuration file, `uems.conf`, before it can be used. Neither UEM server can be run without a valid license.

The product license contains the license parameter keywords and their values, which are specified in the same syntax as all other configuration options.

Product licenses are provided by your Stonebranch, Inc. account representative.

5.7.11 Universal Query Customization

Configuration

Configuration options for Universal Query are stored in configuration file, `uquery.conf`, in directory `/etc/universal` by default.

See the Universal Products Utilities 4.1.0 User Guide for details on configuring Universal Query.

The product executable files intended for command line use are located in the default directory `/opt/universal/bin`. This directory must be added to the PATH environment variable for intended users of the executable files.

5.8 Universal Products for UNIX – File Inventory Lists

The Universal Products installation for UNIX includes the files required for:

- Universal Broker
- Universal Command Manager and Server
- Universal Connector
- Universal Control Manager and Server
- Universal Data Mover Manager and Server
- Universal Event Monitor Manager and Server
- Universal Certificate
- Universal Query
- Universal Spool Utilities

This section identifies the files installed with each of these Universal Products components/utilities.

The file paths listed presume that the installation directory (`/opt`) and the variable file directory (`/var/opt`) are the defaults. These directories can be changed on most UNIX installations.

Universal Broker

File	Description
<code>/opt/universal/ubroker/ubrinst.src</code>	Broker installation source file (included in Generic UNIX packaging only).
<code>/opt/universal/ubroker/ubrokerd</code>	Broker daemon start script.
<code>/opt/universal/ubroker/bin/ubroker</code>	Console broker program.
<code>/opt/universal/ubroker/bin/ubrokerd</code>	Daemon broker program.
<code>/etc/universal/comp</code>	Component definition directory.
<code>/opt/universal/ubroker/samp</code>	Broker sample directory.
<code>/opt/universal/ubroker/tmp1</code>	XML template files used by I-Management Console for remotely configuring Universal Products
<code>/opt/universal/nls</code>	Code page files used for text translation between different operating systems and product message catalogs.
<code>/var/opt/universal/log</code>	Broker message log directory.
<code>/var/opt/universal/trace</code>	Broker trace file directory.
<code>/var/opt/universal/spool</code>	Broker component database files.
<code>/etc/universal/uac1.conf</code>	Universal Access Control List configuration file.
<code>/etc/universal/ubroker.conf</code>	Broker configuration file.

Table 5.14 Universal Broker for UNIX - File Inventory List

Universal Command Manager

File	Description
<code>/opt/universal/ucmdmgr/ucmcinst.src</code>	Manager installation source file (included in Generic UNIX packaging only).
<code>/opt/universal/ucmdmgr/bin/ucmd</code>	Universal Command Manager program.
<code>/opt/universal/ucmdmgr/bin/uencrypt</code>	Universal Encrypt program file.
<code>/opt/universal/ucmdmgr/samp</code>	Manager sample file directory.
<code>/opt/universal/bin/ucmd</code>	Symbolic link to the Manager program.
<code>/opt/universal/bin/uencrypt</code>	Symbolic link to the Universal Encrypt program.
<code>/opt/universal/nls</code>	Code page files used for text translation between different operating systems and product message catalogs.
<code>/etc/universal/ucmd.conf</code>	Manager configuration file.

Table 5.15 Universal Command Manager for UNIX - File Inventory List

Universal Command Server

File	Description
<code>/opt/universal/ucmdsrv/ucmsinst.src</code>	Server installation source file (included in Generic UNIX packaging only).
<code>/opt/universal/ucmdsrv/bin/ucmsrv</code>	Server component program.
<code>/opt/universal/ucmdsrv/bin/ucopy</code>	Utility used for binary file copies. Similar to the UNIX <code>cat</code> command.
<code>/opt/universal/ucmdsrv/bin/umet</code>	Universal Message Translator program.
<code>/opt/universal/ucmdsrv/samp</code>	Server sample directory.
<code>/etc/universal/comp</code>	Universal Command Server component definition file.
<code>/opt/universal/bin/ucopy</code>	Symbolic link to the Universal Copy program.
<code>/opt/universal/bin/umet</code>	Symbolic link to the Universal Message Translator program.
<code>/opt/universal/nls</code>	Code page files used for text translation between different operating systems and product message catalogs.
<code>/var/opt/universal/trace</code>	Server trace file directory.
<code>/var/opt/universal/spool</code>	Server component database files and spool file directory.
<code>/etc/universal/ucmds.conf</code>	Server configuration file.

Table 5.16 Universal Command Server for UNIX - File Inventory List

Universal Connector

File	Description
<code>/opt/universal/usap/usapinst.src</code>	Server installation source file (include in Generic UNIX packaging only).
<code>/opt/universal/usap/bin/usap</code>	USAP program
<code>/opt/universal/usap/samp</code>	Server sample file directory.
<code>/opt/universal/nls</code>	Code page files used for text translation between different operating systems and product message catalogs.
<code>/etc/universal/usap.conf</code>	Server configuration file.

Table 5.17 Universal Connector for UNIX - File Inventory List

Universal Control Manager

File	Description
<code>/opt/universal/uctlmgr/uctcinst.src</code>	Manager installation source file (include in Generic UNIX packaging only).
<code>/opt/universal/uctlmgr/bin/uctl</code>	Manager program.
<code>/opt/universal/uctlmgr/samp</code>	Manager sample file directory.
<code>/opt/universal/bin/uctl</code>	Symbolic link to the Manager program.
<code>/opt/universal/nls</code>	Code page files used for text translation between different operating systems and product message catalogs.
<code>/etc/universal/uctl.conf</code>	Manager configuration file.

Table 5.18 Universal Control Manager for UNIX - File Inventory List

Universal Control Server

File	Description
<code>/opt/universal/uctlsrv/uctsinst.src</code>	Server installation source file (include in Generic UNIX packaging only).
<code>/opt/universal/uctlsrv/bin/uctsrv</code>	Server component program.
<code>/opt/universal/uctlsrv/samp</code>	Server sample file directory.
<code>/etc/universal/comp</code>	Universal Control Server component definition file.
<code>/opt/universal/nls</code>	Code page files used for text translation between different operating systems and product message catalogs.
<code>/var/opt/universal/trace</code>	Server trace file directory.
<code>/etc/universal/uctls.conf</code>	Server configuration file.

Table 5.19 Universal Control Server for UNIX - File Inventory List

Universal Data Mover Manager

File	Description
<code>/opt/universal/udmmgr/udminst.src</code>	Manager installation source file (included in Generic UNIX packaging only).
<code>/opt/universal/udmmg/bin/udm</code>	Manager program.
<code>/opt/universal/udmmgr/samp</code>	Manager sample file directory.
<code>/opt/universal/bin/udm</code>	Symbolic link to the Manager program.
<code>/opt/universal/nls</code>	Code page files used for text translation between different operating systems and product message catalogs.
<code>/etc/universal/udm.conf</code>	Manager configuration file.

Table 5.20 Universal Data Mover Manager for UNIX - File Inventory List

Universal Data Mover Server

File	Description
<code>/opt/universal/udmsrv/udmsinst.src</code>	Server installation source file (include in Generic UNIX packaging only).
<code>/opt/universal/udmsrv/bin/udmsrv</code>	Server program.
<code>/opt/universal/udmsrv/samp</code>	Server sample file directory.
<code>/opt/universal/nls</code>	Code page files used for text translation between different operating systems and product message catalogs.
<code>/etc/universal/udms.conf</code>	Server configuration file.
<code>/etc/universal/comp</code>	Universal Data Mover Server component definition file.

Table 5.21 Universal Data Mover Server for UNIX - File Inventory List

Universal Event Monitor Manager

File	Description
<code>/opt/universal/uemmgr/ueminst.src</code>	Manager installation source file (included in Generic UNIX packaging only).
<code>/opt/universal/uemmgr/bin/uem</code>	Manager program.
<code>/opt/universal/uemmgr/samp</code>	Manager sample file directory.
<code>/opt/universal/bin/uem</code>	Symbolic link to the Manager program.
<code>/opt/universal/nls</code>	Code page files used for text translation between different operating systems and product message catalogs.
<code>/etc/universal/uem.conf</code>	Manager configuration file.

Table 5.22 Universal Event Monitor Manager for UNIX - File Inventory List

Universal Event Monitor Server

File	Description
/opt/universal/uemsv/ueminst.src	Server installation source file (included in Generic UNIX packaging only).
/opt/universal/uemsv/bin/uemsv	Server program.
/opt/universal/uemsv/bin/uemload	UEM database load utility.
/opt/universal/uemsv/samp	Server sample file directory.
/opt/universal/bin/uemload	Symbolic link to the UEM database load utility.
/opt/universal/nls	Code page files used for text translation between different operating systems and product message catalogs.
/etc/universal/uems.conf	Server configuration file.
/etc/universal/comp/uems	Event-driven Universal Event Monitor Server component definition file.
/etc/universal/comp/uemd	Demand-driven Universal Event Monitor Server component definition file.

Table 5.23 Universal Event Monitor Server for UNIX - File Inventory List

Universal Certificate

File	Description
/opt/universal/ucert/bin/ucert	UCERT program

Table 5.24 Universal Certificate for UNIX - File Inventory List

Universal Query

File	Description
/opt/universal/uquery/uqryinst.src	Query installation source file (included in Generic UNIX packaging only).
/opt/universal/uquery/bin/uquery	Universal Query program.
/opt/universal/uquery/samp	Query sample file directory.
/opt/universal/bin/uquery	Symbolic link to the Query program.
/opt/universal/nls	Code page files used for text translation between different operating systems and product message catalogs.
/etc/universal/uquery.conf	Query configuration file.

Table 5.25 Universal Query for UNIX - File Inventory List

Universal Spool Utilities

File	Description
/opt/universal/uspool/bin/uslist	Used to list the contents of Universal Spool files.
/opt/universal/uspool/bin/uslrm	Used to remove records from Universal Spool files.
/opt/universal/uspool/bin/udb_archive /opt/universal/uspool/bin/udb_checkpoint /opt/universal/uspool/bin/udb_deadlock /opt/universal/uspool/bin/udb_dump /opt/universal/uspool/bin/udb_load /opt/universal/uspool/bin/udb_printlog /opt/universal/uspool/bin/udb_recover /opt/universal/uspool/bin/udb_stat /opt/universal/uspool/bin/udb_upgrade /opt/universal/uspool/bin/udb_verify	Miscellaneous spool file utilities. Should be used only for debugging purposes, and only at the request of Stonebranch, Inc. Customer Support .

Table 5.26 Universal Spool Utilities for UNIX - File Inventory List

5.9 Universal Products for SOA for UNIX Package

This section provides information regarding installation of Universal Products for SOA.

The Universal Products for SOA 4.1.0 for UNIX package includes the following product components:

- Universal Application Container Server
- Universal Application Container
- Universal Application Interface

Installation is a straightforward process using the platform-specific package. The installed environment is self-contained and follows current Universal Products installation standards.

The following topics are discussed:

- [Universal Products for SOA for UNIX – Installation Requirements](#)
- [Universal Products for SOA for UNIX – Licensing](#)
- [Universal Products for SOA for UNIX – Deployment Options](#)
- [Universal Products for SOA for UNIX – Pre-Installation / Upgrade Backups](#)
- [Universal Products for SOA for UNIX – Installation](#)
- [Universal Products for SOA for UNIX – File Inventory Lists](#)

5.10 Universal Products for SOA for UNIX – Installation Requirements

To install Universal Products for SOA for UNIX, you must have one of the following versions of UNIX:

- AIX 5.2 and greater
- Linux 2.4 kernel and greater
 - RedHat Package Manager (RPM)
 - Intel (x86) Compatible Systems

In addition, you must have:

- 512MB RAM minimum, 1 GB or more preferred.
- 150 MB free disk space.
- TCP/IP socket implementation.
- Superuser (root) access.
- Bourne shell or compatible.
- Universal Products 3.2.0.0 or later.

5.11 Universal Products for SOA for UNIX – Licensing

Universal Products for SOA allows operation by connector and endpoint count, based on the license information it receives from the UAC Server component. (See your Stonebranch, Inc. account representative for details.)

5.12 Universal Products for SOA for UNIX – Deployment Options

Deployment of Universal Products for SOA has been designed to be flexible in order to fit the needs of your Enterprise IT.

There are two main deployment options:

1. [Single-Server Deployment \(SSD\)](#)
2. [Distributed Server Deployment \(DSD\)](#)

SSD is the default deployment.

5.12.1 Single-Server Deployment (SSD)

Single-Server Deployment (SSD) is where all components, with the exception of the Universal Products managers, are deployed to the same physical server. This includes Universal Products for SOA, Universal Broker and associated components, and your target workload(s).

[Figure 5.1](#), below, illustrates SSD.

The following list identifies the flow of events that occur with SSD deployment:

1. The calling application invokes the Universal Command (UCMD) Manager to execute a remote workload. For this example the remote workload, or target workload, is an internet or message based workload.

Note: Although this figure implies the use of a Java-based workload, you can execute any remote workload, regardless of what language the workload is implemented in, as long as it has an interface that supports HTTP, SOAP, or JMS.

2. As usual, the request is forwarded to the Universal Broker specified in the invocation of the Universal Command Manager, which then spawns the UCMD Server and passes the workload execution request to Universal Products for SOA.
3. Universal Products for SOA will execute the workload deployed on the same physical server and return any messages or data back to the Universal Command Manager.

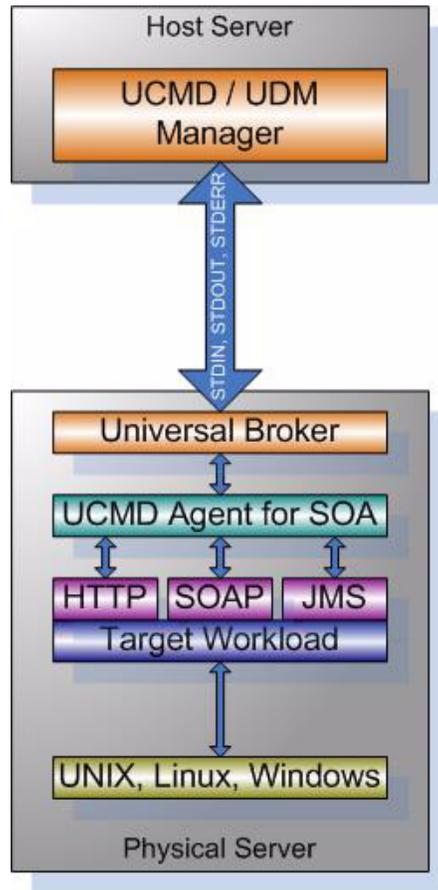


Figure 5.1 Single-Server Deployment - Logical View

5.12.2 Distributed Server Deployment (DSD)

Distributed Server Deployment (DSD) is where:

- Universal Broker and Universal Products for SOA are located together on one physical server.
- Target workload is located on a different physical server.

Note: The target workload location is user-defined. The SERVICE_URL option specifies how Universal Products for SOA knows where to look for the target workload.

The flow is nearly the same as with the Single Server Deployment except that the location of the target workload is not `localhost`.

Figure 5.2, below, illustrates the Distributed Server Deployment (DSD).

