



**Workload Automation 5.1.0**

**Installation and Administration Guide**

© 2013 by Stonebranch, Inc. All Rights Reserved.

1. Workload Automation 5.1.0 Installation and Administration Guide	6
1.1 zOS Installation	9
1.1.1 zOS Installation - Overview	10
1.1.2 zOS Installation - Installation Package	11
1.1.3 zOS Installation - Installation Requirements	12
1.1.4 zOS Installation - Installation Upgrades	13
1.1.5 zOS Installation - Distribution File	15
1.1.6 zOS Installation - Transferring Installation Files to zOS	18
1.1.7 zOS Installation - SMPE Installation	30
1.1.7.1 zOS Installation - SMPE Installation Overview	31
1.1.7.2 zOS Installation - New Install, New CSI	33
1.1.7.3 zOS Installation - Stonebranch Solutions 4.x Upgrade, Existing CSI	35
1.1.7.4 zOS Installation - Universal Products 3.2.0 Upgrade, Existing CSI	36
1.1.8 zOS Installation - Cumulative PTF Maintenance	37
1.1.9 zOS Installation - Configuration	40
1.1.9.1 zOS Configuration - Overview	41
1.1.9.2 zOS Configuration - Started Tasks	42
1.1.9.3 zOS Configuration - Load Library	48
1.1.9.4 zOS Configuration - SMF Exits	51
1.1.9.5 zOS Configuration - JES SYSOUT Processing	55
1.1.9.6 Workload Automation 5 Database Configuration	57
1.1.10 zOS Installation - Customization	61
1.1.11 zOS Installation - Licensing	67
1.1.12 zOS Installation - Time Zone Environment Variable	69
1.1.13 zOS Installation - TCP/IP Configuration	70
1.1.14 zOS Installation - SAP RFC DLL	72
1.1.15 zOS Installation - Configuration of zOS System SSL	74
1.1.15.1 Configuration of zOS System SSL - Overview	75
1.1.15.2 Integrated Cryptographic Service Facility (ICSF)	76
1.1.15.3 Universal Broker Digital Certificate (RACF) Set-up	77
1.1.16 zOS Installation - Configuration of Security	79
1.1.16.1 zOS Installation - Configuration of Security Overview	80
1.1.16.2 Configuration of Security - RACF Class	81
1.1.16.3 Configuration of Security - ACF2 Class	83
1.1.16.4 Configuration of Security - Universal Command Security	84
1.1.16.5 Configuration of Security - Universal Control Security	87
1.1.16.6 Configuration of Security - Universal Event Monitor Security	89
1.1.17 zOS Installation - Performance Guidelines	91
1.1.18 zOS Installation - Converting STC User Profiles to a Non-Zero UID	93
1.1.19 zOS Installation - Data Set Inventory	96
1.2 zOS USS Installation	99
1.2.1 zOS USS Installation - Overview	100
1.2.2 zOS USS Installation - Components	101
1.2.3 zOS USS Installation - Installation Requirements	102
1.2.4 zOS USS Installation - Installation Upgrades	104
1.2.5 zOS USS Installation - Distribution File	105
1.2.6 zOS USS Installation - SMPE Installation	106
1.2.6.1 zOS USS SMPE Installation - Overview	107
1.2.6.2 zOS USS Installation - New Install, New CSI	108
1.2.6.3 zOS USS Installation - 4.x Upgrade, Existing CSI	109
1.2.6.4 zOS USS Installation - 3.2.0 Upgrade, Existing CSI	110
1.2.7 zOS USS Installation - Customization	111
1.2.8 zOS USS Installation - Licensing	113
1.2.9 zOS USS Installation - Data Set Inventory	115
1.3 Windows Installation	116
1.3.1 Windows Installation - Overview	117
1.3.2 Windows Installation - Licensing	121
1.3.3 Workload Automation 5 for Windows	124
1.3.3.1 Workload Automation 5 for Windows - Overview	125
1.3.3.2 Workload Automation 5 for Windows - Installation Package	126
1.3.3.3 Workload Automation 5 for Windows - Installation Requirements	127
1.3.3.4 Workload Automation 5 for Windows - Installation Procedures	130
1.3.3.4.1 Workload Automation 5 for Windows - Installation Procedures Overview	131
1.3.3.4.2 Installing Workload Automation 5 via the Graphical Interface	132
1.3.3.4.3 Modifying a Workload Automation 5 Installation via the Graphical Interface	141
1.3.3.4.4 Installing Workload Automation 5 via the Command Line Interface	149
1.3.3.4.5 Modifying a Workload Automation 5 Installation via the Command Line Interface	157
1.3.3.4.6 Migrating between 32- and 64-bit Workload Automation 5 for Windows Installs	159
1.3.3.4.7 Migrating an Opwise Agent to UAG for Workload Automation 5 for Windows	164
1.3.3.5 32-Bit Workload Automation 5 for Windows on 64-Bit Windows Systems	166
1.3.3.6 Workload Automation 5 for Windows - File Inventory Lists	171
1.3.4 Universal Enterprise Controller for Windows	182
1.3.4.1 Universal Enterprise Controller for Windows - Overview	183
1.3.4.2 Universal Enterprise Controller for Windows - Package	184
1.3.4.3 Universal Enterprise Controller for Windows - Installation Requirements	185