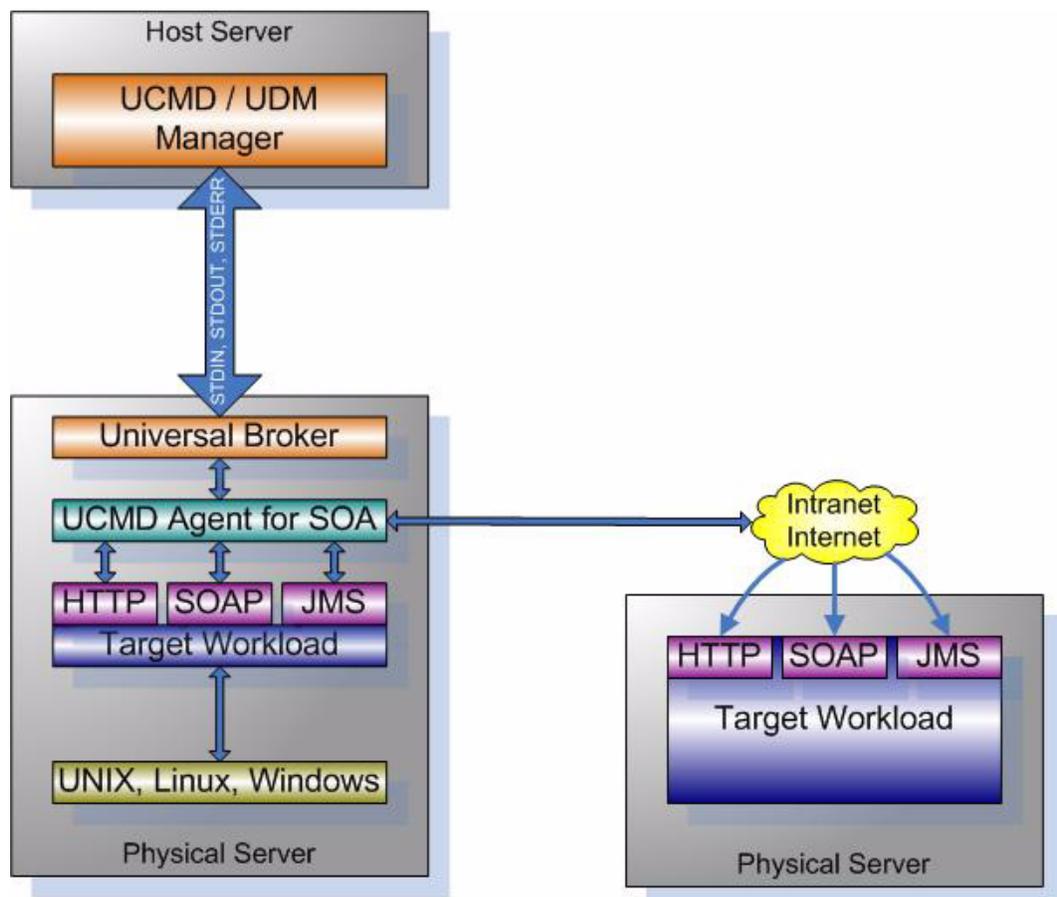


Figure 5.2 Distributed-Server Deployment - Logical View

5.13 Universal Products for SOA for UNIX – Pre-Installation / Upgrade Backups

The installation process overwrites the current files, effectively removing your modifications. Backing up these files will optimize the time it takes you to get up and running after installing or upgrading.

The following list identifies the files – and their locations – that should be backed up or copied before you install a new release or upgrade a current release.

UAC.xml File

`/etc/universa1/uac`

Log4jConfiguration.xml File

`/opt/universa1/uac (UAC)`

`/opt/universa1/uai (UAI)`

JMS Provider Client Jar Files

`/opt/universa1/uac/container/webapps/axis2/WEB-INF/lib`

Note: The names of the jar files differ depending on which JMS Provider you are using.

JMS Provider Client Properties Files

`/opt/universa1/uai/xml`

Note: These are suggested locations; you can place these files anywhere on the files system. If you have located these files under the `uai` directory, you should back them up.

Payload Files

Normally, payload files should be located on the host system where Universal Command Manager is installed.

If you have chosen to store them elsewhere, the suggested location is:

`/opt/universa1/uai/xml`

Note: You can place these files anywhere on the files system. If you have located these files under the `uai` directory, you should back them up.

5.14 Universal Products for SOA for UNIX – Installation

This section provides the following UNIX operating systems installation procedures:

- [Universal Products for SOA – AIX Installation](#)
- [Universal Products for SOA – Linux Installation](#)

Installation of Universal Products for SOA is performed via platform-specific installers.

Note: Before installing Universal Products for SOA 4.1.0, you must install Universal Products 3.2.0.

5.14.1 Universal Products for SOA – AIX Installation

Product Distribution File

The Universal Products for SOA for AIX product distribution file is in a compressed `tar` format.

The name of the Universal Products for SOA for AIX distribution file has the following format:

```
ups-4.1.0.0-aix-5.2.tar.Z
```

Unpacking and Installation Procedures

To unpack and install Universal Products for SOA, perform the following steps:

1. Create (or select an existing) directory in which to save the distribution file.
2. Save the distribution file into that directory.
3. Uncompress and extract the installation files in the current working directory. The command to extract the files is:

```
zcat ups-4.1.0.0-aix-5.2.tar.Z | tar xvf -
```

If your operating system does not support the `zcat` command, use the following command:

```
gunzip ups-4.1.0.0-aix-5.2.tar.Z
```

The output of the `gunzip` command provides you with the `tar` file.

```
tar -xvf ups-4.1.0.0-aix-5.2.tar.Z
```

4. After the extraction is complete, run the `upsinst` script:

```
sh ./upsinst
```

An installation log is written to file `install.log` in the current directory. The installation script, `upsinst`, restarts the Universal Broker daemon, `ubrokerd`, automatically at the end of the install.

5. Use the Universal Query utility (`/opt/universal/bin/uquery`) to validate that Universal Products for SOA 4.1.0 is running:

```
uquery -i localhost (or the name of your server)
```

6. From the license file that was sent to you by Stonebranch, Inc., add the license information to the `/etc/universal/uacs.conf` file.

7. Recycle Universal Broker by issuing the following commands:

```
a. /opt/universal/ubroker/ubrokerd stop
```

```
b. /opt/universal/ubroker/ubrokerd start
```

8. Use Universal Query to validate, again, that Universal Products for SOA 4.1.0 is running after updating the `uacs` configuration file:

```
uquery -i localhost (or the name of your server)
```

Removing Universal Products for SOA for AIX

Note: Before removing Universal Products for SOA for AIX, stop the **ubrokerd** daemon. Also, it is strongly recommended that you back up existing data before removing Universal Products for SOA for AIX.

Universal Products for SOA for AIX is removed with the **installp** command. The command must be executed with the superuser ID.

To remove AIX, issue the following command:

```
installp -u UPSuac
```

All entries in the **/etc/inittab** file that reference **ubroker** are removed.

Listing Universal Products for SOA for AIX Information

On AIX, information on an installed product is listed with the **ls1pp** command. The command must be executed with the superuser ID.

Issue the following command to list information for Universal Products for SOA for AIX:

```
ls1pp -La 'UPS*'
```

5.14.2 Universal Products for SOA – Linux Installation

Product Distribution File

Universal Products for SOA 4.1.0 is packaged as an RPM file (extension `.rpm`). It is installed using the Linux `rpm` command.

The name of the Universal Products for SOA for Linux distribution file has the following format:

```
ups-4.1.0.0-linux-2.4-i386.tar.Z
```

Unpacking and Installation Procedures

To unpack and install Universal Products for SOA, perform the following steps:

1. Create (or select an existing) directory in which to save the distribution file.
2. Save the distribution file into that directory.
3. Uncompress and extract the installation files in the current working directory. The command to extract the files is:

```
zcat ups-4.1.0.0-linux-2.4-i386.tar.Z | tar xvf -
```

If your operating system does not support the `zcat` command, use the following command:

```
gunzip ups-4.1.0.0-linux-2.4-i386.tar.Z
```

The output of the `gunzip` command provides you with the `tar` file.

```
tar -xvf ups-4.1.0.0-linux-2.4-i386.tar
```

4. After the extraction is complete, run the `upsinst` script:

```
sh ./upsinst
```

An installation log is written to file `install.log` in the current directory. The installation script, `upsinst`, restarts the Universal Broker daemon, `ubrokerd`, automatically at the end of the install.

5. Use the Universal Query utility (`/opt/universal/bin/uquery`) to validate that Universal Products for SOA 4.1.0 is running:

```
uquery -i localhost (or the name of your server)
```

6. From the license file that was sent to you by Stonebranch, Inc., add the license information to the `/etc/universal/uacs.conf` file.

7. Recycle Universal Broker by issuing the following commands:

```
a. /opt/universal/ubroker/ubrokerd stop
```

```
b. /opt/universal/ubroker/ubrokerd start
```

8. Use Universal Query to validate, again, that Universal Products for SOA 4.1.0 is running after updating the `uacs` configuration file:

```
uquery -i localhost (or the name of your server)
```

Removing Universal Products for SOA for Linux

Note: Before removing Universal Products for SOA for Linux, stop the **ubrokerd** daemon. Also, it is strongly recommended that you back up existing data before removing Universal Products for SOA for Linux.

Universal Products for SOA for Linux is removed with the **rpm** command. The command must be executed with the superuser ID.

To remove all packages, issue the following command:

```
rpm -e ups
```

Listing Universal Products for SOA for Linux Information

Information on installed packages is listed with the **rpm** command. The command must be executed with the superuser ID.

To list information for the Universal Products for SOA for Linux, issue the following command:

```
rpm -qi ups
```

5.15 Universal Products for SOA for UNIX – File Inventory Lists

This section identifies the Universal Products for SOA file system hierarchy and its contents.

The parent directories under which Universal Products for SOA operates are based on the existing Universal Products deployment. There should be no product components in these directories, just the product directories.

The product directories contain the product components either directly or in sub directories and are divided into two categories: runtime and logging.

- Runtime directories are for runtime operation of the product; they are not written to.
- Logging directories are where database components, audit and logging files reside, all components that are written to.

5.15.1 Parent Directories

For UNIX, the parent directories are:

- `/opt/universa1`
- `/var/opt/universa1`

The following sections identify the directories and files located under each of these parent directories.

5.15.2 Product Directories and Files

/opt/universal Parent Directory

Table 5.27, below, identifies the Universal Products for SOA for UNIX product directories and files located under the `/opt/universal` parent directory.

Directory / File	Description
uac	Directory containing artifacts for the UAC component. It includes the following sub-directories and files.
bin	Subdirectory containing the <code>uacsvr</code> executable. It requires no user interaction.
lib	Subdirectory containing the <code>uacsvr.jar</code> file needed for communication between the <code>uacsvr</code> and <code>uac</code> components. It requires no user interaction.
container	Subdirectory containing the libraries and other deployable objects needed for UAC operation. The only reason to explore this directory is if you are running the JMS Connector and need to deploy your JMS provider client jar files.
log4jconfiguration.xml	Logger configuration file that you may need to modify if you want to change the logging level of the UAC component.
shutdownUAC.sh	Shutdown script for UAC. You usually will not need to use this script, as it is the responsibility of Universal Broker to shut down UAC.
startUAC.sh	Startup script for UAC. You usually will not need to use this script, as it is the responsibility of Universal Broker to start UAC.
uacValidateInbound.sh	Script that validates the contents of the <code>UAC.xml</code> file using the schema definition contained in <code>UAC.xsd</code> .
UAC.xsd	File used by UAC to validate content of the <code>UAC.xml</code> inbound configuration file. It requires no user interaction.
uai	Directory containing the artifacts for the UAI component. It includes the following sub-directories and files.
lib	Subdirectory containing the libraries needed for UAI operation. It usually requires no user interaction.
log4jconfiguration.xml	Logger configuration file that you may need to modify if you want to change the logging level of the UAI component.
UAI.sh	Shell script used to invoke the UAI component. It usually requires no user interaction.
jre	Subdirectory containing the Java Runtime Environment for the UAC, UAI, and UAC Server components. It usually requires no user interaction.
nls/uacmceng.umc	Universal Application Container Server English message catalog.
Readme.ups	Universal Products for SOA for UNIX installation README file.
ubroker/tmp1/uaccfg	Universal Application Container Server configuration template file.
ubroker/tmp1/uaccmp	Universal Application Container Server component definition template file.

Table 5.27 Product Directories and Files (/opt/universal) – Universal Products for SOA for UNIX

/var/opt/universal Parent Directory

Table 5.28, below, identifies the Universal Products for SOA for UNIX product directories and files located under the `/var/opt/universal1` parent directory.

Directory / File	Description
<code>log/uac</code>	Directory containing logging and work artifacts for the UAC component. It includes the following sub-directories and files.
<code>temp</code>	Subdirectory used by UAC for temporary file persistence. It requires no user interaction.
<code>work</code>	Subdirectory used by the web services framework for temporary operations. It requires no user interaction.
<code>catalina.out</code>	Container engine log file.
<code>container.log</code>	Web services framework log file.
<code>derby.log</code>	Database log file.
<code>uac.log</code>	UAC log file.
<code>log/uai</code>	Directory containing the logging and work artifacts for the UAI component. It is empty by default. If <code>trace</code> or <code>info</code> is enabled, the output would be written to this directory.
<code>uac_db</code>	Directory containing the database components, log, property, and lock files. It requires no user interaction.

Table 5.28 Product Directories and Files (/var/opt/universal) – Universal Products for SOA for UNIX

/etc/universal Parent Directory

Table 5.29, below, identifies the Universal Products for SOA for UNIX product directories and files located under the `/etc/universal1` parent directory.

Directory / File	Description
<code>uacs.conf</code>	Universal Application Container Server configuration file.
<code>UAC.xml</code>	Inbound operation configuration file that you may need to modify to support your inbound workload operations.
<code>comp/uac</code>	Universal Application Container Server component definition file.

Table 5.29 Product Directories and Files (/etc/universal) – Universal Products for SOA for UNIX

Chapter 7

HP NonStop Installation

7.1 Overview

This chapter provides information on the installation of Stonebranch, Inc. Universal Command on the HP NonStop operating system.

**Currently, HP NonStop runs Universal Command 2.1.1.
This chapter provides information
for the installation of that version.**

Note: All Universal Products 2.1.1 for HP NonStop (see Section [7.2 Universal Command for HP NonStop Components](#)) are included as part of Universal Command.

7.2 Universal Command for HP NonStop Components

The Universal Command for HP NonStop installation includes the following Universal Products components:

- Universal Broker 2.1.1
- Universal Command Manager and Server 2.1.1
- Universal Control Manager and Server 2.1.1
- Universal Query 2.1.1
- Universal Encrypt 2.1.1
- Universal Message Translator 2.1.1

7.2.1 Product Compatibility

Table 7.1, below, identifies the compatibility of Universal Command 2.1.1 for HP NonStop with previous product versions.

Product	Compatibility
Universal Command 2.1.1	Universal Command 2.2.0, 2.1.0, and 1.2.0

Table 7.1 Universal Command for HP NonStop - Product Compatibility

7.3 Universal Command for HP NonStop – Installation Requirements

Universal Command installation requirements for HP NonStop are:

- HP NonStop system:
 - HP NSK S-series (running the G06.13 or greater version of the OS)
 - HP Integrity (running the H06 OS)
- Open System Services (OSS) configured and running. The TACL command TESTOSS can be used to check the OSS configuration on the HP NonStop.
- TACL shell.
- Open System Services Local Services (OSSLS).
- TCP/IP Socket implementation.
- One available TCP/IP port.
- Approximately 20 megabytes of disk space for the installation. More disk space is required for variable files, such as, log files and trace files.
- super.super access.