1.3.4.4 Universal Enterprise Controller for Windows - Installation Procedures	186
1.3.4.4.1 Universal Enterprise Controller for Windows - Installation Procedures Overview	187
1.3.4.4.2 Installing UEC via the Graphical Interface	188
1.3.4.4.3 Modifying a UEC Installation via the Graphical Interface	193
1.3.4.4.4 Installing UEC via the Command Line Interface	201
1.3.4.4.5 Modifying a UEC Installation via the Command Line Interface	206
1.3.4.5 Universal Enterprise Controller for Windows - 64-Bit Windows Editions	207
1.3.4.6 Universal Enterprise Controller for Windows - Database Configuration	209
1.3.4.7 Universal Enterprise Controller for Windows - File Inventory Lists	210
1.3.5 UEC Client Applications	213
1.3.5.1 UEC Client Applications - Overview	214
1.3.5.2 UEC Client Applications - Package	215
1.3.5.3 UEC Client Applications - Installation Requirements	216
1.3.5.4 UEC Client Applications - Installation Procedures	217
1.3.5.4.1 UEC Client Applications - Installation Procedures Overview	218
1.3.5.4.2 Installing UEC Client Applications via the Graphical Interface	219
1.3.5.4.3 Modifying a UEC Client Applications Installation via the Graphical Interface	226
1.3.5.4.4 Installing UEC Client Applications via the Command Line Interface	236
1.3.5.4.5 Modifying a UEC Client Applications Installation via the Command Line Interface	242
1.3.5.5 UEC Client Applications - 64-Bit Windows Editions	244
1.3.5.6 UEC Client Applications - File Inventory Lists	245
1.3.6 Workload Automation 5 for SOA for Windows	247
1.3.6.1 Workload Automation 5 for SOA for Windows - Overview	248
1.3.6.2 Workload Automation 5 for SOA for Windows - Package	249
1.3.6.3 Workload Automation 5 for SOA for Windows - Installation Requirements	250
1.3.6.4 Workload Automation 5 for SOA for Windows - Pre-Installation - Upgrade Backups	251
1.3.6.5 Workload Automation 5 for SOA for Windows - Installation Procedures	252
1.3.6.5.1 Workload Automation 5 for SOA for Windows - Installation Procedures Overview	253
1.3.6.5.2 Installing SOA for Windows via the Graphical Interface	254
1.3.6.5.3 Modifying a SOA for Windows Installation via the Graphical Interface	258
1.3.6.5.4 Installing SOA for Windows via the Command Line Interface	264
1.3.6.5.5 Modifying a SOA for Windows Installation via the Command Line Interface	268
1.3.6.6 Workload Automation 5 for SOA for Windows - Configuring and Starting UAC Server	269
1.3.6.7 Workload Automation 5 for SOA for Windows - 64-Bit Windows Editions	271
1.3.6.8 Workload Automation 5 for SOA for Windows - File Inventory Lists	272
1.4 UNIX Installation	275
1.4.1 UNIX Installation - Overview	276
1.4.2 UNIX Installation - Licensing	277
1.4.3 Workload Automation 5 for UNIX	279
1.4.3.1 Workload Automation 5 for UNIX - Installation Package	280
1.4.3.2 Workload Automation 5 for UNIX - Installation Requirements	281
1.4.3.3 Workload Automation 5 for UNIX - Installation Upgrades	285
1.4.3.4 Workload Automation 5 for UNIX - Distribution File	286
1.4.3.5 Workload Automation 5 for UNIX - Installation Procedures	287
1.4.3.5.1 Workload Automation 5 for UNIX - Installation Procedures Overview	288
1.4.3.5.2 Workload Automation 5 for AIX Installation	289
1.4.3.5.3 Workload Automation 5 for HP-UX Installation	294
1.4.3.5.4 Workload Automation 5 for Solaris Installation	299
1.4.3.5.5 Workload Automation 5 for Linux Installation	305
1.4.3.5.6 Workload Automation 5 for UNIX Installation (Generic)	312
1.4.3.5.7 Migrating an Opswise Agent to JAG for Workload Automation 5 for UNIX	315
1.4.3.6 Workload Automation 5 for UNIX - Customization	317
1.4.3.7 Workload Automation 5 for UNIX - File Inventory Lists	321
1.4.4 Workload Automation 5 for SOA for UNIX	329
1.4.4.1 Workload Automation 5 for SOA for UNIX - Installation Overview	330
1.4.4.2 Workload Automation 5 for SOA for UNIX - Deployment Options	331
1.4.4.3 Workload Automation 5 for SOA for UNIX - Pre-Installation-Upgrade Backups	334
1.4.4.4 Workload Automation 5 for SOA for UNIX - Distribution File	335
1.4.4.5 Workload Automation 5 for SOA for UNIX - Installation Procedures	336
1.4.4.5.1 Workload Automation 5 for SOA for UNIX - Installation Procedures Overview	337
1.4.4.5.2 Workload Automation 5 for SOA for AIX Installation	338
1.4.4.5.3 Workload Automation 5 for SOA for Linux Installation	341
1.4.4.6 Workload Automation 5 SOA for UNIX Installation - File Inventory Lists	344
1.5 IBM i Installation	347
1.5.1 IBM i Installation - Overview	348
1.5.2 IBM i Installation - Installation Package	349
1.5.3 IBM i Installation - Installation Requirements	351
1.5.4 IBM i Installation - Distribution File	354
1.5.5 IBM i Installation - Transferring to IBM i	356
1.5.6 IBM i Installation - Installation Procedures	359
1.5.6.1 IBM i Installation - Installation Procedures Overview	360
1.5.6.2 IBM i Installation - New Installation (Default)	361
1.5.6.3 IBM i Installation - New Installation (Custom)	362
1.5.6.4 IBM i Installation - Re-Installation of Same Release	363
1.5.6.5 IBM i Installation - Upgrade Installation to New Release	364

1.5.6.6 IBM i Installation - Propagating New Release to Additional Systems	366
1.5.6.7 IBM i Installation - Upgrade Installation for Maintenance Release	368
1.5.6.8 IBM i Installation - UCHGRS (Change Release Tag) Program	369
1.5.6.9 IBM i Installation - Product Removal	371
1.5.7 IBM i Installation - Customization	376
1.5.8 IBM i Installation - Licensing	380
1.5.9 IBM i Installation - Object Inventory Lists	382
1.6 HP NonStop Installation	389
1.6.1 HP NonStop Installation - Overview	390
1.6.2 HP NonStop Installation - HP NonStop Components	391
1.6.3 HP NonStop Installation - Installation Requirements	392
1.6.4 HP NonStop Installation - Distribution File	394
1.6.5 HP NonStop Installation - Installation Procedures	396
1.6.6 HP NonStop Installation - Customization	397
1.6.7 HP NonStop Installation - Licensing	400
1.6.8 HP NonStop Installation - File Inventory Lists	401
1.7 Workload Automation 5 Administration	404
1.7.1 Workload Automation 5 Administration - Overview	405
1.7.2 Workload Automation 5 - Databases	406
1.7.2.1 Databases - Overview	407
1.7.2.2 Databases - Component Information Database	408
1.7.2.3 Databases - Universal Event Monitor Databases	410
1.7.2.3.1 Universal Event Monitor Databases - Overview	411
1.7.2.3.2 Universal Event Monitor Event Definition Database	412
1.7.2.3.3 Universal Event Monitor Event Handler Database	413
1.7.2.3.4 Universal Event Monitor Event Spool Database	414
1.7.2.3.5 Controlling Universal Event Monitor Database Access	415
1.7.2.4 Universal Enterprise Controller Databases	416
1.7.2.5 Database Backup and Recovery	418
1.7.2.6 Database Recovery for Universal Broker	419
1.7.2.7 Database Recovery for Universal Enterprise Controller	420
1.7.2.8 Listing Workload Automation 5 Database Records Examples	421
1.7.2.8.1 List Universal Broker Database	422
1.7.2.8.2 List Universal Command Server Database Records	423
1.7.2.8.3 List Universal Broker Detail for a Component	424
1.7.2.8.4 List Standard Out for a Component	425
1.7.2.9 Removing Workload Automation 5 Database Records Examples	426
1.7.2.9.1 Remove Component Records	427
1.7.2.9.2 Remove Component Records - Change Database Directory	428
1.7.3 Workload Automation 5 - Security	429
1.7.3.1 Security Overview	430
1.7.3.2 Security of Workload Automation 5 Components	431
1.7.3.2.1 Security of Workload Automation 5 Components - Overview	432
1.7.3.2.2 Security - Universal Broker	433
1.7.3.2.3 Security - Universal Command Manager	437
1.7.3.2.4 Security - Universal Command Server	438
1.7.3.2.5 Security - Universal Data Mover Manager	440
1.7.3.2.6 Security - Universal Data Mover Server	441
1.7.3.2.7 Security - Universal Event Monitor Manager	443
1.7.3.2.8 Security - Universal Event Monitor Server	444
1.7.3.2.9 Security - Universal Control Manager	446
1.7.3.2.10 Security - Universal Control Server	447
1.7.3.2.11 Security - Universal Event Log Dump	448
1.7.3.2.12 Security - Universal Spool List	449
1.7.3.2.13 Security - Universal Spool Remove	450
1.7.3.3 Security - Universal Access Control List (UACL)	451
1.7.3.3.1 Universal Access Control List - Overview	452
1.7.3.3.2 Universal Access Control List - UACL Entries	453
1.7.3.4 Security - Universal Access Control List (UACL) Examples	463
1.7.3.4.1 UACL Examples - Universal Broker for zOS	464
1.7.3.4.2 UACL Examples - Universal Command Server for zOS	465
1.7.3.4.3 UACL Examples - Universal Data Mover Server for zOS	466
1.7.3.4.4 UACL Examples - Universal Control Server for zOS	467
1.7.3.4.5 UACL Examples - Universal Broker for Windows	468
1.7.3.4.6 UACL Examples - Universal Command Server for Windows	469
1.7.3.4.7 UACL Examples - Universal Data Mover Server for Windows	470
1.7.3.4.8 UACL Examples - Universal Control Server for Windows	471
1.7.3.4.9 UACL Examples - Universal Event Monitor Server for Windows	472
1.7.3.4.10 UACL Examples - Universal Broker for UNIX	473
1.7.3.4.11 UACL Examples - Universal Command Server for UNIX	474
1.7.3.4.12 UACL Examples - Universal Data Mover Server for UNIX	475
1.7.3.4.13 UACL Examples - Universal Control Server for UNIX	476
1.7.3.4.14 UACL Examples - Universal Event Monitor Server for UNIX	477
1.7.3.4.15 UACL Examples - Universal Broker for IBM i	478
1.7.3.4.16 UACL Examples - Universal Command Server for IBM i	479