Note: In order to install these products, you must be able to write to the directory from which the installation is launched.

7.3.1 \$SYSTEM Volume

All product files that are written to during product execution are stored in the **\$SYSTEM** volume. The following sections describe the space and security requirements for all of the subdirectories.

Log Directory

Universal Broker can be configured to write its messages to a log file. The current log file and previous log file generations are stored in the **\$SYSTEM.UNVLOG** subvolume.

Space

A log file size grows to about 150,000 bytes and is then rolled over to a generation file. Five generations of log files are saved. The oldest generation log file is deleted. The amount of space required for the five generations and the current log file is about 900,000 bytes.

Security

Universal Broker requires read/write access to the log subvolume and read/write access to all files in the log subvolume. No other Universal Products use the log subvolume at this time. No general user access is required.

Trace Directory

Universal Broker and its server components (for example, Universal Command Server) create product trace files when configured to do so. A trace file is used by Stonebranch, Inc. [Customer Support](#) to resolve product problems. The trace files are stored in the `$SYSTEM.UNVTRACE` subvolume.

Space

Trace files can grow to significant size depending on how long the trace is active and how much work the program is doing during the tracing period. A trace file size of about 10MB is not unusual.

No space is required under normal operation for trace files. Trace files are requested by Stonebranch, Inc. [Customer Support](#) only for problem resolution. When trace files are required, at least 20MB of disk space should be available.

Security

Universal Broker and the Broker components (Universal Command Server and Universal Control Server) require read/write access to the trace subvolume. No other Universal Products access the trace subvolume. No general user access is required.

7.4 Universal Command for HP NonStop – Product Distribution File

The product distribution file contains the installation files required for the installation of Universal Command for HP NonStop.

To obtain the distribution file, you must download it from the Support / Current Products area of the Stonebranch, Inc. website: <http://www.stonebranch.com>.

Note: A customer user ID and password – provided by Stonebranch, Inc. – are required to access this area.

Stonebranch, Inc. provides a separate product distribution file for the following HP NonStop systems:

- NSK S-series system
- Integrity system

7.4.1 NSK S-series System

The Universal Products for HP NonStop (NSK S-series systems) product distribution file is in a Windows, self-extracting archive file format.

The name of the distribution file has the following format:

ucmd-Version.Release.Modification Level.Maintenance Level-operating system-platform.exe

In this format:

- **Version** is the current version of Universal Products.
- **Release** is the current release of Universal Products.
- **Modification Level** is the current Universal Products feature set.
- **Maintenance Level** is the Universal Products build level.
- **operating system** is the name of the operating system.
- **platform** is the targeted hardware platform.

For example, the name of the distribution file for Universal Products 2.1.1 Level 2, HP NonStop version NSK, release G06 would be:

ucmd-2.1.1.2-tandem-NSK-G06.exe

Extracting the Installation Files

To extract the installation files from the distribution file, simply execute the file.

7.4.2 Integrity System

The Universal Products for HP NonStop (Integrity systems) product distribution file is in a compressed `tar` format.

The name of the distribution file has the following format:

ucmd-Version.Release.Modification Level.Maintenance Level-operating system-platform.tar.Z

In this format:

- **Version** is the current version of Universal Products.
- **Release** is the current release of Universal Products.
- **Modification Level** is the current Universal Products feature set.
- **Maintenance Level** is the Universal Products build level.
- **operating system** is the name of the operating system.
- **platform** is the targeted hardware platform.

For example, the name of the distribution file for Universal Products 2.1.1 Level 3, HP NonStop version NSK, release H06 would be:

ucmd-2.1.1.3-tandem-NSK-H06.tar.z

Extracting the Installation Files

To uncompress and extract the installation files from the distribution file, issue the following command:

```
zcat ucmd-2.1.1.3-tandem-NSK-H06.tar.z | tar xvf -
```

This command assumes that:

- Name of the distribution file is **ucmd-2.1.1.3-tandem-NSK-H06.tar.z**.
- File is located in the current working directory.

7.4.3 Distribution File Contents

[Table 7.2](#), below, lists the installation files included in the distribution file for Universal Command 2.1.1 for HP NonStop (NSK S-series and Integrity systems).

File Name	Description
<code>Readme.unv</code>	Summary of the installation procedure.
<code>ucmdins</code>	Installation script.
<code>ucmdpkg</code>	Installation package file.

Table 7.2 Universal Command for HP NonStop - Installation Files

If your Universal Products for HP NonStop distribution file does not contain these files, contact Stonebranch, Inc. [Customer Support](#) to obtain a correct distribution file.

7.5 Universal Command for HP NonStop – Installation

Installation of Universal Command 2.1.1 for HP NonStop is performed with the Universal Command installation script, `ucmdinst`.

To install Universal Command, perform the following steps:

1. Transfer the installation files (in a BINARY transfer mode, not ASCII) to the HP NonStop operating system.
2. To start the install as user `super.super`, the installation script file (`ucmdins`) must be converted to a Guardian EDIT file by using the `ctoedit` function.

Execute the following commands from the subvolume in which the files are located:

```
fup alter ucmdins, code 180
ctoedit ucmdins, ucmdinst
```

This prepares the script to be used with the `obey` command.

3. To start the installation, execute the following `obey` command:
`obey ucmdinst`

This copies the files to the proper locations, after which the system will be ready to configure.

7.6 Universal Command for HP NonStop – Customization

This section provides information on customizing the Universal Products components that comprise Universal Command 2.1.1 for HP NonStop.

7.6.1 Universal Broker Customization

Configuration

Configuration options for Universal Broker are stored in configuration file, **UBRCFG**, in subvolume **\$SYSTEM.UNVCONF**.

See the Universal Broker 4.1.0 User Guide for details on configuring Universal Broker.

Once installed, the **\$SYSTEM.UNVBIN.UBROKERD** startup script should be modified and the appropriate time zone set for the **TZ** variable.

Value	Description
North America	
EST5EDT	Eastern Standard Time, Eastern Daylight Time
CST6CDT	Central Standard Time, Central Daylight Time
MST7MDT	Mountain Standard Time, Mountain Daylight Time
PST8PDT	Pacific Standard Time, Pacific Daylight Time
AKST9AKDT	Alaska Standard Time, Alaska Daylight Time
Europe	
GMT0BST	Greenwich Mean Time, British Summer Time
WET0WEST	Western Europe Time, Western Europe Summer Time
CET-1CEST	Central Europe Time, Central Europe Summer Time
EET-2EES	Eastern Europe Time, Eastern Europe Summer Time

Table 7.3 Universal Products for HP NonStop - TZ Environment Variable Values

Note: Other common time zone abbreviations can be used. Not all possible values are indicated in this document.

System Initialization

A Broker daemon start-up script is provided in file **\$SYSTEM.UNVBIN.UBROKERD**. A single command line argument — either **-start**, **-stop**, **-status**, or **-restart** — instructs the script on the action to take.

See the Universal Broker 4.1.0 User Guide for details on the Universal Broker daemon script.

7.6.2 Universal Command Manager Customization

Configuration

Configuration options for Universal Command Manager are stored in configuration file, **UCMDCFG**, in subvolume **\$SYSTEM.UNVCONF**.

See the Universal Command 4.1.0 User Guide for details on configuring Universal Command Manager.

The product executable files intended for command line use are located in the subvolume **\$SYSTEM.UNVBIN**.

Product License

Universal Command Manager must be configured with a Universal Command product license before it is used. It will not run without a valid license.

The product license contains the license parameter keywords and their values, which must be entered into the Universal Command Manager configuration file, **UCMDCFG**. The values are specified in the same syntax as all other configuration options.

Product licenses are provided by your Stonebranch, Inc. account representative.

7.6.3 Universal Command Server Customization

Configuration

Configuration options for Universal Command Server are stored in configuration file, **UCMDSCFG**, in subvolume **\$SYSTEM.UNVCONF**.

See the Universal Command 3.1.1 User Guide for details on configuring Universal Command Server.

Universal Command Server runs as a component managed by Universal Broker. Universal Command Server provides a component definition file that Universal Broker uses to start the Server and establish its runtime environment. Component definition file **UCMD** is located in subvolume **\$SYSTEM.UNVCOMP**.

The product executable files intended for command line use are located in the subvolume **\$SYSTEM.UNVBIN**.

Universal Command Server uses the Universal Access Control List (UACL) configuration file as a level of product security. See the Universal Command 3.1.1 User Guide for information on how Universal Command Server utilizes the UACL file.

7.6.4 Universal Control Manager Customization

Configuration

Configuration options for Universal Control Manager are stored in configuration file, **UCTLCFG**, in subvolume **\$SYSTEM.UNVCONF**.

See the Universal Products Utilities 4.1.0 User Guide for details on configuring Universal Control Manager.

The product executable files intended for command line use are located in the subvolume **\$SYSTEM.UNVBIN**.

7.6.5 Universal Control Server Customization

Configuration

Configuration options for Universal Control Server are stored in configuration file, **UCTLSCFG**, in subvolume **\$SYSTEM.UNVCONF**.

See the Universal Products Utilities 4.1.0 User Guide for details on configuring Universal Control Server.

Universal Control Server runs as a component managed by Universal Broker. Universal Control Server provides a component definition file that Universal Broker uses to start the Server and establish its runtime environment. Component definition file **UCTL** is located in subvolume **\$SYSTEM.UNVCOMP**.

The product executable files intended for command line use are located in subvolume **\$SYSTEM.UNVBIN**.

Universal Control Server uses the Universal Access Control List (UACL) configuration file as a level of product security.

See the Universal Products Utilities 4.1.0 User Guide for information on how Universal Control Server utilizes the UACL file.

7.6.6 Universal Query Customization

Configuration

Configuration options for Universal Query are stored in configuration file, **UQRYCFG**, in subvolume **\$SYSTEM.UNVCONF**.

See the Universal Products Utilities 4.1.0 User Guide for details on configuring Universal Query.

The product executable files intended for command line use are located in the subvolume **\$SYSTEM.UNVBIN**.

7.7 Universal Command for HP NonStop – File Inventory Lists

The Universal Command for HP NonStop installation includes the files required for the following Universal Products components:

- Universal Broker
- Universal Command Manager and Server
- Universal Control Manager and Server
- Universal Query

This section identifies the files installed with each component.

Universal Broker

File	Description
<code>\$\$SYSTEM.UNVBIN.UBROKERD</code>	Broker daemon start script.
<code>\$\$SYSTEM.UNVBIN.UBROKER</code>	Console broker program.
<code>\$\$SYSTEM.UNVBIN.UBRD</code>	Daemon broker program.
<code>\$\$SYSTEM.UNVCOMP</code>	Component definition directory.
<code>\$\$SYSTEM.UNVCOMP.UCMD</code>	Universal Command Server component definition file.
<code>\$\$SYSTEM.UNVCOMP.UCTL</code>	Universal Control Server component definition file.
<code>\$\$SYSTEM.UNVNLS</code>	Code page files used for text translation between different operating systems and platforms and product message catalogs.
<code>\$\$SYSTEM.UNVLOG</code>	Broker message log directory.
<code>\$\$SYSTEM.UNVTRACE</code>	Broker trace file directory.
<code>\$\$SYSTEM.UNVCONF.UBRCFG</code>	Broker configuration file.

Table 7.4 Universal Broker for HP NonStop - File Inventory List

Universal Command Manager

File	Description
<code>\$\$SYSTEM.UNVBIN.UCMD</code>	Universal Command Manager program.
<code>\$\$SYSTEM.UNVBIN.UENCRYPT</code>	Universal Encrypt program file.
<code>\$\$SYSTEM.UNVNLS</code>	Code page files used for text translation between different operating systems and platforms and product message catalogs.
<code>\$\$SYSTEM.UNVCONF.UCMDCFG</code>	Manager configuration file.

Table 7.5 Universal Command Manager for HP NonStop - File Inventory List

Universal Command Server

File	Description
<code>\$\$SYSTEM.UNVBIN.UCMSRV</code>	Server component program.
<code>\$\$SYSTEM.UNVBIN.UCOPY</code>	Utility used for binary file copies. Similar to the UNIX cat command.
<code>\$\$SYSTEM.UNVBIN.UMET</code>	Universal Message Translator program.
<code>\$\$SYSTEM.UNVNLS</code>	Code page files used for text translation between different operating systems and platforms and product message catalogs.
<code>\$\$SYSTEM.UNVTRACE</code>	Server trace file directory.
<code>\$\$SYSTEM.UNVCONF.UCMDSCFG</code>	Server configuration file.

Table 7.6 Universal Command Server for HP NonStop - File Inventory List

Universal Control Manager

File	Description
<code>\$\$SYSTEM.UNVBIN.UCTL</code>	Manager program.
<code>\$\$SYSTEM.UNVNLS</code>	Code page files used for text translation between different operating systems and platforms and product message catalogs.
<code>\$\$SYSTEM.UNVCONF.UCTLCFG</code>	Manager configuration file.

Table 7.7 Universal Control Manager for HP NonStop - File Inventory List

Universal Control Server

File	Description
<code>\$\$SYSTEM.UNVBIN.UCTSRV</code>	Server component program.
<code>\$\$SYSTEM.UNVNLS</code>	Code page files used for text translation between different operating systems and platforms and product message catalogs.
<code>\$\$SYSTEM.UNVTRACE</code>	Server trace file directory.
<code>\$\$SYSTEM.UNVCONF.UCTLSCFG</code>	Server configuration file.

Table 7.8 Universal Control Server for HP NonStop - File Inventory List

Universal Query

File	Description
<code>\$\$SYSTEM.UNVBIN.UQUERY</code>	Universal Query program.
<code>\$\$SYSTEM.UNVNLS</code>	Code page files used for text translation between different operating systems and platforms and product message catalogs.
<code>\$\$SYSTEM.UNVCONF.UQRYCFG</code>	Query configuration file.

Table 7.9 Universal Query for HP NonStop - File Inventory List

Chapter 6

OS/400 Installation

6.1 Overview

This chapter provides information on the installation of Stonebranch, Inc. Universal Products 4.1.0 on the OS/400 operating system. Unless otherwise specified, all references to Universal Products for OS/400 in this chapter refer to version 4.1.0.

All Universal Products for OS/400 are provided in an easily installed, one-time installation package.

Note: Starting with the 3.2.0 release of Universal Products, a Universal Broker must run on all systems on which a Universal Product component is running, including manager components. The local Broker maintains product configuration data for all components that have a configuration file.

6.1.1 Naming Conventions

In Universal Products 4.1.0 for OS/400, some library names and object names include a **vr_m** (version, release, and modification) suffix, **410**. This **vr_m** suffix will change, as appropriate, for subsequent releases.

Additionally, some library names can be changed. This chapter uses the default names for these libraries:

- **UNVPRD410** (product library)
- **UNVTMP410** (temporary library)
- **UNVSPL410** (spool file library)
- **UNVCMDREF** (command reference library)

You can change the **UNVPRD410**, **UNVTMP410**, and **UNVSPL410** library names at installation time.