1.7.3.4.17 UACL Examples - Universal Data Mover Server for IBM i .....	480
1.7.3.4.18 UACL Examples - Universal Control Server for IBM i .....	481
1.7.3.4.19 UACL Examples - Universal Broker for HP NonStop .....	482
1.7.3.4.20 UACL Examples - Universal Command Server for HP NonStop .....	483
1.7.3.4.21 UACL Examples - Universal Control Server for HP NonStop .....	484
1.7.3.5 Security - X.509 Certificates .....	485
1.7.3.5.1 X.509 Certificates - Overview .....	486
1.7.3.5.2 X.509 Certificates - Sample X.509 Certificate .....	487
1.7.3.5.3 X.509 Certificates - SSL Peer Authentication .....	489
1.7.3.6 Security - Creating Certificates Examples .....	492
1.7.3.6.1 Creating a Certificate .....	493
1.7.3.6.2 Creating a Certificate Authority (CA) Certificate .....	494
1.8 Licenses and Copyrights .....	495

# Workload Automation 5.1.0 Installation and Administration Guide

- Installation Packages
- Installation Information
- Installation Requirements
  - Platform Requirements
  - Network Requirements
  - Ports Configuration
  - Additional Requirements
- Product Distribution
  - Opwise Automation Center
- Product Licensing
- Additional Information

## Installation Packages

These pages provide information and instructions for the installation of Workload Automation 5 packages on one or more operating systems.

The following table identifies the different operating systems on which Workload Automation 5 can be installed, and the packages that can be installed on each system.

Operating System	Workload Automation 5	Universal Enterprise Controller	UEC Client Applications	Workload Automation 5 for SOA
z/OS <sup>1</sup>	✓			
z/OS USS	✓			
Windows	✓	✓	✓	✓
UNIX – AIX	✓			✓
UNIX – HP-UX	✓			
UNIX – Linux	✓			✓
UNIX – Solaris	✓			
UNIX – (generic)	✓			
IBM i	✓			
HP NonStop <sup>2</sup>	✓			

<sup>1</sup> Workload Automation 5 for z/OS contains Universal Enterprise Controller.

<sup>2</sup> Universal Command 2.1.1 is installed on HP NonStop.

## Installation Information

The information in these pages 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

## Installation Requirements

Installation requirements, including system requirements, for each platform are located on the Installation Requirements page specific to that platform:

- [z/OS Installation Requirements](#)
- [Windows Installation Requirements](#)
- [UNIX Installation Requirements](#)
- [IBM i Installation Requirements](#)
- [HP NonStop Installation Requirements](#)

Requirements relevant to all platforms are shown below.

## Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opwise Automation Center 5.1.1 and Indesca-Infitran 5.1.0](#) page to make sure that your platform is supported before performing an installation.

## Network Requirements

Workload Automation components run on z/OS, Windows, UNIX, IBM i, and HP NonStop operating systems. These systems must be connected with a network that supports TCP/IP. For example, the Universal Command Manager running on a z/OS system must be able to establish a TCP/IP socket connection with the Universal Command Server running on a UNIX or Windows system.

TCP/IP provides a set of commands to verify network connectivity between systems. For example, the ping command can determine if an IP connection is possible between two systems. However, the ping command may not work in all network environments. In addition, the ping command may work, but a firewall may deny all other connection attempts between the two systems. Check with your local network administrator to determine the capabilities and topology of your network.

Workload Automation offers configuration parameters that can facilitate connections through network firewalls. Due to the large variety of firewall configurations, all possibilities cannot be discussed in this document. Check with your local network administrator to determine if a firewall is between the computer systems involved.

## Ports Configuration

Ports configured for Workload Automation 5 components and prerequisites cannot be blocked by a firewall.

The following table identifies the default ports, which you can change during installation or configuration:

Component or Prerequisite	Default Port
MySQL	3306
Microsoft SQL Server	1433
Oracle	1521
Universal Broker	7887
Universal Broker (for IPC)	7987
Universal Application Container (local port)	7889
Universal Enterprise Controller	8778
Opwise Controller (Tomcat)	8080
Opwise Message Hub	6776

Opwise Transport	4803
Opwise Inter-Transport Communications*	4804, 4805

\* **NOTE:** These ports currently cannot be changed.

## Additional Requirements

In order to install any of these Workload Automation components, you must be able to write to the directory from which the installation is launched.

## Product Distribution

To install a Workload Automation 5 package, you first must download the corresponding product distribution file for your specific operating system version and hardware platform.

All Workload Automation 5 distribution files are available for download from the Stonebranch [Customer Portal](#).

A customer user name and password – provided by Stonebranch, Inc. – are required to access the Customer Portal.

These pages identify the specific product distribution files to be downloaded for the different Workload Automation 5 packages.

## Opwise Automation Center

Workload Automation 5 packages do not include the Opwise Automation Center components. For information on installing Automation Center, see [Opwise Automation Center Installation and Upgrade](#).

## Product Licensing

A Workload Automation 5 package installation may include products that must be licensed before they can be used.

These pages identify the products that must be licensed, and how the licensing is made, for Workload Automation 5.1.0 on each operating system.

Licenses for Workload Automation 5 are provided by your Stonebranch, Inc. account representative.

## Additional Information

In addition to this Installation Guide, Stonebranch, Inc. provides the following information:

[Workload Automation 5.1.0 System Requirements and Installation Summary](#) identifies network requirements for all Workload Automation 5 installations and, for each operating system, the following information:

- System requirements
- Installation summary
- Customization

[Workload Automation 5.1.0 Installation Quick Start Guides](#) provide the following information:

- System Requirements
- Downloading Workload Automation 5.1.0 Distribution File
- Installing Workload Automation 5.1.0
- Licensing your 5.1.0 Components
- Encrypting User ID and Password for Job Execution
- Executing a Job
- Running a System Query

There is a separate quick start guide for each Indesca and Infitran supported platform.



## **zOS Installation**

Error formatting macro: redirect: java.lang.NullPointerException

## zOS Installation - Overview

- [Introduction](#)
- [Installation Checklist](#)

### Introduction

These pages describe the installation of Stonebranch Inc.'s Workload Automation 5 for z/OS on a z/OS operating system. Unless otherwise specified, all references to Workload Automation for z/OS refer to version 5.1.0.

Workload Automation for z/OS is provided as an easily installed SMP/E package (see [z/OS Installation - Installation Package](#)). After the SMP/E installation steps are complete, z/OS must be configured to meet the product requirements. Lastly, the individual product components must be customized to meet local production requirements.

The z/OS package includes the components for both Workload Automation for z/OS and optional components for Workload Automation for z/OS UNIX System Services (USS). For information on installation of Workload Automation for z/OS USS, see [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 Workload Automation component is running, including manager components. The Broker maintains product configuration data for all components that have a configuration file.

### Installation Checklist

The following installation checklist provides a concise overview of the z/OS installation steps. Each step refers to the page in this installation guide that describe that step in complete detail. All z/OS installation pages should be read to avoid product installation and configuration problems.

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.

<b>Step 1</b>	Download the Workload Automation 5.1.0 for z/OS product distribution file to your Windows workstation. The distribution file is packaged as an Express/OS Windows executable file. See <a href="#">zOS Installation - Distribution File</a> for complete details.
<b>Step 2</b>	Execute the Express/OS package on a Windows workstation that has TCP/IP connectivity to the z/OS system. The Express/OS application will start and prompt you for the information it needs to transfer the z/OS installation libraries to z/OS. Once the transfer is complete, Express/OS will prompt you to submit a z/OS batch job to unpack the libraries, which have been packaged in a TSO XMIT format. Select YES to submit the batch job. The installation JCL is located in the #HLQ.UNV.V5R1M0.INSTALL library (simply referred to as the INSTALL library). The INSTALL library is created on z/OS by the batch job that unpacked the libraries. See <a href="#">zOS Installation - Transferring Installation Files to zOS</a> for complete details.
<b>Step 3</b>	Edit the INSTALL(#SETUP) member to meet local requirements as described in the #SETUP comment prolog. #SETUP creates customized installation JCL members in the INSTALL library. Execute the installation jobs as required for your SMP/E environment. See <a href="#">zOS Installation - SMPE Installation</a> for complete details.
<b>Step 4</b>	Create started task user and group profiles for Universal Broker and Universal Enterprise Controller. See <a href="#">zOS Configuration - Started Tasks</a> for complete details.
<b>Step 5</b>	Define the SUNVLOAD library to APF, Program Control, and LNKLST. See <a href="#">zOS Configuration - Load Library</a> for complete details.
<b>Step 6</b>	Define SMF exits for Universal Command and Universal Automation Center Agent. See <a href="#">zOS Configuration - SMF Exits</a> for complete details.
<b>Step 7</b>	Configure the UNIX System Services file systems used by the Universal Broker and Universal Enterprise Controller. See <a href="#">Workload Automation 5 Database Configuration</a> for complete details.
<b>Step 8</b>	Perform individual component customization, including JCL procedure edits and adding component licenses. See <a href="#">zOS Installation - Customization</a> for complete details.

## zOS Installation - Installation Package

- [Workload Automation for z/OS Package Components](#)
- [Component Compatibility](#)

### Workload Automation for z/OS Package Components

The Workload Automation 5 for z/OS package contains the following Workload Automation for z/OS components:

- Universal Broker 5.1.0
- Universal Automation Center Agent 5.1.0
- Universal Command Manager and Server 5.1.0
- Universal Control Manager and Server 5.1.0
- Universal Data Mover Manager and Server 5.1.0
- Universal Encrypt 5.1.0
- Universal Enterprise Controller 5.1.0
- Universal Event Monitor Manager 5.1.0
- Universal Message to Exit Code Translator 5.1.0
- Universal Query 5.1.0
- Universal Connector 5.1.0
- Universal Certificate 5.1.0
- Opwise Automation Center Command Line Interface 5.1.0



#### Note

For the list of Workload Automation for z/OS USS components included in the z/OS package, see [z/OS USS Installation - Components](#).

### Component Compatibility

The following table identifies the compatibility of Workload Automation 5 for z/OS components with previous component / product versions.

Component	Compatibility
Universal Broker 5.1.0	Stonebranch Solutions / Universal Products releases 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Automation Center Agent 5.1.0	Universal Automation Center Agent is compatible with Automation Center 5.1.0 and above.
Universal Command 5.1.0	Universal Command 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Control 5.1.0	Universal Control 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Data Mover 5.1.0	Universal Data Mover 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Encrypt 5.1.0	Universal Encrypt 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Query 5.1.0	Universal Broker 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Enterprise Controller 5.1.0	Not compatible with previous versions of Universal Enterprise Controller Client Applications.
Universal Event Monitor 5.1.0	Universal Event Monitor 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, and 3.1.0.

The component references pertain to all supported platforms for that version.

## zOS Installation - Installation Requirements

- [System Requirements](#)
- [Platform Requirements](#)
- [Data Set Space Requirements](#)

### System Requirements

Workload Automation 5 for z/OS require the following software releases:

- z/OS 1.10 or above.
- SMP/E 3.5 or above.
- IBM Communication Server for z/OS 1.10 or above.
- IBM Language Environment (LE) for z/OS 1.10 or above.
- Windows workstation capable of establishing a TCP/IP network connection to the z/OS system.
- TSO user ID with an OMVS segment.
- About 1900 cylinders of DASD.
- Two available TCP/IP ports on z/OS.

All Workload Automation 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.

### Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opwise Automation Center 5.1.1](#) and [Indesca-Infitran 5.1.0](#) page to make sure that your platform is supported before performing an installation.

### Data Set Space Requirements

As part of the Workload Automation 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 [z/OS Installation - Data Set Inventory](#).