You can change **UNVCMDREF** only via the Universal Command Server configuration file.

6.2 Universal Products for OS/400 Package

The Universal Products 4.1.0 for OS/400 package includes the following Universal Products components:

- Universal Broker 4.1.0
- Universal Command (Manager and Server) 4.1.0
- Universal Control (Manager and Server) 4.1.0
- Universal Copy 4.1.0
- Universal Data Mover (Manager and Server) 4.1.0
- Universal Display Log File for AS/400 log files 4.1.0
- Universal Encrypt 4.1.0
- Universal Message Translator 4.1.0
- Universal Query 4.1.0
- Universal Submit Job with z/OS WTO support 4.1.0

Universal Products 4.1.0 for OS/400 is packaged with product identifier **0UNV410**. The product can be managed using the OS/400 commands for working with licensed programs (such as **RSTLICPGM**, **SAVLICPGM**, and **DLTLICPGM**).

Note: For the 4.1.0 release, Universal Command and Universal Encrypt are packaged as one OS/400 licensed program.

6.2.1 Product Compatibility

Table 6.1, below, identifies the compatibility of Universal Products 4.1.0 for OS/400 with previous product versions.

Product	Compatibility
Universal Broker 4.1.0	Universal Broker 3.2.0, 3.1.1, 3.1.0, and 1.2.1
Universal Command 4.1.0	Universal Command 3.2.0, 3.1.1, 3.1.0, and 1.2.1
Universal Control 4.1.0	Universal Control 3.2.0, 3.1.1, 3.1.0, and 1.2.1
Universal Copy 4.1.0	Universal Copy 3.2.0, 3.1.1, 3.1.0, and 1.2.1
Universal Data Mover 4.1.0	Universal Data Mover 3.2.0, 3.1.0, and 1.2.1
Universal Display Log File for OS/400 log files 4.1.0	Universal Display Log File for AS/400 log files 3.2.0, 3.1.1, 3.1.0, and 1.2.1
Universal Encrypt 4.1.0	Universal Encrypt 3.2.0, 3.1.1, 3.1.0, and 1.2.1
Universal Message Translator 4.1.0	Universal Message Translator 3.2.0, 3.1.1, 3.1.0, and 1.2.1
Universal Query 4.1.0	Universal Query 3.2.0, 3.1.1, 3.1.0, and 1.2.1
Universal Submit Job with z/OS WTO support 4.1.0	Universal Submit Job with z/OS WTO support 3.2.0, 3.1.1, 3.1.0, and 1.2.1

Table 6.1 Universal Products for OS/400 - Product Compatibility

The product references pertain to all support platforms for that version.

6.3 Universal Products for OS/400 – Installation Requirements

The requirements for installation of Universal Products 4.1.0 for OS/400 are:

- OS/400 V5R3M0 or above.
- TCP/IP.
- User profile with *ALLOBJ, *SPLCTL, *JOBCTL, and *SECADM authorities.
- About 360 megabytes of disk space.
- Network-attached workstation.

The Universal Products installation process creates a product user profile, **UNVUBR410**, that is given *ALLOBJ special authority. **UNVUBR410**, with *ALLOBJ special authority, is required to successfully complete the installation.

Note: Some organizations and companies require the removal of *ALLOBJ authority from non-administrative user profiles. Universal Products may be configured to run without *ALLOBJ authority; however, to do so requires additional administrative overhead. The steps required to remove product *ALLOBJ authority are described in the Universal Broker 4.1.0 User Guide.

The *BASE option contains the complete installation. This includes all of the components and utilities listed above.

6.3.1 Libraries and Logs

Under the OS/400 native file system, Universal Products write to product files residing in various libraries and write information to job logs.

This section specifies the following information for various libraries and logs:

- Estimated amount of space required
- Required security access
- Location of the production files

Universal Broker Job Log

Under OS/400, Universal Broker writes its messages to the **UBROKER** job log.

Space

Job log file growth is dependent on use of the Universal Products Servers. The disposition and size of the job log depends on the job definition as well as system variables **QJOBMSGQMX** (maximum job log size) and **QJOBMSGQFL** (action when job log is full).

Security

Since Universal Broker uses a normal job log, no special security is required.

Command Reference Library

Universal Command Server can execute commands of type **cmdref**. A command reference is a predefined command or script to which the Universal Command Manager refers by its file name.

The default command reference library name is **UNVCMDREF**. For security reasons, the name of this library can be changed only via the Universal Command Server configuration, located in the **UNVCONF** file and **UCMDS** member. The configuration file entry has precedence over the default value. If the name is changed to something other than the default or configured value, Universal Command Manager will no longer be able to read the command reference files; it will generate appropriate error messages if an attempt is made to do so.

Space

The amount of space is dependent solely on the number of command reference files defined.

Security

Universal Command Server requires read access to the **UNVCMDREF** library files. Administrator accounts require appropriate access in order to maintain the command reference files. No general user access is required.

Trace File Location Library

Universal Broker and its server components (for example, Universal Command Server) create product trace files when configured to do so. A trace file is used by Stonebranch, Inc. [Customer Support](#) to resolve product problems.

On OS/400, Universal Products trace files normally are written to *CURLIB (current library). Under Universal Broker and the associated servers, the current library is the temporary library designated at installation time (UNVTMP410).

Otherwise, current library is the default current library of the user.

Space

Trace files can grow to significant size, depending on how long the trace is active and how much work the program is doing during the tracing period. Due to the information associated with OS/400 pointers and fact that physical database files are fixed record lengths, trace files quickly can grow very large on a system with high Universal Broker and server activity.

If the trace file size is increased beyond 500,000 records, the maximum file size must be changed or the associated job will hang. The hang results from a system generated inquiry message, which is issued when the maximum file size is exceeded. By default, trace files wrap before reaching the maximum file size, thus avoiding the system inquiry message.

The MAX_TRACE_LINES configuration option sets the number of records at which the trace file wrapping occurs. When the maximum size is reached, the trace file will wrap to the beginning.

No space is required under normal operation for trace files. Trace files are requested by Stonebranch, Inc. [Customer Support](#) only for problem resolution. When trace files are required, at least 40MB of disk space should be available.

Security

The UNVUBR410 user profile requires at least *CHANGE authority to the UNVTMP410 library to create and use the Universal Broker and server trace files. No general user access is required.

Spool Library

The spool library is used to store the following types of information:

- Execution information for Universal Products components started by Universal Broker, **UBR_CMP_DB**.
- Universal Command Server status, **SRV_CMP_DB**.
- Redirected standard I/O files (stdin, stdout, and stderr) captured by Universal Command when run with manager fault tolerance enabled.

Spool files are stored in the **UNVSPL410** library.

Space

The spool files are located in the **UNVSPL410** library. The amount of disk space required for the spool directory depends on these factors:

1. Number of spooling user processes that will be executing simultaneously.
A user process is created for each command requested by a Universal Command Manager. The default maximum number is 50. Once a user process ends and a Manager has received all the spool files, the spool files themselves are deleted.
2. Average number of records of the user processes for standard input, output, and error files.
Each record is 32764 bytes in length. Keep in mind that spooling is not intended as a feature for file transfer purposes. File transfer related processes should execute without spooling enabled.

Once these numbers are determined, the average amount of disk space is calculated with the following formula:

$\text{MAX-PROCESSES} \times \text{AVERAGE NUMBER-OF-RECORDS} \times 32764 = \text{required disk space.}$

As an example, if the maximum number simultaneous user processes is estimated at 20 and the average number of records in files is 1,000 bytes, the average amount of required disk space is 655MB (20 x 1000 x 32764).

The Universal Command Server is configured with spooling disabled to prevent unintentional disk utilization. The feature must be turned on through the **ALLOW_SPOOLING** configuration settings.

Further details on the Manager Fault Tolerant feature and the spooling of redirected standard I/O files can be found in the Universal Command 4.1.0 User Guide.

Security

The **UNVUBR410** user profile requires at least ***CHANGE** authority to the **UNVSPL410** library to create and use the spool files. No general user access is required.

No other Universal Products access the spool library.

6.4 Universal Products for OS/400 – Product Distribution File

The Universal Products 4.1.0 for OS/400 product distribution file contains all of the files required for the installation of the Universal Products 4.1.0 for OS/400 package.

To obtain the Universal Products 4.1.0 for OS/400 package, you must download the corresponding product distribution file from the Support / Current Products area of the Stonebranch, Inc. website: <http://www.stonebranch.com>.

Note: A customer user ID and password — provided by Stonebranch, Inc. — are required to access this area.

6.4.1 Distribution File Format

The format of the Universal Products 4.1.0 for OS/400 distribution file name is different for a full release than it is for a maintenance release.

Full Release

For a full release installation, the product distribution file name has the following format:

unv-Version.Release.Modification Level.Maintenance Level-operating system-version.release.tar.Z

For example: **unv-4.1.0.1-as400-5.3.tar.Z**

In this format:

- **Version** is the current version of Universal Products.
- **Release** is the current release of Universal Products.
- **Modification Level** is the current Universal Products feature set.
- **Maintenance Level** is the Universal Products build level.
- **operating system** is the name of the operating system.
- **version.release** is the minimum supported version and release of the operating system.

Maintenance Release

For a maintenance release installation, the product distribution file name has a modified format:

- **-ptfs** is included after the **operating system** name.
- **version.release** is not included after the **operating system** name.

For example: **unv-4.1.0.2-as400-ptfs.tar.Z**

(See the [Transferring for a Maintenance Release](#) section for information about PTFs in maintenance release distribution.)

6.4.2 Distribution File Contents

The distribution file contains the installation files required for the installation of Universal Products for OS/400.

The Universal Products Licensed Program for OS/400 is distributed as an OS/400 Save File.

Depending on whether the installation is for a full release or a maintenance release of Universal Products for OS/400, the distribution file contains different installation files.

Full Release

[Table 6.2](#), below, lists the installation files included in the distribution file for a full release of Universal Products 4.1.0 for OS/400.

File Name	Description
UNV410 . SAVF	Universal Products Licensed Program Save File.
README . TXT	Summary of the installation procedure.

Table 6.2 Universal Products for OS/400 - Installation Files (Full Release)

Maintenance Release

[Table 6.3](#), below, lists the installation files included in the distribution file for a maintenance release of Universal Products 4.1.0 for OS/400.

File Name	Description
CUMUNV410 . SAVF	Universal Products Cumulative PTF Save File.
README . TXT	Summary of the installation procedure.

Table 6.3 Universal Products for OS/400 - Installation Files (Maintenance Release)

6.5 Universal Products for OS/400 – Transferring Universal Products to OS/400

In order to install Universal Products for OS/400, you must first transfer the Universal Products Licensed Program for OS/400 to the OS/400 operating system.

The transfer is comprised of three procedures:

1. Download the product distribution file from the Stonebranch, Inc. website (see [Section 6.4 Universal Products for OS/400 – Product Distribution File](#)) to Windows, UNIX, or OS/400 shell with tar and zcat utilities.
2. Extract all files from the distribution file.
3. Transfer the OS/400 Save File (the Universal Products Licensed Program) to a library on an iSeries.

This section provides separate transferring procedures for:

- [Transferring for a Full Release](#)
- [Transferring for a Maintenance Release](#)

6.5.1 Transferring for a Full Release

To transfer a full release package to OS/400 — whether for a new installation of a full release or an upgrade installation to a new release — perform the following steps:

1. Download the distribution file to a computer running a supported version of Windows or UNIX.
2. Extract the installation files from the distribution file:
 - To extract the installation files on UNIX, QShell, or PASE shell, run:

```
zcat *.Z | tar xvf -
```

Note: The space and - at the end (after xvf) is critical.
 - To extract the installation files on Windows, use a utility capable of extracting files from a .Z file.

The files will be extracted to the specified location.

3. Create a save file on the target OS/400 system in library **QGPL**.

On the OS/400 command line, execute the following command:

```
CRTSAVF FILE(QGPL/UNV410)
```

4. If needed, FTP the SAVF file extracted in Step 2 to the save file created in Step 3.

Note: The following example assumes that the SAVF file resides in:

```
c:\temp\Universal Products for as400
```

```
ftp your_as400
Name: your_name
Password: your_password
ftp> cd QGPL
ftp> bin
ftp> lcd "c:\temp\Universal Products for as400"
ftp> put UNV410.SAVF UNV410
ftp> quit
```

6.5.2 Transferring for a Maintenance Release

To upgrade Universal Products from a currently installed 4.1.0 release to 4.1.0 maintenance release, cumulative PTFs are used. These cumulative PTFs are distributed as OS/400 Save Files and can be downloaded from the Stonebranch, Inc. website: <http://www.stonebranch.com>. PTFs should be installed as user QSECOFR or a user with equivalent authority.

To transfer a maintenance release package to OS/400, perform the following steps:

1. Download the distribution file to a computer running a supported version of Windows or UNIX.
2. Extract the distribution file:
 - To extract the *.tar.Z file on UNIX, run: `zcat *.Z | tar xvf -`
Note: The space and - at the end (after `xvf`) is critical.
 - To extract the distribution file for Windows, use a utility capable of extracting files from a .Z file.

The files will be extracted to the specified location:

3. Create a save file on the target OS/400 system in library QGPL.

On the OS/400 command line, execute the following command:

```
CRTSAVF FILE(QGPL/CUMUNV410)
```

4. FTP the .SAVF file extracted in Step 2 to the save file created in Step 3.

For example:

```
ftp your_as400
Name: your_name
Password: your_password
ftp> cd QGPL
ftp> bin
ftp> lcd c:\temp
ftp> put CUMUNV410.SAVF CUMUNV410
ftp> quit
```

6.6 Universal Products for OS/400 – Installation

This section provides five different processes for installing Universal Products 4.1.0 on an OS/400 system:

1. [New Installation - Default](#)
2. [New Installation - Custom](#)
3. [Re-Installation of Same Release](#)
4. [Upgrade Installation to New Release](#)
5. [Upgrade Installation for Maintenance Release](#)

Your installation environment, currently installed Universal Products (if any), and Universal Products release to be installed determine the process to use.

This section also provides the following procedures related to installation:

- [UCHGRLS \(Change Release Tag\) Program](#) for changing Universal Products for OS/400 command names in the OS/400 **QSYS** library.
- [Product Removal](#) procedures that are required or optional, depending on the installation process.

6.6.1 New Installation - Default

The default process for a new installation of Universal Products 4.1.0 for OS/400 installs to the following libraries:

- Product library (**UNVPRD410**)
- Temporary library (**UNVTMP410**)
- Spool file library (**UNVSPL410**)
- Command reference library (**UNVCMDREF**)

1. Execute the following command to install Universal Products to the default libraries:

```
RSTLICPGM LICPGM(0UNV410) DEV(*SAVF) SAVF(QGPL/UNV410)
```

2. Verify that the installation was correct by executing the following command:

```
CHKPRDOPT 0UNV410
```

The following message should display:

```
No errors detected by CHKPRDOPT.
```

3. Optionally, use the [UCHGRLS \(Change Release Tag\) Program](#) to change the names of the Universal Products for OS/400 commands in the OS/400 QSYS library.

6.6.2 New Installation - Custom

The custom method for a new installation of Universal Products 4.1.0 for OS/400 allows installation to libraries other than **UNVPRD410**, **UNVTMP410**, and **UNVSPL410**, as required by your environment.