## zOS Installation - Installation Upgrades

- Workload Automation for z/OS - Installation Upgrades
  - Universal Command 1.2.0
  - Universal Command 2.1.0
  - Universal Products 2.2.0
  - Universal Products 3.1.1
  - Universal Products 3.2.0
  - Universal Products 4.1.0
  - Stonebranch Solutions 4.3.0
  - Workload Automation 5
- Platform Requirements

### Workload Automation for z/OS – Installation Upgrades

This page 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 **SMPPTS** and **SMPPTS** data sets and **SMPPLIB** 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.

#### Universal Products 4.1.0

The Universal Broker and Universal Enterprise Controller STCs can now use user profiles with non-zero UID values instead of UID 0 as required in previous releases. See [Converting STC User Profiles to a Non-Zero UID](#) for details on how to convert the STC user profiles from UID 0 to a non-zero UID value.

#### Stonebranch Solutions 4.3.0

Universal Automation Center Registration Server (UAR) is installed as FMID TUAR430. UAR provides the ability to register the Universal Broker with Automation Center 1.6 and 1.7.

## Workload Automation 5

Universal Automation Center Agent (UAG) is installed as FMID TUAG510. UAG provides agent services for an Opwise Automation Center Controller, enabling it to schedule workload, transfer files, and monitor events on the agent system.

UAG replaces Universal Automation Center Registration Server (UAR), introduced in Stonebranch Solutions 4.3.0.



**Note**

Installation of Workload Automation 5 for z/OS does not upgrade an Opwise 1.7, 1.6, or 1.5 z/OS agent to UAG.

## Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opwise Automation Center 5.1.1](#) and [Indesca-Infitran 5.1.0](#) page to make sure that your platform is supported before upgrading.

## z/OS Installation - Distribution File

- z/OS Distribution File
- Obtaining the Distribution File
- Distribution File Format
- Distribution File Contents
  - z/OS USS Files

### z/OS Distribution File

The Workload Automation 5 for z/OS product distribution file contains all of the files required for the installation of the Workload Automation for z/OS package.

### Obtaining the Distribution File

To obtain the Workload Automation for z/OS package, download the corresponding product distribution file from the Stonebranch [Customer Portal](#).



**Note**

A customer user name and password — provided by Stonebranch, Inc. — are required to access the Customer Portal.

### Distribution File Format

The Workload Automation 5 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 Workload Automation 5 for z/OS distribution file has the following format:

**sb-Version.Release.Modification Level.Maintenance Level-operating system.exe**

For example: **sb-5.1.0.0-zOS.exe**

### Distribution File Contents

The following table lists the installation files (in XMT file format) included in the Workload Automation 5 for z/OS distribution file.



**Note**

These files include the files for z/OS USS.

File Name	Description
INSTALL.XMT	Workload Automation 5 package installation JCL.
README.TXT	Documentation on the package and transfer methods.
SMPMCS.XMT	Workload Automation 5 SMP/E MCS statements.
UAG510F1.XMT	Universal Automation Center Agent SMP/E FMID TUAG510 relative file 1.

<b>UAG510F2.XMT</b>	Universal Automation Center Agent SMP/E FMID TUAG510 relative file 2.
<b>UAG510F3.XMT</b>	Universal Automation Center Agent SMP/E FMID TUAG510 relative file 3.
<b>UBR510F1.XMT</b>	Universal Broker SMP/E FMID TUBR510 relative file 1.
<b>UBR510F2.XMT</b>	Universal Broker SMP/E FMID TUBR510 relative file 2.
<b>UBR510F3.XMT</b>	Universal Broker SMP/E FMID TUBR510 relative file 3.
<b>UCM510F1.XMT</b>	Universal Command SMP/E FMID TUCM510 relative file 1.
<b>UCM510F2.XMT</b>	Universal Command SMP/E FMID TUCM510 relative file 2.
<b>UCM510F3.XMT</b>	Universal Command SMP/E FMID TUCM510 relative file 3.
<b>UDM510F1.XMT</b>	Universal Data Mover SMP/E FMID TUDM510 relative file 1.
<b>UDM510F2.XMT</b>	Universal Data Mover SMP/E FMID TUDM510 relative file 2.
<b>UDM510F3.XMT</b>	Universal Data Mover SMP/E FMID TUDM510 relative file 3.
<b>UEC510F1.XMT</b>	Universal Enterprise Controller SMP/E FMID TUEC510 relative file 1.
<b>UEC510F2.XMT</b>	Universal Enterprise Controller SMP/E FMID TUEC510 relative file 2.
<b>UEM510F1.XMT</b>	Universal Event Monitor SMP/E FMID TUEM510 relative file 1.
<b>UEM510F2.XMT</b>	Universal Event Monitor SMP/E FMID TUEM510 relative file 2.
<b>UEM510F3.XMT</b>	Universal Event Monitor SMP/E FMID TUEM510 relative file 3.
<b>UNV510F1.XMT</b>	Universal Common SMP/E FMID TUNV510 relative file 1.
<b>USD510F1.XMT</b>	SAP RFC DLL FMID TUSD510 relative file 1.
<b>USP510F1.XMT</b>	Universal Connector FMID TUSP510 relative file 1.
<b>USP510F2.XMT</b>	Universal Connector FMID TUSP510 relative file 2.



<b>USP510F3.XMT</b>	Universal Connector SMP/E FMID TUSP510 relative file 3.
<b>UTL510F1.XMT</b>	Universal Utilities SMP/E FMID TUTL510 relative file 1.
<b>UTL510F2.XMT</b>	Universal Utilities SMP/E FMID TUTL510 relative file 2.
<b>UXD510F1.XMT</b>	IBM XML Toolkit FMID TUXD510 relative file 1.
<b><i>z/OS USS Files</i></b>	
<b>UBR510U1.XMT</b>	USS Universal Broker FMID UUBR510 relative file 1.
<b>UBR510U2.XMT</b>	USS Universal Broker FMID UUBR510 relative file 2.
<b>UCM510U1.XMT</b>	USS Universal Command FMID UUCM510 relative file 1.
<b>UCM510U2.XMT</b>	USS Universal Command FMID UUCM510 relative file 2.
<b>UDM510U1.XMT</b>	USS Universal Data Mover FMID UUDM510 relative file 1.
<b>UDM510U2.XMT</b>	USS Universal Data Mover FMID UUDM510 relative file 2.
<b>UEM510U1.XMT</b>	USS Universal Event Monitor FMID UUEM510 relative file 1.
<b>UEM510U2.XMT</b>	USS Universal Event Monitor FMID UUEM510 relative file 2.
<b>UNV510U1.XMT</b>	USS Universal Common FMID UUNV510 relative file 1.
<b>UTL510U1.XMT</b>	USS Universal Utilities FMID UUTL510 relative file 1.

## z/OS Installation - Transferring Installation Files to z/OS

- z/OS Installation - Transferring Installation Files to z/OS
- Methods of Transferring the Installation Files
  - Express/OS
  - Manual
- Express/OS Method
- Manual Method
  - Extract the Installation Files
  - Allocate z/OS Data Sets
  - Transfer the Installation Files
  - Unpack the Data Sets

## z/OS Installation – Transferring Installation Files to z/OS

After downloading the distribution file, you must extract the installation files and transfer them to the z/OS system on which to install Workload Automation for z/OS.

The XMIT (TSO TRANSMIT format) data sets must be unpacked with the TSO RECEIVE command.

### Methods of Transferring the Installation Files

There are two methods of transferring the installation files.

#### Express/OS

The [Express/OS method](#) 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.



#### Note

With z/OS versions 1.13 and later, there is an issue with Express/OS login credentials that inhibits its use for transferring installation files to z/OS. You must use the manual method.

#### Manual

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.

In cases where you cannot use FTP to transfer files from the Windows workstation to the z/OS system, the manual method is required.

### 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 `sb-5.1.0.<level>-zOS.exe` file in Windows Explorer. Express/OS then prompts you for information as it performs the following steps:

**Step 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.

<b>Step 2</b>	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.
<b>Step 3</b>	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.
<b>Step 4</b>	Express/OS prompts you for the data set high-level qualifier with which to allocate the transferred XMT files. <ul style="list-style-type: none"> <li>• For non-SMS data sets, provide the unit and volume serial number on which to allocate the data sets.</li> <li>• For SMS data sets, provide the management class and storage class with which to allocate the data sets.</li> </ul>
<b>Step 5</b>	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.
<b>Step 6</b>	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.
<b>Step 7</b>	Express/OS transfers the installation files to the allocated data sets on z/OS using FTP.
<b>Step 8</b>	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 Workload Automation 5 data sets are unpacked on the z/OS system and ready for installation (see [z/OS Installation - SMPE Installation](#)).

## Manual Method

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

To transfer the installation files to z/OS manually, you must perform the following four tasks:

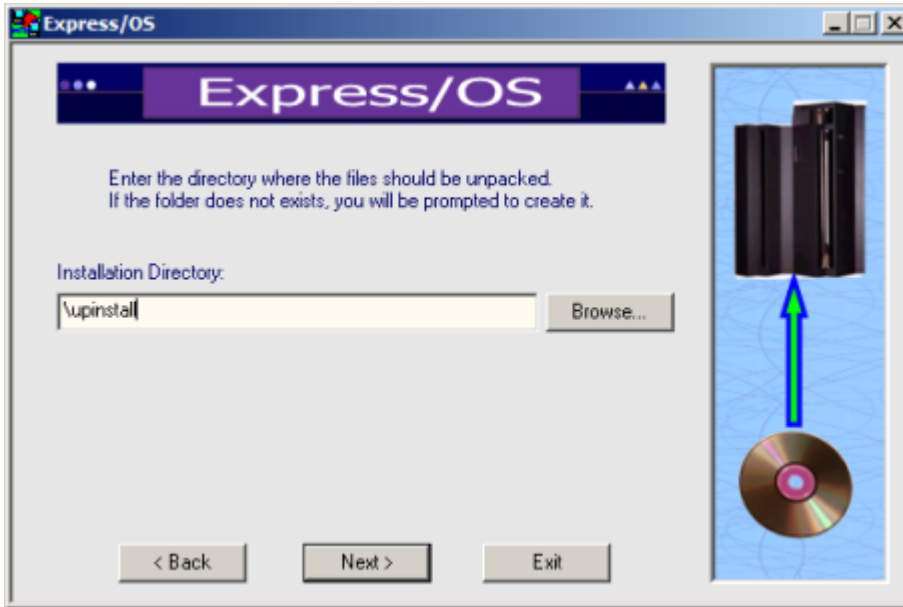
<b>1</b>	Extract the installation files from the distribution file (using Express/OS).
<b>2</b>	Allocate data sets on z/OS for the installation files.
<b>3</b>	Transfer the Installation Files.
<b>4</b>	Unpack the data sets.

### Extract the Installation Files

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

<b>Step 1</b>	Double-click the name of the distribution file, <b>sb-5.1.0.&lt;level&gt;-zOS.exe</b> , in Windows Explorer.
<b>Step 2</b>	On the introductory Express/OS screen, click the <b>Next&gt;</b> button.

**Step 3** The next screen, shown below, prompts for a directory in which to extract the installation files.



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**.)

**Step 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.

**Step 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)

You can use the following sample JCL to allocate and catalog the XMIT data sets on z/OS. Transfer it – to the z/OS system on which Workload Automation 5 for z/OS is being installed – 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 2000-2011 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.V5R1M0.SMPMCS.XMIT,
//           DISP=(,CATLG),
//           DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//           UNIT=SYSALLDA,VOL=SER=#VOLSER,
//           SPACE=(CYL,(200,1))
//INSTALL    DD  DSN=#HLQ.UNV.V5R1M0.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.V5R1M0.TUNV510.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.V5R1M0.TUTL510.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.V5R1M0.TUTL510.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.V5R1M0.TUBR510.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.V5R1M0.TUBR510.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.V5R1M0.TUBR510.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.V5R1M0.TUCM510.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.V5R1M0.TUCM510.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.V5R1M0.TUCM510.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.V5R1M0.TUDM510.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.V5R1M0.TUDM510.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.V5R1M0.TUDM510.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.V5R1M0.TUEM510.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.V5R1M0.TUEM510.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.V5R1M0.TUEM510.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.V5R1M0.TUXD510.F1.XMIT,
//          DISP=( ,CATLG) ,
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120) ,
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(50,1))
//UAGF1    DD  DSN=#HLQ.UNV.V5R1M0.TUAG510.F1.XMIT,
//          DISP=( ,CATLG) ,
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120) ,
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(1,1))
//UAGF2    DD  DSN=#HLQ.UNV.V5R1M0.TUAG510.F2.XMIT,
//          DISP=( ,CATLG) ,
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120) ,
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(1,1))
//UAGF3    DD  DSN=#HLQ.UNV.V5R1M0.TUAG510.F3.XMIT,
//          DISP=( ,CATLG) ,
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120) ,
//          UNIT=SYSALLDA,VOL=SER=#VOLSER,
//          SPACE=(CYL,(15,1))
//UECF1    DD  DSN=#HLQ.UNV.V5R1M0.TUEC510.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.V5R1M0.TUEC510.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.V5R1M0.TUSD510.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.V5R1M0.TUSP510.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.V5R1M0.TUSP510.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.V5R1M0.TUSP510.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.V5R1M0.UUBR510.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.V5R1M0.UUBR510.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.V5R1M0.UUCM510.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.V5R1M0.UUCM510.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.V5R1M0.UUDM510.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.V5R1M0.UUDM510.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.V5R1M0.UUEM510.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.V5R1M0.UUEM510.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.V5R1M0.UUTL510.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.V5R1M0.UUNV510.F1.XMIT,
//          DISP=(,CATLG),
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
```

```
//          UNIT=SYSALLDA, VOL=SER=#VOLSER,
//          SPACE=(CYL,(1,1))
```

## Transfer the Installation Files

Transfer the installation files to the [allocated z/OS XMIT data sets](#). A binary transfer must be performed; otherwise, the files will not unpack correctly.

Due to the numerous file transfer products available on the market, it is impractical to provide instructions for them all. FTP is one of the most popular; an example Windows FTP script is provided below. It can be used from a Windows workstation that can establish a TCP/IP connection to the z/OS host.

The following FTP script modifications are required:

<b>Step 1</b>	Change #HOSTNAME to the z/OS host name or IP address.
<b>Step 2</b>	Change #USERNAME to a z/OS user ID that has updated access to the XMIT data sets being updated.
<b>Step 3</b>	Change #PASSWORD to the z/OS user ID's password.
<b>Step 4</b>	Change #HLQ to the data set high level qualifier used to allocate the XMIT data sets.