1. Execute the following command to install Universal Products to user-specified libraries:

```
RSTLICPGM LICPGM(0UNV410) DEV(*SAVF) SAVF(QGPL/UNV410) LIB(prodLib  
tmpLib spoolLib)
```

Note: Replace **prodLib** with the desired product library name, **tmpLib** with the desired temporary library name, and **spoolLib** with the desired spool library name.

2. Verify that the installation was correct by executing the following command:

```
CHKPRDOPT 0UNV410
```

The following message should display:

No errors detected by CHKPRDOPT.

3. Optionally, use the [UCHGRLS \(Change Release Tag\) Program](#) to change the names of the Universal Products for OS/400 commands in the OS/400 **QSYS** library.

6.6.3 Re-Installation of Same Release

Re-Installation Using Same Libraries

If you are reinstalling a Universal Products for OS/400 release (4.1.0 and later) into the same libraries, you do not have to remove the currently installed release.

Note: Re-installing to the same libraries will preserve configuration and other user-modified files, as well as objects created as part of the installation process. The re-installation process will replace programs, commands, etc.

Re-installation to the same libraries can be performed without deleting the current release.

1. Back up all file objects in all of the Universal Products libraries – product, temporary, spool, and command reference – prior to re-installation (in case errors occur during the re-installation process).
2. End the Universal Broker subsystem, **UNVUBR410**.
3. Run **RSTLICPGM** to install Universal Products using the library names used during the initial 4.1.0.0 installation.
4. After installation is complete, check the configuration and component files to ensure that the re-installation process preserved all changes previously made. Restore any files not correctly preserved from the backed up file objects.
5. Restart **UNVUBR410**.
6. Make sure that previous events needed by the customer are displayed via the I-Activity Monitor and that any disconnected jobs, as well as any appropriate MFT spool files, are correctly displayed using the Universal Query. If problems exist, restore the spool library from the backed up file objects.
7. If **UCHGRLS** was previously run to remove 410 tags from the Universal Products commands in **QSYS**, manually delete the newly created, 410-tagged Universal Product commands in **QSYS**. A list of those commands can be found in [Table 6.8 Universal Products for OS/400 - QSYS Library Object Inventory List](#).

Note: Do not remove the **UNVUBR410** user profile from **QSYS**.

Re-Installation Using Different Library Names

Before reinstalling a Universal Products for OS/400 release (4.1.0 and later) into different libraries, you must remove the currently installed release.

1. Save copies of all configuration files and component definition files; otherwise, the configuration and component information will be lost.
2. Remove the currently installed release (see [Universal Products 3.2.0](#) in Section [6.6.7 Product Removal](#)).
3. Re-install the 4.1.0 release. See Sections [6.6.1 New Installation - Default](#) and [6.6.2 New Installation - Custom](#) to determine which process is most appropriate for your environment.
4. Merge the saved configuration into the newly installed configuration files manually. Conversely, the saved configuration files can be copied in place of the reinstalled configuration files.
5. Optionally, use the [UCHGRLS \(Change Release Tag\) Program](#) to change the names of the Universal Products for OS/400 commands in the OS/400 QSYS library.

6.6.4 Upgrade Installation to New Release

To upgrade Universal Products for OS/400 from a currently installed pre-4.1.0 release (3.2.0 or earlier) to a 4.1.0 release, the following procedures are required:

1. [Install \(Default or Custom\)](#)
2. [Post-Install Tasks](#)
3. [Pre-Production Tasks](#)

Currently Installed Release

You are not required to delete a currently installed pre-4.1.0 release (3.2.0 or earlier) of Universal Products for OS/400 before installing a 4.1.0 release. You can do so either before or after installing the 4.1.0 release (see Section [6.6.7 Product Removal](#)).

However, if you want to install a 4.1.0 release to the previously used installation libraries of a pre-4.1.0 release, you must first uninstall that pre-4.1.0 release:

1. Save configuration, component, NLS, and any other files from the product library (**UNVPRD410**) that may contain information needed after the new installation.
2. Save any files from the command reference library (**UNVCMDREF**) and temporary library (**UNVTMP410**) that are needed, as these libraries will be deleted by the uninstall process.
3. Follow the release-appropriate uninstall process (see Section [6.6.7 Product Removal](#)) and the New Release installation procedure (see [Install \(Default or Custom\)](#), below) to install the new release.
4. After the new installation is complete, merge required configuration and any other saved information as well as any needed files and/or members from the previous release into the new release.

Install (Default or Custom)

The processes (default or custom) for the upgrade installation to a new Universal Products release (4.1.0) are the same as the processes for a new installation.

See Sections [6.6.1 New Installation - Default](#) and [6.6.2 New Installation - Custom](#) to determine which process is most appropriate for your environment.

Post-Install Tasks

After the installation upgrade of Universal Products 3.2.0 (or previous release) to Universal Products 4.1.0, perform the following steps:

1. Copy any customized code page files from the **UNVNLS** file members from the previous release to the newly installed **UNVNLS** file.

Note: Do NOT simply copy the old **UNVNLS** file to replace the new **UNVNLS** file. The new files contain many updates.

2. If needed, manually update the newly installed configuration file, **UNVCONF**, from the file of the previous release. The new file contains many new configuration options. At a minimum, the Universal Products license entries must be entered into the configuration file members for the managers of each licensed product.
3. If the original component file (**UNVCOMP**) was modified, update the newly installed **UNVCOMP** file for each of the modified members.
4. If either or both of the exit programs (**UCMSJOB1** and **UCMSJOB2**) in the previous release had been customized, you likewise must customize the newly installed exit programs. The exit programs are located in the product library, **UNVPRD410**.

Note: You cannot replace **UCMSJOB1** for 4.1.0 with programs of the same name from previous releases without a possible conflict. If you must use older programs because their source is no longer available, rename them and call them from the new exit program. See **UCMSJOB1** in the Universal Command 4.1.0 User Guide for additional information.

Pre-Production Tasks

Change Release Tags

When you are ready to take the new release to production, you can use the **UCHGRSL** (Change Release Tag) program to:

- Rename the Universal Products for OS/400 commands in **QSYS** to the untagged command names in **UNVPRD410**.
- Tag the pre-4.1.0 release command names in **QSYS** with the version / release / modification number of that release (for example, **320**).

See Section [6.6.6 UCHGRSL \(Change Release Tag\) Program](#) for information on using **UCHGRSL**.

Change Port Address

No two Universal Products installed under the same IBM-i can have the same port number associated with the Universal Broker subsystem (**UBROKER**). Change the default port address, 7887, to another port number for all but one of the **UBROKER** subsystems. Universal Products components and utilities intended to use the altered port number must specify a port number in the commands or program calls. Whether or not a command or program call is used depends on the platform from which the request is sent.

Note: Do not replace a configuration file with one from a different release.

6.6.5 Upgrade Installation for Maintenance Release

To upgrade Universal Products 4.1.0 for OS/400 from a currently installed release to a maintenance release, perform the following steps to install the cumulative PTFs:

1. If it is running, end the Universal Broker subsystem (**UNVUBR410**) on the OS/400 system.

On the OS/400, enter the following command:

```
====> ENDSBS SBS (UNVUBR410)
```

2. Load the PTF.

On the OS/400, enter the following command:

```
====> LODPTF LICPGM(0UNV410) DEV(*SAVF) SAVF(QGPL/CUMUNV410)
```

3. Apply the PTF.

On the OS/400, enter the following command:

```
====> APYPTF LICPGM(0UNV410)
```

This command applies all PTFs contained in the cumulative PTF file.

If you want to apply only selected PTFs, enter the following command:

```
====>APYPTF LICPGM(0UNV410) SELECT(ptf1 ptf2 ...)
```

This command applies the PTF temporarily.

If you want to apply the PTFs permanently after successful testing, re-execute the command with the parameter **APY (*PERM)**.

4. Restart **UNVUBR410**.

6.6.6 UCHGRLS (Change Release Tag) Program

All installations of Universal Products 4.1.0 for OS/400, except an upgrade for maintenance, place the Universal Products for OS/400 commands in the product library (**UNVPRD410**) and the OS/400 **QSYS** library.

Starting with the 3.2.0 release of Universal Products for OS/400, the command names in **QSYS** are tagged with the Universal Products for OS/400 **version / release / modification** number.

To maintain consistency across releases, you may prefer to use untagged names in your production environment. The **UCHGRLS** (Change Release Tag) program lets you change command names in the OS/400 **QSYS** library from tagged to untagged.

Using UCHGRLS

UCHGRLS resides in **UNVPRD410**. It provides two parameters, **SETTAGVRM** and **RMVTAGVRM**.

- **SETTAGVRM** specifies the release number of the release to be tagged.
- **RMVTAGVRM** specifies the release number of the tag to be removed.

Removing the tags changes the command names in **QSYS** to the normal production command names. (See [UCHGRLS Examples](#), below.)

For a new installation, specify the following **UCHGRLS** command options and their values:

- Set tags (**SETTAGVRM**) option; value = ***NEW**.
- Remove tags (**RMVTAGVRM**) option; value = **410**.

This changes the tagged command names in **QSYS** to the untagged command names in **UNVPRD410**.

For an upgrade installation (from a pre-4.1.0 release to 4.1.0), **UCHGRLS** tags the pre-4.1.0 release command names in **QSYS** with the **version / release / modification** number of that release (for example, **410**). See Section [6.8 Universal Products for OS/400 – Object Inventory Lists](#) for tagged command name details.

UCHGRLS Examples

This example runs **UCHGRLS** with the default values, following installation of **0UNV410** with **0UNV320** existing on the system.

```
UNVPRD410/UCHGRLS
```

This example runs **UCHGRLS** to revert back to **0UNV320** for the default names in **QSYS** with **0UNV410** installed. (This should be done prior to re-installing **0UNV410**, assuming that **0UNV320** remains installed on the system.)

```
UNVPRD410/UCHGRLS SETTAGVRM(410) RMVTAGVRM(320)
```

This example runs **UCHGRLS** to change the **0UNV410** commands in **QSYS** to the default names and to tag the **0UCM121** and **0UEN121** commands in **QSYS** with 121.

```
UNVPRD410/UCHGRLS SETTAGVRM(121) RMVTAGVRM(410)
```

This example runs **UCHGRLS** to set the **0UNV410** commands in **QSYS** to the default names, where **0UNV410** is the first Universal Products installation on an OS/400 system.

```
UNVPRD410/UCHGRLS SETTAGVRM(*NEW) RMVTAGVRM(410)
```

6.6.7 Product Removal

This section provides processes for the removal of current (4.1.0) and previous (3.2.0 or earlier) releases of Universal Products for OS/400:

- [Universal Encrypt 1.2.1](#)
- [Universal Command 1.2.1](#)
- [Universal Products 3.1.1](#)
- [Universal Products 3.2.0](#)

If you remove the currently installed release before installing or re-installing a 4.1.0 release, the removal process deletes the default command reference library, **UNVCMDFREF**, in which the command reference files are located.

(If you remove the currently installed release after installing or re-installing a 4.1.0 release, **UNVCMDFREF** is not deleted.)

To keep your command reference files from being deleted, rename **UNVCMDFREF** and change the library owner. Release 4.1.0 will let you specify this library – or any library – as your command reference library via the **CMD_REFERENCE_DIRECTORY** option.

Before removing a 4.1.0 release, check **CMD_REFERENCE_DIRECTORY** in the Universal Command server member (**UCMDS**) of the Universal Products configuration file (**UNVCONF**) – located in the product library (**UNVPRD410**); an alternate command reference library previously may have been selected. If it has, manually clear and delete the default command reference library, **UNVCMDFREF**, after removing the 4.1.0 release.

WARNING: If the **UCHGRSL** command was used to rename the 410-tagged names to normal production names (see Section [6.6.6 UCHGRSL \(Change Release Tag\) Program](#)), you must perform the following steps prior to uninstalling either 3.2.0, 3.1.1, or 1.2.1:

1. Run the **UCHGRSL** command to temporarily rename the 4.1.0 Universal Products commands in **QSYS** to the 410-tagged names and the tagged names of the older product to the normal production names. Without this step, Universal Products may no longer run as expected.
2. After removal of the older product is complete, use the ***NEW** option with the **UCHGRSL** command to change the 410-tagged names back to the normal, production names.

Universal Encrypt 1.2.1

Note: Before performing this uninstall, read the text at the beginning of this section (Section [6.6.7 Product Removal](#)).

You must uninstall Universal Encrypt 1.2.1 before uninstalling Universal Command 1.2.1.

The user ID used for uninstalling Universal Encrypt must have authority to delete all Universal Encrypt product-related objects.

To uninstall Universal Encrypt, perform the following steps:

1. To remove the licensed program, execute the following command:
====>DLTLICPGM LICPGM(OUEN121)
2. If the following message displays, use the **DSPJOBLOG** command to identify the problem:
Objects for product OUEN121 option *ALL release *ONLY not deleted."
Correct the problem and repeat Step 1.

Universal Command 1.2.1

Note: Before performing this uninstall, read the text at the beginning of this section (Section [6.6.7 Product Removal](#)).

Before uninstalling Universal Command 1.2.1, you must uninstall Universal Encrypt 1.2.1.

The user ID used for uninstalling Universal Command must have *ALLOBJ and *SECADM special authorities.

To uninstall Universal Command, perform the following steps:

1. If it is active, end the Universal Broker subsystem (**UBROKER**).
On the OS/400 command line, execute the following command:
====>ENDSBS UBROKER
2. To remove the licensed program, execute the following command:
====>DLTLICPGM LICPGM(OUCM121)
3. If the following message displays, use the **DSPJOBLOG** command to identify the problem:
Objects for product OUCM121 option *ALL release *ONLY not deleted."
Correct the problem and repeat Step 2.

Universal Products 3.1.1

Note: Before performing this uninstall, read the text at the beginning of this section (Section [6.6.7 Product Removal](#)).

The user ID used for uninstalling Universal Products 3.1.1 must have *ALLOBJ and *SECADM special authorities.

To uninstall Universal Products 3.1.1, perform the following steps:

1. If it is active, end the Universal Broker subsystem (**UBROKER**).

On the OS/400 command line, execute the following command:

```
====>ENDSBS UBROKER
```

2. To remove the licensed program, execute the following command:

```
====>DLTLICPGM LICPGM(OUNV311)
```

3. If the following message displays, use the **DSPJOBLOG** command to identify the problem:

```
Objects for product OUNV311 option *ALL release *ONLY not deleted."
```

Correct the problem and repeat Step 2.

Universal Products 3.2.0

Note: Before performing this uninstall, read the text at the beginning of this section (Section [6.6.7 Product Removal](#)).

The user ID used for uninstalling Universal Products 3.2.0 must have *ALLOBJ and *SECADM special authorities.

To uninstall Universal Products 3.2.0, perform the following steps:

1. If it is active, end the Universal Broker subsystem (**UNVUBR320**).

On the OS/400 command line, execute the following command:

```
====>ENDSBS UNVUBR320
```

2. To remove the licensed program, execute the following command:

```
====>DLTLICPGM LICPGM(OUNV320)
```

3. If the following message displays, use the **DSPJOBLOG** command to identify the problem:

```
Objects for product OUNV320 option *ALL release *ONLY not deleted."
```

Correct the problem and repeat Step 2.

Universal Products 4.1.0

Note: Before performing this uninstall, read the text at the beginning of this section (Section [6.6.7 Product Removal](#)).