```
open #HOSTNAME
#USERNAME
#PASSWORD

binary
put INSTALL.XMT '#HLQ.UNV.V5R1M0.INSTALL.XMIT'
put SMPMCS.XMT '#HLQ.UNV.V5R1M0.SMPMCS.XMIT'
put UNV510F1.XMT '#HLQ.UNV.V5R1M0.TUNV510.F1.XMIT'
put UTL510F1.XMT '#HLQ.UNV.V5R1M0.TUTL510.F1.XMIT'
put UTL510F2.XMT '#HLQ.UNV.V5R1M0.TUTL510.F2.XMIT'
put UBR510F1.XMT '#HLQ.UNV.V5R1M0.TUBR510.F1.XMIT'
put UBR510F2.XMT '#HLQ.UNV.V5R1M0.TUBR510.F2.XMIT'
put UBR510F3.XMT '#HLQ.UNV.V5R1M0.TUBR510.F3.XMIT'
put UCM510F1.XMT '#HLQ.UNV.V5R1M0.TUCM510.F1.XMIT'
put UCM510F2.XMT '#HLQ.UNV.V5R1M0.TUCM510.F2.XMIT'
put UCM510F3.XMT '#HLQ.UNV.V5R1M0.TUCM510.F3.XMIT'
put UDM510F1.XMT '#HLQ.UNV.V5R1M0.TUDM510.F1.XMIT'
put UDM510F2.XMT '#HLQ.UNV.V5R1M0.TUDM510.F2.XMIT'
put UDM510F3.XMT '#HLQ.UNV.V5R1M0.TUDM510.F3.XMIT'
put UEM510F1.XMT '#HLQ.UNV.V5R1M0.TUEM510.F1.XMIT'
put UEM510F2.XMT '#HLQ.UNV.V5R1M0.TUEM510.F2.XMIT'
put UEM510F3.XMT '#HLQ.UNV.V5R1M0.TUEM510.F3.XMIT'
put UXD510F1.XMT '#HLQ.UNV.V5R1M0.TUXD510.F1.XMIT'
put UAG510F1.XMT '#HLQ.UNV.V5R1M0.TUAG510.F1.XMIT'
put UAG510F2.XMT '#HLQ.UNV.V5R1M0.TUAG510.F2.XMIT'
put UAG510F3.XMT '#HLQ.UNV.V5R1M0.TUAG510.F3.XMIT'
put UEC510F1.XMT '#HLQ.UNV.V5R1M0.TUEC510.F1.XMIT'
put UEC510F2.XMT '#HLQ.UNV.V5R1M0.TUEC510.F2.XMIT'
put USD510F1.XMT '#HLQ.UNV.V5R1M0.TUSD510.F1.XMIT'
put USP510F1.XMT '#HLQ.UNV.V5R1M0.TUSP510.F1.XMIT'
put USP510F2.XMT '#HLQ.UNV.V5R1M0.TUSP510.F2.XMIT'
put USP510F3.XMT '#HLQ.UNV.V5R1M0.TUSP510.F3.XMIT'
put UBR510U1.XMT '#HLQ.UNV.V5R1M0.UUBR510.F1.XMIT'
put UBR510U2.XMT '#HLQ.UNV.V5R1M0.UUBR510.F2.XMIT'
put UCM510U1.XMT '#HLQ.UNV.V5R1M0.UUCM510.F1.XMIT'
put UCM510U2.XMT '#HLQ.UNV.V5R1M0.UUCM510.F2.XMIT'
put UDM510U1.XMT '#HLQ.UNV.V5R1M0.UUDM510.F1.XMIT'
put UDM510U2.XMT '#HLQ.UNV.V5R1M0.UUDM510.F2.XMIT'
put UEM510U1.XMT '#HLQ.UNV.V5R1M0.UUEM510.F1.XMIT'
put UEM510U2.XMT '#HLQ.UNV.V5R1M0.UUEM510.F2.XMIT'
put UTL510U1.XMT '#HLQ.UNV.V5R1M0.UUTL510.F1.XMIT'
put UNV510U1.XMT '#HLQ.UNV.V5R1M0.UUNV510.F1.XMIT'
quit
```

The FTP script is executed with the FTP `-s` option. Using a Command Prompt window, change to the directory in which Express/OS unpacked the installation files. The example above used directory name `\upinstall`. Assuming the script is saved in file name `ftp.script`, the FTP script is executed with the following FTP command:



```
C:\upinstall> ftp -s:ftp.script
```

The FTP output must be reviewed to confirm each of the files are transferred successfully.

## Unpack the Data Sets

After transferring the installation files to their allocated data sets on z/OS, you must unpack them using the TSO RECEIVE command. You can execute the TSO RECEIVE command interactively or in batch.

You can use the following sample JCL to unpack the data sets. Transfer it – to the z/OS system on which the Workload Automation 5 for z/OS is being installed – 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=*
//SYSPRT   DD SYSOUT=*
//SYSTEM   DD SYSOUT=*
//SYSTSIN  DD DUMMY
//         PEND
//*
//SMPMCS   EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.SMPMCS.XMIT')
DA('#HLQ.UNV.V5R1M0.SMPMCS') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(3000 150) RELEASE
//INSTALL EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.INSTALL.XMIT')
DA('#HLQ.UNV.V5R1M0.INSTALL') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(15 10)
//UNV510F1 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUNV510.F1.XMIT')
DA('#HLQ.UNV.V5R1M0.TUNV510.F1') +
UNIT(SYSALLDA) VOL(#VOLSER) +
```

```

TRACKS SPACE(15 1) RELEASE
//UTL510F1 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUTL510.F1.XMIT')
DA('#HLQ.UNV.V5R1M0.TUTL510.F1') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(15 1) RELEASE
//UTL510F2 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUTL510.F2.XMIT')
DA('#HLQ.UNV.V5R1M0.TUTL510.F2') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(910 10) RELEASE
//UBR510F1 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUBR510.F1.XMIT')
DA('#HLQ.UNV.V5R1M0.TUBR510.F1') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(15 1) RELEASE
//UBR510F2 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUBR510.F2.XMIT')
DA('#HLQ.UNV.V5R1M0.TUBR510.F2') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(15 1) RELEASE
//UBR510F3 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUBR510.F3.XMIT')
DA('#HLQ.UNV.V5R1M0.TUBR510.F3') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(1600 15) RELEASE
//UCM510F1 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUCM510.F1.XMIT')
DA('#HLQ.UNV.V5R1M0.TUCM510.F1') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(10 1) RELEASE
//UCM510F2 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUCM510.F2.XMIT')
DA('#HLQ.UNV.V5R1M0.TUCM510.F2') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(10 1) RELEASE
//UCM510F3 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUCM510.F3.XMIT')
DA('#HLQ.UNV.V5R1M0.TUCM510.F3') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(600 15) RELEASE
//UDM510F1 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUDM510.F1.XMIT')
DA('#HLQ.UNV.V5R1M0.TUDM510.F1') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(10 1) RELEASE
//UDM510F2 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUDM510.F2.XMIT')
DA('#HLQ.UNV.V5R1M0.TUDM510.F2') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(10 1) RELEASE
//UDM510F3 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUDM510.F3.XMIT')
DA('#HLQ.UNV.V5R1M0.TUDM510.F3') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(750 15) RELEASE
//UEM510F1 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUEM510.F1.XMIT')
DA('#HLQ.UNV.V5R1M0.TUEM510.F1') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(10 1) RELEASE
//UEM510F2 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUEM510.F2.XMIT')
DA('#HLQ.UNV.V5R1M0.TUEM510.F2') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(10 1) RELEASE
//UEM510F3 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUEM510.F3.XMIT')
DA('#HLQ.UNV.V5R1M0.TUEM510.F3') +