The user ID used for uninstalling Universal Products 4.1.0 must have *ALLOBJ and *SECADM special authorities.

To uninstall Universal Products 4.1.0, perform the following steps:

1. If it is active, end the Universal Broker subsystem (**UNVUBR410**).

On the OS/400 command line, execute the following command:

```
====>ENDSBS UNVUBR410
```

2. To remove the licensed program, execute the following command:

```
====>DLTLICPGM LICPGM(OUNV410)
```

3. If the following message displays, use the **DSPJOBLOG** command to identify the problem:

```
Objects for product OUNV410 option *ALL release *ONLY not deleted."
```

Correct the problem and repeat Step 2.

6.7 Universal Products for OS/400 – Customization

Modify the IPL Start-up Program so that the Universal Broker subsystem (**UNVUBR410**) is started at IPL time.

The subsystem start command is:

```
QSYS/STRSBS UNVPRD410/UNVUBR410
```

The IPL Start-up program name is displayed by:

```
DSPSYSVAL SYSVAL (QSTRUPPGM)
```

Refer to the AS/400 publication [AS/400e series: Basic System Operation, Administration, and Problem Handling](#) for complete details on modifying the IPL Start-up program.

6.7.1 Multiple-Installation Implementation

Universal Products 4.1.0 for OS/400 provides the ability to install, configure, and test a new release prior to placing it into production.

You can keep an older release available during this initial production phase in case problems occur. This allows you to manually merge component files, configuration files, and exit programs without having to save copies of those files and programs prior to installation of the newer release.

When initially installed, the names of the Universal Products commands in the **QSYS** library are tagged with the **version / release / modification number, 410**. This renaming allows currently installed pre-4.1.0 releases to remain fully functional without modifying production code until pre-production testing is completed.

When you are ready to bring the newly installed release into production, run the [UCHGRLS \(Change Release Tag\) Program](#) to untag the 4.1.0 command names and tag the pre-4.1.0 command names with that pre-4.1.0 release number.

After running **UCHGRLS**, check the job log for errors because missing objects do not result in program termination. There will generally be one or more missing commands flagged; only those commands included with the installed release present a problem if they are missing.

Universal Broker

For the Universal Broker, both a job queue (*JOBQ) and a class (*CLS) were added; they are both named **UBROKER** and are located in the product library, **UNVPRD410**.

The **UNVUBR410** subsystem and Universal Products job descriptions use the **UNVPRD410 / UBROKER** class and job queue instead of the **QGPL / QBATCH** class and job queue.

To revert back to the **QGPL / QBATCH** class:

1. Invoke the CHGRTGE command using **SBSD(UNVPRD410 / UNVUBR410)**, **SEQNBR(10)**, and **CLS(QGPL / QBATCH)**.
2. Invoke the CHGPJE command using **SBSD(UNVPRD410 / UNVUBR410)**, **PGM(QSYS / QP0ZSPWP)**, **JOB(UCMSINIT)**, and **CLS(QGPL / QBATCH)**.
3. Invoke the CHGPJE command using **SBSD(UNVPRD410 / UNVUBR410)**, **PGM(QSYS / QP0ZSPWT)**, **JOB(UNVSRV)**, and **CLS(QGPL / QBATCH)**.
4. Use the RMVJOBQE command with **SBSD(UNVPRD410 / UNVUBR410)** and **JOBQ(UNVPRD410 / UBROKER)** to remove the job queue entry from the **UNVUBR410** subsystem.

To revert back to the **QGPL / QBATCH** job queue:

1. Invoke the CHGJOB command specifying **JOBQ(QGPL / QBATCH)** for the **UNVPRD410 / UBROKER**, **UNVPRD410 / UCMSINIT**, and **UNVPRD410 / UNVSRV** job descriptions.

Active Jobs

Whereas the **QGPL / QBATCH** job queue entry for the **QBASE** subsystem limits the maximum active jobs to only six (by default), the **UNVUBR410 / UBROKER** job queue has no limit. Thus, it allows many additional Universal Products jobs to run in parallel. Using the new settings, the Universal Broker **RUNNING_MAX** configuration option controls the maximum number of components allowed to run simultaneously.

For consistency with earlier releases, the initial settings of the **UNVUBR410** class is the same as the **QGPL / QBATCH** class. Using the new class and job description, you may better tune your systems without affecting those jobs running under the default **QGPL / QBATCH** class and job description.

6.7.2 Universal Broker Customization

Configuration

Configuration options for Universal Broker are stored in configuration file **UNVCONF(UBROKER)**, in product library **UNVPRD410**.

See the Universal Broker 4.1.0 User Guide for details on configuring Universal Broker.

The file can be edited using the Source Entry Utility (SEU).

System Initialization

Universal Broker runs as a job, **UBROKER**, in the Universal Broker subsystem (**UNVUBR410**) located in product library **UNVPRD410**.

To start Universal Broker, enter the following on the command line: **STRSBS UNVPRD410/UNVUBR410**. There are two sets of pre-start jobs initiated: **UNVSRV** and **UCMSINIT**:

- **UNVSRV** is the pre-start job for any of the Universal Product servers.
- **UCMSINIT** is the pre-start job for the **UCMSINIT** program.

The Universal Command Server initiates **UCMSINIT**, which initiates and monitors the command originated from either the Universal Command Manager or the **exec** command under the Universal Data Mover Manager.

6.7.3 Universal Command Manager Customization

Configuration

Configuration options for Universal Command Manager are stored in configuration file **UNVCONF(UCMD)**, in product library **UNVPRD410**.

See the Universal Command 4.1.0 User Guide for details on configuring Universal Command Manager.

The file can be edited using the Source Entry Utility (SEU).

Product License

Universal Command Manager must be configured with a Universal Command product license before it is used. It will not run without a valid license.

The product license contains the license parameter keywords and their values, which must be entered into the Universal Command Manager configuration file, **UNVCONF(UCMD)**. The values are specified in the same syntax as all other configuration options.

Product licenses are provided by your Stonebranch Inc. account representative.

6.7.4 Universal Command Server Customization

Configuration

Configuration options for Universal Command Server are stored in configuration file **UNVCONF(UCMDS)**, in product library **UNVPRD410**.

See the Universal Command 4.1.0 User Guide for details on configuring Universal Command Server.

Universal Command Server runs as a component managed by Universal Broker. It provides a component definition file that Universal Broker uses to start the Server and establish its runtime environment. The component definition file, **UNVCOMP(UCMD)**, is located in product library **UNVPRD410**.

Universal Command Server uses the Universal Access Control List (UACL) configuration file as a level of product security. The UACL file is located in **UNVCONF(UACL)**, in product library **UNVPRD410**. See the Universal Command 4.1.0 User Guide for information on how Universal Command Server utilizes the UACL file.

The file can be edited using the Source Entry Utility (SEU).

6.7.5 Universal Control Manager Customization

Configuration

Configuration options for Universal Control Manager are stored in configuration file **UNVCONF(UCTL)**, in product library **UNVPRD410**.

See the Universal Products Utilities 4.1.0 User Guide for details on configuring Universal Control Manager.

The file can be edited using the Source Entry Utility (SEU).

6.7.6 Universal Control Server Customization

Configuration

Configuration options for Universal Control Server are stored in configuration file **UNVCONF(UCLS)**, in product library **UNVPRD410**.

See the Universal Products Utilities 4.1.0 User Guide for details on configuring Universal Control Server.

Universal Control Server runs as a component managed by Universal Broker. It provides a component definition file that Universal Broker uses to start the Server and establish its runtime environment. Component definition file **UNVCOMP(UCTL)** is located in library **UNVPRD410** by default

Universal Control Server uses the Universal Access Control List (UACL) configuration file as a level of product security. The UACL file **UNVCONF(UACL)** is located in product library **UNVPRD410**. See the Universal Products Utilities 4.1.0 User Guide for information on how Universal Control Server utilizes the UACL file.

The file can be edited using the Source Entry Utility (SEU).

6.7.7 Universal Data Mover Manager Customization

Configuration

Configuration options for Universal Data Mover Manager are stored in configuration file **UNVCONF(UDM)**, in product library **UNVPRD410**.

See the Universal Data Mover 4.1.0 User Guide for details on configuring Universal Data Mover Manager.

The file can be edited using the Source Entry Utility (SEU).

Product License

Universal Data Mover Manager must be configured with a Universal Data Mover product license before it is used. It will not run without a valid license.

The product license contains the license parameter keywords and their values, which must be entered into the Universal Data Mover Manager configuration file, **UNVCONF(UDM)**. The values are specified in the same syntax as all other configuration options.

Product licenses are provided by your Stonebranch, Inc. account representative.

6.7.8 Universal Data Mover Server Customization

Configuration

Configuration options for Universal Data Mover Server are stored in configuration file **UNVCONF(UDMS)**, in product library **UNVPRD410**.

Universal Data Mover Server runs as a component managed by Universal Broker. It provides a component definition file that Universal Broker uses to start the Server and establish its runtime environment. Component definition file **UNVCOMP(UDM)** is located in product library **UNVPRD410**.

Universal Data Mover Server uses the Universal Access Control List (UACL) configuration file as a level of product security. The UACL file is located in **UNVCONF(UACL)** in product library **UNVPRD410**.

See the Universal Data Mover 4.1.0 User Guide for information on how the Universal Data Mover Server utilizes the UACL file and for details on configuring Universal Data Mover Server.

The file can be edited using the Source Entry Utility (SEU).

6.7.9 Universal Query Customization

Configuration

Configuration options for Universal Query are stored in configuration file **UNVCONF(UQRY)**, in product library **UNVPRD410**.

See the Universal Products Utilities 4.1.0 User Guide for details on configuring Universal Query.

The file can be edited using the Source Entry Utility (SEU).

6.7.10 National Language Customization

The default OS/400 CCSID value (QCCSID) is pre-set to 65535 or *HEX, both of which mean: Do not interpret data unless specifically required. You can set QCCSID to a value consistent with your country CCSID.

However, Universal Products configuration files contain non-invariant characters (such as #, which marks the start of a comment). Some CCSIDs, such as 278, result in the translation of these characters into characters that the Universal Products software cannot recognize. As a result, Universal Products components, including Universal Broker, fail to start.

To resolve this, either:

1. Use a Universal Broker job CCSID of 65535. Both the Universal Products configuration files and the internal Universal Broker code are CCSID 037-based. Start Universal Broker and check the job log output to verify the correct translation.
2. Use 037 as the CCSID for the Universal Broker user profile (**UNVUBR410**). This sets the CCSID of the associated Universal Broker job to 037. The job's 37 CCSID forces compatibility with the configuration files and the internal program CCSID value of 037, which is consistent with standard OS/400 development practice.

The **CRTPGM** command sets the program CCSID to 65535, which allows the program to inherit the job CCSID. Writing to the job log should result in the correct translation of characters between the job CCSID and the system CCSID.

6.8 Universal Products for OS/400 – Object Inventory Lists

The Universal Products for OS/400 installation includes the objects required for:

- Universal Broker
- Universal Command Manager and Server
- Universal Control Manager and Server
- Universal Data Mover Manager and Server
- Universal Encrypt

These objects are contained in four user-designated installation libraries:

- Product library **UNVPRD410** (formerly **UNIVERSAL**)
Library containing all Universal Products for OS/400 products.
- Temporary library **UNVTMP410** (formerly **UNVTMP**)
Library used as a temporary work area by Universal Products. It may contain such items as trace files and temporary job log files.
- Spool library **UNVSPL410** (formerly **UNVSPool**)
Library containing a spool database of all Universal Products spool files. The database is allocated the first time that Universal Broker is started.
- Command reference library **UNVCMDREF**
Library containing user-defined command references

In addition, an installation places the following in the OS/400 **QSYS** library:

- Copy of the Universal Products for OS/400 commands contained in **UNVPRD410**.
- Universal Broker user profile, **UNVUBR410**.

The names of the commands in **QSYS** are tagged with the Universal Products for OS/400 version / release / modification number, **410**. However, since command names can have a maximum nine characters, the following names are shortened:

- **UDSPLOGF** becomes **UDSPLF410**.
- **USBMJOB** becomes **USBMJ410**.
- **UMSGHDLR** becomes **UMSGH410**.
- **UJOBINIT** becomes **UJOBI410**.

Note: For further information on library and object names, see Section [6.1.1 Naming Conventions](#).

Source File Record Lengths

Universal Products 4.1.0 for OS/400 source file record lengths must be a maximum 240 characters in order to use the Source Edit Utility, SEU.

The configuration file (**UNVCONF**), component file (**UNVCOMP**), template file (**UNVTMPL**), National Language Support file (**UNVNLS**), and the source files – all of which are located in the Universal Products installation library (**UNVPRD410**, by default) – are editable by SEU as they are initially installed.

Product Library (UNVPRD410)

Name	Type	Description
PRCDSPL0GF	*PGM	Universal Display Log File program.
PUMSGHNDLR	*PGM	Universal Submit Job, WTO msg callback program.
STRUCP	*PGM	UCOPY command program.
STRUEN	*PGM	Universal Encrypt command program.
STRUME	*PGM	UMET command program.
UBROKER	*PGM	*IN USE
UCMCP	*PGM	UCMD Manager command processing program.
UCHGRS	*PGM	Universal Change Release program.
UCMD	*PGM	UCMD Manager program.
UCMDEF01	*PGM	UCMD object definition program.
UCMDEXIT01	*PGM	UCMD install exit program.
UCMSG01	*PGM	UCMD message definition program.
UCMSINIT	*PGM	UCMD Server Initiator program.
UCMSJOBI	*PGM	UCMD Server Initiator job initialization program.
UCMSJOB	*PGM	UCMD Server Initiator job termination program.
UCMSRV	*PGM	Universal Command Server program.
UCOPY	*PGM	Universal Copy program.
UCTCP	*PGM	UCTL Manager command processing program.
UCTL	*PGM	Universal Control Manager program.
UCTSRV	*PGM	Universal Command Server program.
UDM	*PGM	Universal Data Mover Manager program.
UDMCP	*PGM	UDM Manager command processing program.
UDMSRV	*PGM	UDM Server program.
UDSPL0GF	*PGM	Universal Display Log File cmd processing program.
UENCRYPT	*PGM	Universal Encrypt program.
UJOBINIT	*PGM	USBMJOB initialization program.
UMET	*PGM	Universal Message Translator program.
UMSGHNDLR	*PGM	Universal Submit Job, PUMSGHNDLR interface.
UQRCP	*PGM	Universal Query command program.
UQUERY	*PGM	Universal Query program.
USBMJOB	*PGM	Universal Submit Job program.
USBMUSRJOB	*PGM	Universal Submit User Job program.
UCINST	*MSGF	UCMD install message file.
UNVMSG	*MSGF	Universal products message file.
USBMJOB	*MSGF	Universal Submit Job message file.
CP2CCSID_X	*FILE	Example code page to CCSID mapping file.
UCMNLSTMP	*FILE	Temporary National Language Support file.

Name	Type	Description
UNVCLSRC	*FILE	CL source.
UNVCOMP	*FILE	Universal component definition members.
UNVCONF	*FILE	Universal configuration members.
UNVDDSSRC	*FILE	DDS Src for Stonebranch Products.
UNVNLS	*FILE	National Language Support.
UCMSVRPRTF	*FILE	Universal Command server job log print file.
USBMJPRTF	*FILE	USBMJOB job log print file.
UBROKER	*JOB	Job Definition for Universal Broker.
UCMSINIT	*JOB	Job Definition for UCMD Server Initiator.
UNVSRV	*JOB	Job Definition for Universal Servers.
STRUCM	*CMD	Universal Command Manager command.
STRUCP	*CMD	Universal Copy command.
STRUCT	*CMD	Universal Control Manager command.
STRUDM	*CMD	Universal Data Mover Manager command.
STRUEN	*CMD	Universal Encrypt command.
STRUME	*CMD	Universal Message Translator command.
STRUQR	*CMD	Universal Query command.
UCHGRLS	*CMD	Universal Change Release command.
UDSPLOGF	*CMD	Universal Display Log File command.
UJOBINIT	*CMD	USBMJOB initialization program command.
UMSGHNDLR	*CMD	WTO message callback program command.
USBMJOB	*CMD	Universal Submit Job command.
UBROKER	*SBSD	Subsystem Definition for Universal Broker
STRUCM	*PNLGRP	UCMD Manager panel definition.
STRUCP	*PNLGRP	UCOPY panel definition.
STRUCT	*PNLGRP	UCTL Manager panel definition.
STRUDM	*PNLGRP	UDM Manager panel definition.
STRUEN	*PNLGRP	Universal Encrypt panel definition.
STRUME	*PNLGRP	UMET panel definition.
STRUQR	*PNLGRP	Universal Query panel definition.
UCHGRLS	*PNLGRP	Universal Change Release panel definition.
UDSPLOGF	*PNLGRP	Universal Display Log File panel definition.
UJOBINIT	*PNLGRP	USBMJOB initialization program panel definition.
UMSGHNDLR	*PNLGRP	WTO message callback program panel definition.
USBMJOB	*PNLGRP	Universal Submit Job panel definition.
UCM410	*PRDDFN	Universal Products product definition.
UCM410	*PRDL0D	Universal Products product load.

Table 6.4 Universal Products for OS/400 - Product Library (UNVPRD410) Object Inventory List

Temporary Library (UNVTMP410)

Name	Type	Description
UCMDEF02	*PGM	UNVTMP object definition program.
UCMDEXIT02	*PGM	UNVTMP install exit program.

Table 6.5 Universal Products for OS/400 - Temporary Library (UNVTMP410) Object Inventory List

Spool Library (UNVSPL410)

Name	Type	Description
UNVDEFSP	*PGM	UNVSPL410 object definition program.
UNVEXITSPL	*PGM	UNVSPL410 install exit program.

Table 6.6 Universal Products for OS/400 - Spool Library (UNVSPL410) Object Inventory List

Command Reference Library (UNVCMDREF)

Name	Type	Description
UNVCRFSRC	*FILE	Universal Command command reference file.

Table 6.7 Universal Products for OS/400 - Command Reference Library (UNVCMDREF) Object Inventory List

QSYS Library

Name	Type	Description
STRUCM410	*CMD	Universal Command Manager command.
STRUCP410	*CMD	Universal Copy command.
STRUCT410	*CMD	Universal Control Manager command.
STRUDM410	*CMD	Universal Data Mover Manager command.
STRUEN410	*CMD	Universal Encrypt command.
STRUME410	*CMD	Universal Message Translator command.
STRUQR410	*CMD	Universal Query command.
UDSPLF410	*CMD	Universal Display Log File command.
UJOB410	*CMD	Universal Job Initializer command.
UMSGH410	*CMD	Universal Message Handler command.
USBMJ410	*CMD	Universal Submit Job command.
UNVUBR410	*USRPRF	Universal Broker user profile.

Table 6.8 Universal Products for OS/400 - QSYS Library Object Inventory List

Appendix A

Licenses and Copyrights

A.1 Overview

This appendix provides the following license files for the third-party libraries used within Universal Products:

- [Berkeley DB License](#)
- [OpenSSL License](#)
- [zlib License](#)

A.2 Berkeley DB License

The Berkeley DB library is used in binary form on z/OS, Windows, and UNIX ports. The following is the Berkeley DB library license.

```
/*
 * Copyright (c) 1990-2005
 * Sleepycat Software. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 *
 * 3. Redistributions in any form must be accompanied by information on
 * how to obtain complete source code for the DB software and any
 * accompanying software that uses the DB software. The source code
 * must either be included in the distribution or be available for no
 * more than the cost of distribution plus a nominal fee, and must be
 * freely redistributable under reasonable conditions. For an
 * executable file, complete source code means the source code for all
 * modules it contains. It does not include source code for modules or
 * files that typically accompany the major components of the operating
 * system on which the executable file runs.
 *
 * THIS SOFTWARE IS PROVIDED BY SLEEPYCAT SOFTWARE ``AS IS'' AND ANY EXPRESS
 * OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR
 * NON-INFRINGEMENT, ARE DISCLAIMED. IN NO EVENT SHALL SLEEPYCAT SOFTWARE
 * BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
 * THE POSSIBILITY OF SUCH DAMAGE.
 */
```

/*

* Copyright (c) 1990, 1993, 1994, 1995

* The Regents of the University of California. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. Neither the name of the University nor the names of its contributors

* may be used to endorse or promote products derived from this software

* without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND

* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE

* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

* SUCH DAMAGE.

*/

/*

* Copyright (c) 1995, 1996

* The President and Fellows of Harvard University. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

- * 3. Neither the name of the University nor the names of its contributors
- * may be used to endorse or promote products derived from this software
- * without specific prior written permission.
- *
- * THIS SOFTWARE IS PROVIDED BY HARVARD AND ITS CONTRIBUTORS ``AS IS" AND
- * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- * ARE DISCLAIMED. IN NO EVENT SHALL HARVARD OR ITS CONTRIBUTORS BE LIABLE
- * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
- * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
- * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
- * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
- * SUCH DAMAGE.
- */

A.3 OpenSSL License

The OpenSSL toolkit is used in binary form on z/OS, Windows, UNIX, and OS/400 ports. The following is the OpenSSL license.

```
/* =====  
* Copyright (c) 1998-2007 The OpenSSL Project. All rights reserved.  
*  
* Redistribution and use in source and binary forms, with or without  
* modification, are permitted provided that the following conditions  
* are met:  
*  
* 1. Redistributions of source code must retain the above copyright  
* notice, this list of conditions and the following disclaimer.  
*  
* 2. Redistributions in binary form must reproduce the above copyright  
* notice, this list of conditions and the following disclaimer in  
* the documentation and/or other materials provided with the  
* distribution.  
*  
* 3. All advertising materials mentioning features or use of this  
* software must display the following acknowledgment:  
* "This product includes software developed by the OpenSSL Project  
* for use in the OpenSSL Toolkit. (http://www.openssl.org/)"  
*  
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to  
* endorse or promote products derived from this software without  
* prior written permission. For written permission, please contact  
* openssl-core@openssl.org.  
*  
* 5. Products derived from this software may not be called "OpenSSL"  
* nor may "OpenSSL" appear in their names without prior written  
* permission of the OpenSSL Project.  
*  
* 6. Redistributions of any form whatsoever must retain the following  
* acknowledgment:  
* "This product includes software developed by the OpenSSL Project  
* for use in the OpenSSL Toolkit (http://www.openssl.org/)"  
*  
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
```

* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.

* =====

*
 * This product includes cryptographic software written by Eric Young
 * (eay@cryptsoft.com). This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).

*
 */

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are adhered to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.

```
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   "This product includes cryptographic software written by
*    Eric Young (eay@cryptsoft.com)"
*   The word 'cryptographic' can be left out if the routines from the library
*   being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*   the apps directory (application code) you must include an acknowledgement:
*   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publicly available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/
```

A.4 zlib License

The zlib library is used in binary form on z/OS, Windows, UNIX, and OS/400 ports.

The following is the zlib library license.

(C) 1995-2002 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly	Mark Adler
jloup@gzip.org	madler@alumni.caltech.edu

Appendix B

Customer Support

Stonebranch, Inc. provides customer support, via telephone and e-mail, for the installation of all Indesca / Infitran components.

E-MAIL

All Locations

support@stonebranch.com

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

www.stonebranch.com

TELEPHONE

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

North America

(+1) 678 366-7887, extension 6

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

Europe

+49 (0) 700 5566 7887

INDEX

A

ACF2 class (Universal Products for z/OS) *117*
adding
 Windows
 UEC Client Applications components
 via command line interface *242*
 via graphical interface *227*
 Universal Enterprise Controller
 components
 via command line interface *211*
 via graphical interface *197*
 Universal Products components
 via command line interface *174*
 via graphical interface *157*
allocating
 z/OS data sets *52*
 sample JCL *52*
associating started tasks with profiles (z/OS) *85*
authorization for z/OS libraries and programs
 87

B

backing up files before installation
 UNIX
 Universal Products for SOA *319*
 Windows
 Universal Products for SOA *249*
BYPASS parameter *78*

C

Change Release Tag (UCHGRSL) program

346
changing
 OS/400 command names *346*
 OS/400 library names *328*
command reference directory (UNIX) *275*
command reference library (OS/400) *331*
components
 HP NonStop *364*
 UNIX
 Universal Products for SOA *313*
 Windows
 UEC Client Applications *218*
 Universal Enterprise Controller *190*
 Universal Products *149*
 Universal Products for SOA *247*
 z/OS *41*
 z/OS USS *132*
configuration
 HFS *93*
 zFS *93*
configuring
 databases for z/OS *92*
 for z/OS security *115*
 z/OS *81*
cumulative PTF maintenance distribution file
 (z/OS) *77*
customizing
 HP NonStop *370*
 OS/400 *352*
 UNIX *300*
 z/OS *97*
 z/OS USS *140*
customizing for national language (OS/400)

D

data sets

z/OS

non-SMP/E inventory 130

SMP/E inventory 129

space requirements 42

user profiles access 84

z/OS USS

SMP/E inventory 142

space requirements 133

databases for z/OS

configuring 92

default

OS/400 library names 328

deployment of Universal Products for SOA

(UNIX)

Distributed Server Deployment (DSD) 318

options 316

Single-Server Deployment (SSD) 316

detecting completion of silent install

Windows

UEC Client Applications 241

Universal Enterprise Controller 210

Universal Products 173

determining if Windows Installer will be installed

or upgraded 146

digital certificate (RACF) set-up for Universal

Broker for z/OS 113

directories

parent

Universal Products for SOA 325

product

Universal Products for SOA 325

disk space requirements

HP NonStop 365

OS/400 330

UNIX

Universal Products 275

Universal Products for SOA 314

Windows

UEC Client Applications 219

Universal Enterprise Controller 191

Universal Products 150

spool directory 151

Universal Products for SOA 248

dispatch authority

z/OS 127

Distributed Server Deployment (DSD) of

Universal Products for SOA (UNIX) 318

distribution file

Itanium 368

NSK S-series 367

E

Express/OS 45

transferring a cumulative PTF maintenance

file to z/OS 78

transferring z/OS installation files 49

F

false network time-out errors

z/OS 127

file inventory lists

HP NonStop 374

UNIX

Universal Products 307

Universal Products for SOA 325

Windows

UEC Client Applications 245

Universal Enterprise Controller 215

Universal Products 181

Universal Products for SOA 269

files

cumulative PTF maintenance distribution file

(z/OS) 77

installation files

Windows

Universal Products 152

z/OS 46

z/OS USS 46, 136

product distribution file

HP NonStop 367

OS/400 334

UNIX

Universal Products (AIX) 283

Universal Products (generic) 298

Universal Products (HP-UX) 286

Universal Products (Linux) 293

Universal Products (Solaris) 289, 367

Universal Products for SOA (AIX) 321

Universal Products for SOA (Linux)

323

Windows

UEC Client Applications 144

Universal Enterprise Controller 144

Universal Products 144

- Universal Products for SOA *144*
 - z/OS *45*
 - z/OS USS *136*
 - product distribution files *37*
 - transferring z/OS installation files *57*
 - Express/OS *49*
 - manually *50*
- UNIX
 - Universal Products for SOA
 - backing up before installation *319*
- Windows
 - UEC Client Applications
 - installation files *220*
 - un-installed files *233*
 - Universal Enterprise Controller
 - installation files *192*
 - un-installed files *203*
 - Universal Products
 - un-installed files *164*
 - Universal Products for SOA
 - backing up before installation *249*
 - installation files *251*
 - system files *271*
- z/OS UNIX
 - file system *92*

H

- HFS configuration *93*
- HP NonStop
 - customizing *370*
 - disk space requirements *365*
 - file inventory lists *374*
 - installation requirements *365*
 - installing *369*
 - NSK S-series product distribution file *368*
 - product compatibility *364*
 - product distribution file *367*
 - S-series *367*
 - Universal Products components *364*
- HP NonStop (Integrity) *367*
- HP-UX
 - installing patches *286*

I

- IBM S/390 and zSeries systems
 - Linux *293*
- ICSF *112*
- installation files
 - Windows

- UEC Client Applications *220*
- Universal Enterprise Controller *192*
- Universal Products *152*
- Universal Products for SOA *251*
 - z/OS *46*
 - z/OS USS *46, 136*
- installation paths
 - Windows *144*
- installation pre-requisite for z/OS USS *133*
- installation requirements
 - HP NonStop *365*
 - OS/400 *330*
 - UNIX
 - Universal Products *275*
 - directories and files *275*
 - Universal Products for SOA *314*
 - Windows
 - UEC Client Applications *219*
 - Universal Enterprise Controller *191*
 - Universal Products *150*
 - Universal Products for SOA *248*
 - z/OS *42*
 - z/OS USS *133*
- installation upgrades
 - UNIX *279*
 - z/OS *44*
- installing
 - HP NonStop *369*
 - HP-UX patches *286*
 - OS/400 *338*
 - new installation - custom *340*
 - new installation - default *339*
 - upgrade installation for maintenance release *345*
 - upgrade installation for new release *343*
 - UNIX
 - Universal Products (AIX) *283*
 - Universal Products (generic) *298*
 - Universal Products (HP-UX) *286*
 - Universal Products (Linux) *293*
 - Universal Products (Solaris) *289*
 - Universal Products for SOA (AIX) *321*
 - Universal Products for SOA (Linux) *323*
 - Windows
 - UEC Client Applications
 - 64-bit Edition *244*
 - via command line interface *234*
 - via graphical interface *220*
 - Universal Enterprise Controller
 - via command line interface *204*

- via graphical interface *192*
- Windows 2003, 64-bit edition *212*
- Universal Products
 - 64-bit Edition *177, 212*
 - via command line interface *165*
 - via graphical interface *152*
- Universal Products for SOA
 - 64-bit Windows editions *268*
 - via command line interface *260*
 - via graphical interface *251*
- Windows Installer *146*
 - separate from Universal Products *148*
- z/OS *39*
 - 3.1.0 upgrade *74*
 - 3.1.1 upgrade *71*
 - 3.2.0 upgrade *69*
 - newly allocated SMP/E CSI *66*
- z/OS USS *137*
 - 3.2.0 upgrade *139*
 - newly allocated SMP/E CSI *137*
- Integrated Cryptographic Service Facility *112*
- Integrity (HP NonStop) *367*
- Intel-based systems
 - Linux *293*
 - Solaris *289*
- Itanium
 - distribution file *368*
- Itanium systems
 - Linux *293*

L

- libraries
 - OS/400 *359*
- libraries and logs (OS/400) *331*
- library authorization (z/OS) *87*
- library name defaults
 - OS/400 *328*
- library names
 - OS/400 *328*
- licensing
 - UNIX
 - Universal Products for SOA *315*
- listing information
 - UNIX
 - Universal Products (AIX) *285*
 - Universal Products (HP-UX) *288*
 - Universal Products (Linux) *296*
 - Universal Products (Solaris) *292*

- Universal Products for SOA (AIX) *322*
- Universal Products for SOA (Linux) *324*
- local Broker requirement *328*
- log directory
 - UNIX *276*

M

- modifying
 - Windows
 - UEC Client Applications
 - via command line interface *242*
 - via graphical interface *227*
 - Universal Enterprise Controller
 - via command line interface *211*
 - via graphical interface *197*
 - Universal Products
 - via command line interface *174*
 - via graphical interface *157*
 - working folder for Universal Server components (64-bit edition) *177*
 - Universal Products for SOA
 - via command line interface *265*
 - via graphical interface *255*
- msiexec.exe (Windows Installer) *146*

N

- naming conventions
 - OS/400 *328*
- national language customization
 - OS/400 *358*
- non-SMP/E data sets (z/OS) *130*
- NSK S-series
 - distribution file *367*

O

- object inventory lists (OS/400) *359*
- object names (OS/400) *328*
- OS/400
 - command reference library *331*
 - customizing *352*
 - default library names *328*
 - disk space requirements *330*
 - installation requirements *330*
 - installing
 - new installation - custom *340*
 - new installation - default *339*
 - upgrade installation for maintenance

- release 345
- upgrade installation for new release 343
- libraries 359
- libraries and logs 331
- library names 328
- multiple installation implementation 352
- naming conventions 328
- national language customization 358
- object inventory lists 359
- object names 328
- pre-production tasks 344
- product compatibility 329
- product distribution file 334, 335
- product removal 348
- re-installation
 - different libraries 342
 - same libraries 341
- re-installing same release 341
- reset QCCSID value 358
- Spool Library 333
- Trace Library 332
- transferring 336
 - full release 336
 - maintenance release 337
- transferring the product package 336
- UCHGRS program 346
- Universal Broker Job Log 331
- Universal Products components 329
- user profile 330
- OS/400 shell 336

P

- PAM Customization (UNIX - Universal Products for Linux) 297
- performance guidelines
 - z/OS 126
 - Universal Broker 128
 - Universal Products managers 127
 - Universal Products servers 128
- per-machine installation
 - Windows
 - UEC Client Applications 222
- per-user installation
 - Windows
 - UEC Client Applications 222
- pre-installation
 - UNIX
 - Universal Products for SOA 319
 - Windows

- Universal Products 144
- Universal Products for SOA 249
- pre-production tasks
 - OS/400 344
- product compatibility
 - HP NonStop 364
 - OS/400 329
 - UNIX 274
 - Windows
 - Universal Enterprise Controller 190
 - Universal Products 149
 - z/OS 41
 - z/OS USS 132
- product distribution file
 - HP NonStop 367
 - HP NonStop NSK S-series 368
 - OS/400 334, 335
 - UNIX
 - Universal Products (AIX) 283
 - Universal Products (generic) 298
 - Universal Products (HP-UX) 286
 - Universal Products (Linux) 293
 - Universal Products (Solaris) 289, 367
 - Universal Products for SOA (AIX) 321
 - Universal Products for SOA (Linux) 323
 - Windows
 - UEC Client Applications 144
 - Universal Enterprise Controller 144
 - Universal Products 144
 - Universal Products for SOA 144
 - z/OS 45
 - z/OS USS 136
- product distribution files
 - downloading 37
- product licenses
 - Windows
 - Universal Products 175
- product removal
 - OS/400 348
- program authorization (z/OS) 87
- PTF HOLD 78

R

- RACF class (Universal Products for z/OS) 115
- re-installing same release
 - OS/400 341
- remote access profile
 - Universal Command for z/OS 118
 - Universal Control for z/OS 122

Universal Event Monitor for z/OS 124
removing
UNIX
 Universal Products (AIX) 285
 Universal Products (generic) 299
 Universal Products (HP-UX) 288
 Universal Products (Linux) 296
 Universal Products (Solaris) 292
 Universal Products for SOA (AIX) 322
 Universal Products for SOA (Linux) 324

Windows

Enterprise Controller Client Applications
 components
 via graphical interface 227

UEC Client Applications

 via command line interface 243
 via graphical interface 232

UEC Client Applications components

 via command line interface 242
 via graphical interface 227

Universal Enterprise Controller

 via command line interface 211
 via graphical interface 202

Universal Enterprise Controller components

 via command line interface 211
 via graphical interface 197

Universal Products

 via command line interface 174
 via graphical interface 163

Universal Products components

 via command line interface 174
 via graphical interface 157

Universal Products for SOA

 via command line interface 265
 via graphical interface 258

repairing

Windows

UEC Client Applications

 via command line interface 242
 via graphical interface 231

Universal Enterprise Controller

 via command line interface 211
 via graphical interface 201

Universal Products

 via command line interface 174
 via graphical interface 161

Universal Products for SOA

 via graphical interface 255

requirements

UNIX

 Universal Products for SOA
 installation 314

z/OS

 data set space 42

 z/OS installation 42

z/OS USS

 data set space 133

 resetting OS/400 QCCSID value 358

S

security

 configuring for z/OS 115

 Universal Command for z/OS 118

Windows

Universal Products

 pool directory 151

 security environment (z/OS) 83

security profile

 Universal Command for z/OS 120

 Universal Control for z/OS 123

 Universal Event Monitor for z/OS 125

 setup.exe 166, 205, 235

silent install

Windows

 UEC Client Applications 238

 detecting completion 241

 using distribution file 241

 Universal Enterprise Controller

 detecting completion 210

 using distribution file 210

 Universal Products 170, 172

 detecting completion 173

 using distribution file 173

 Single-Server Deployment (SSD) of Universal
 Products for SOA (UNIX) 316

 SMF exit routine (z/OS) 90

SMP/E back-up

 z/OS 43

SMP/E data sets

 z/OS 129

 z/OS USS 142

SMP/E FMIDs

 z/OS 43

 z/OS USS 134

 SMP/E installation (z/OS) 65

SPARC-based systems

 Solaris 289

pool directory

- UNIX 277
- Windows
 - Universal Products 150
 - disk space requirements 151
 - security 151
- spool library (OS/400) 333
- S-series (HP NonStop) 367
- standard I/O access profile
 - Universal Command for z/OS 119
- started procedure (z/OS) 82
- started tasks associated with profiles (z/OS) 85
- system files
 - Windows
 - Universal Products for SOA 271
- System SSL 111

T

- TCP/IP configuration (z/OS) 109
- time zone (TZ) environment variable (z/OS) 108
- trace directory (Universal Products for UNIX) 276
- trace library (OS/400) 332
- transferring
 - cumulative PTF maintenance file to z/OS
 - Express/OS 78
 - manually 79
 - OS/400 336
 - z/OS installation files 57
 - Express/OS 49
 - manually 50

U

- UCHGRSL (Change Release Tag) program 346
- un-installed files
 - Windows
 - UEC Client Applications 233
 - Universal Enterprise Controller 203
 - Universal Products 164
- un-installing previous version of z/OS USS 135
- Universal Command for z/OS
 - security profiles 120
- Universal Enterprise Controller
 - Windows
 - product compatibility 190
- Universal Products (Linux)
 - installing 293
- Universal Products for SOA
 - parent directories 325
 - product directories 325
- UNIX

- installation requirements 314
- UNIX
 - command reference directory 275
 - customizing 300
 - installation upgrades 279
 - installing (generic installation) 298
 - log directory 276
 - product compatibility 274
 - spool directory 277
 - Universal Products
 - disk space requirements 275
 - file inventory lists 307
 - installation requirements 275
 - directories and files 275
 - trace directory 276
 - Universal Products (AIX)
 - installing 283
 - listing information 285
 - package filesets 284
 - product distribution file 283
 - removing 285
 - Universal Products (generic)
 - environment variables 299
 - product distribution file 298
 - removing 299
 - Universal Products (HP-UX)
 - installing 286
 - listing information 288
 - package contents 287
 - product distribution file 286
 - removing 288
 - Universal Products (Linux)
 - IBM S/390 and zSeries systems 293
 - Intel-based systems 293
 - Itanium systems 293
 - listing information 296
 - PAM Customization 297
 - product distribution file 293
 - removing 296
 - Universal Products (Solaris)
 - installing 289
 - Intel-based systems 289
 - listing information 292
 - package contents 291
 - product distribution file 289, 367
 - removing 292
 - SPARC-based systems 289
 - Universal Products components 274
 - Universal Products for SOA
 - components 313

- deployment 316
 - disk space requirements 314
 - file inventory list 325
 - installation 320
 - installation requirements 314
 - licensing 315
 - Universal Products for SOA (AIX)
 - installation 321
 - listing information 322
 - product distribution file 321
 - removing 322
 - Universal Products for SOA (Linux)
 - installation 323
 - listing information 324
 - product distribution file 323
 - removing 324
 - unpacking
 - z/OS data sets 58
 - unpacking z/OS data sets
 - sample JCL 58
 - upgrading
 - Windows Installer 146
 - z/OS
 - from Universal Command 2.1.0, 1.2.0, or 1.1.0 65
 - from Universal Products 2.2.0 65
 - user profile
 - OS/400 330
 - user profiles (z/OS)
 - data sets access 84
- W**
- Windows
 - default installation paths 144
 - product compatibility
 - Universal Enterprise Controller 190
 - product distribution files 144
 - UEC Client Applications
 - adding components
 - via command line interface 242
 - via graphical interface 227
 - components 218
 - detecting completion of silent install 241
 - disk space requirements 219
 - file inventory lists 245
 - installation requirements 219
 - installing
 - via command line interface 234
 - via graphical interface 220
 - modifying
 - via command line interface 242
 - via graphical interface 227
 - per-machine installation 222
 - per-user installation 222
 - removing
 - via command line interface 243
 - via graphical interface 232
 - removing components
 - via command line interface 242
 - via graphical interface 227
 - repairing
 - via command line interface 242
 - via graphical interface 231
 - silent install 238
 - using distribution file 241
 - un-installed files 233
 - Windows Installer command line
 - parameters 236
 - Windows Installer command line syntax 239
 - Windows Installer installation commands 240
 - Universal Enterprise Controller
 - adding components
 - via command line interface 211
 - via graphical interface 197
 - components 190
 - detecting completion of silent install 210
 - disk space requirements 191
 - file inventory lists 215
 - installation requirements 191
 - installing
 - via command line interface 204
 - via graphical interface 192
 - Windows 2003, 64-bit edition 212
 - modifying
 - via command line interface 211
 - via graphical interface 197
 - removing
 - via command line interface 211
 - via graphical interface 202
 - removing components
 - via command line interface 211
 - via graphical interface 197
 - repairing
 - via command line interface 211
 - via graphical interface 201
 - silent install
 - using distribution file 210

- un-installed files 203
 - Windows Installer
 - installation commands 209
 - Windows Installer command line
 - parameters 206
 - Windows Installer command line syntax 208
 - Universal Products
 - adding components
 - via command line interface 174
 - via graphical interface 157
 - components 149
 - detecting completion of silent install 173
 - disk space requirements 150
 - spool directory 151
 - file inventory lists 181
 - installation requirements 150
 - installing
 - via command line interface 165
 - via graphical interface 152
 - installing on Windows Server 2003, 64-bit Edition 177
 - modifying
 - via command line interface 174
 - via graphical interface 157
 - pre-installation 144
 - product compatibility 149
 - product licenses 175
 - removing
 - via command line interface 174
 - via graphical interface 163
 - removing components
 - via command line interface 174
 - via graphical interface 157
 - repairing
 - via command line interface 174
 - via graphical interface 161
 - silent install 170, 172
 - using distribution file 173
 - spool directory 150
 - security 151
 - un-installed files 164
 - Windows Installer command line
 - parameters 167
 - Windows Installer command line syntax 171
 - Windows Installer installation commands 172
 - Universal Products for SOA
 - disk space requirements 248
 - file inventory 269
 - installation requirements 248
 - installing
 - 64-bit Windows editions 268
 - via command line interface 260
 - via graphical interface 251
 - modifying
 - via command line interface 265
 - via graphical interface 255
 - products and components 247
 - removing
 - via command line interface 265
 - via graphical interface 258
 - repairing
 - via graphical interface 255
 - Windows Installer 146
 - installing separate from Universal Products 148
 - interfaces 152
 - UEC Client Applications
 - command line parameters 236
 - command line syntax 239
 - installation commands 240
 - Universal Enterprise Controller
 - command line parameters 206
 - command line syntax 208
 - installation commands 209
 - Universal Products
 - command line parameters 167
 - command line syntax 171
 - installation commands 172
 - Windows Installer (msiexec.exe) 146
 - working folder
 - Windows
 - Universal Products
 - Universal Server components (64-bit edition) 177
- ## Z
- z/OS
 - authorization for libraries and programs 87
 - configuring 81
 - configuring security 115
 - data set inventory
 - non-SMP/E 130
 - SMP/E 129
 - data set space requirements 42
 - data sets
 - allocating 52

- sample JCL 52
 - unpacking 58
- dispatch authority 127
- false network time-out errors 127
- files
 - cumulative PTF maintenance distribution file 77
 - installation files 46
 - product distribution file 45
- installation files
 - transferring 57
- installation requirements 42
- installation upgrade for 3.1.0 74
- installation upgrade for 3.1.1 71
- installation upgrade for 3.2.0 69
- installation upgrades 44
- new installation 66
- performance guidelines 126
 - Universal Broker 128
 - Universal Enterprise Controller 128
 - Universal Products managers 127, 128
- product compatibility 41
- security environment 83
- SMF exit routine 90
- SMP/E back-up 43
- SMP/E FMIDs 43
- SMP/E installation 65
- started procedure 82
- started tasks 81
- System SSL 111
- TCP/IP configuration 109
- time zone (TZ) environment variable 108
- transferring installation files
 - Express/OS 49
 - manually 50
- Universal Broker
 - digital certificate set-up 113
- Universal Command
 - remote access profile 118
 - security 118
 - standard I/O access profile 119
- Universal Control
 - remote access profile 122
 - security profile 123
- Universal Event Monitor
 - remote access profile 124
 - security profile 125
- Universal Products
 - ACF2 class 117
 - RACF class 115
- Universal Products components 41
- unpacking data sets
 - sample JCL 58
- upgrading from Universal Command 2.1.0, 1.2.0, or 1.1.0 65
- upgrading from Universal Products 2.2.0 65
- z/OS Language Environment 126
- z/OS UNIX
 - directories
 - z/OS USS 142
 - file system 92
- z/OS USS
 - customization 140
 - files
 - installation files 46
 - installation 137
 - installation pre-requisite 133
 - installation requirements 133
 - installation upgrade for 3.2.0 139
 - new installation 137
 - product compatibility 132
 - product distribution file 136
 - requirements
 - data set space 133
 - SMP/E data sets 142
 - SMP/E FMIDs 134
 - un-installing previous version 135
 - Universal Products components 132
 - z/OS UNIX directories 142
- zFS configuration 93



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