```

```

UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(200 15) RELEASE
//UXD510F1 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUXD510.F1.XMIT')
DA('#HLQ.UNV.V5R1M0.TUXD510.F1') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(700 10) RELEASE
//UAG510F1 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUAG510.F1.XMIT')
DA('#HLQ.UNV.V5R1M0.TUAG510.F1') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(10 1) RELEASE
//UAG510F2 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUAG510.F2.XMIT')
DA('#HLQ.UNV.V5R1M0.TUAG510.F2') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(10 1) RELEASE
//UAG510F3 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUAG510.F3.XMIT')
DA('#HLQ.UNV.V5R1M0.TUAG510.F3') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(300 15) RELEASE
//UEC510F1 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUEC510.F1.XMIT')
DA('#HLQ.UNV.V5R1M0.TUEC510.F1') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(10 1) RELEASE
//UEC510F2 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUEC510.F2.XMIT')
DA('#HLQ.UNV.V5R1M0.TUEC510.F2') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(1000 15) RELEASE
//USD510F1 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUSD510.F1.XMIT')
DA('#HLQ.UNV.V5R1M0.TUSD510.F1') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(300 15) RELEASE
//USP510F1 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUSP510.F1.XMIT')
DA('#HLQ.UNV.V5R1M0.TUSP510.F1') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(10 1) RELEASE
//USP510F2 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUSP510.F2.XMIT')
DA('#HLQ.UNV.V5R1M0.TUSP510.F2') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(10 1) RELEASE
//USP510F3 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.TUSP510.F3.XMIT')
DA('#HLQ.UNV.V5R1M0.TUSP510.F3') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(200 15) RELEASE
//UBR510U1 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.UUBR510.F1.XMIT')
DA('#HLQ.UNV.V5R1M0.UUBR510.F1') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(10 1) RELEASE
//UBR510U2 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.UUBR510.F2.XMIT')
DA('#HLQ.UNV.V5R1M0.UUBR510.F2') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(500 50) RELEASE
//UCM510U1 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.UUCM510.F1.XMIT')
DA('#HLQ.UNV.V5R1M0.UUCM510.F1') +
UNIT(SYSALLDA) VOL(#VOLSER) +
TRACKS SPACE(10 1) RELEASE
//UCM510U2 EXEC TSO
//S1.SYSTSIN DD *
RECEIVE INDA('#HLQ.UNV.V5R1M0.UUCM510.F2.XMIT')

```

```

    DA('#HLQ.UNV.V5R1M0.UUCM510.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(500 50) RELEASE
//UDM510U1 EXEC TSO
//S1.SYSTSIN DD *
    RECEIVE INDA('#HLQ.UNV.V5R1M0.UUDM510.F1.XMIT')
    DA('#HLQ.UNV.V5R1M0.UUDM510.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UDM510U2 EXEC TSO
//S1.SYSTSIN DD *
    RECEIVE INDA('#HLQ.UNV.V5R1M0.UUDM510.F2.XMIT')
    DA('#HLQ.UNV.V5R1M0.UUDM510.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(500 50) RELEASE
//UEM510U1 EXEC TSO
//S1.SYSTSIN DD *
    RECEIVE INDA('#HLQ.UNV.V5R1M0.UUEM510.F1.XMIT')
    DA('#HLQ.UNV.V5R1M0.UUEM510.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(10 1) RELEASE
//UEM510U2 EXEC TSO
//S1.SYSTSIN DD *
    RECEIVE INDA('#HLQ.UNV.V5R1M0.UUEM510.F2.XMIT')
    DA('#HLQ.UNV.V5R1M0.UUEM510.F2') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(500 50) RELEASE
//UTL510U1 EXEC TSO
//S1.SYSTSIN DD *
    RECEIVE INDA('#HLQ.UNV.V5R1M0.UUTL510.F1.XMIT')
    DA('#HLQ.UNV.V5R1M0.UUTL510.F1') +
    UNIT(SYSALLDA) VOL(#VOLSER) +
    TRACKS SPACE(650 15) RELEASE
//UNV510U1 EXEC TSO
//S1.SYSTSIN DD *
    RECEIVE INDA('#HLQ.UNV.V5R1M0.UUNV510.F1.XMIT')
    DA('#HLQ.UNV.V5R1M0.UUNV510.F1') +

```

```
UNIT(SYSALLDA) VOL(#VOLSER) +  
TRACKS SPACE(10 1) RELEASE
```

## **zOS Installation - SMPE Installation**

Error formatting macro: redirect: java.lang.NullPointerException

## zOS Installation - SMPE Installation Overview

- [SMP/E Installation of Workload Automation 5 for z/OS](#)
- [SMP/E](#)
  - [SMP/E FMIDs](#)

### SMP/E Installation of Workload Automation 5 for z/OS

Workload Automation 5 for z/OS is installed using SMP/E. These pages describe how to perform SMP/E installation of the Workload Automation for z/OS package in a step-by-step process.

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

1. Installing any Workload Automation 5 package from Stonebranch, Inc. for the first time, or installing a Workload Automation 5 package in a new SMP/E CSI.  
See [New Install, New CSI](#) for installation instructions.
2. Upgrading a Stonebranch Solutions 4.x package install. In this case, the Workload Automation 5 package is installed into an SMP/E CSI that contains a Stonebranch Solutions 4.x package.  
See [Stonebranch Solutions 4.x Upgrade, Existing CSI](#) for installation instructions.
3. Upgrading a Universal Products 3.2.0 package install. In this case, the Workload Automation 5 package is installed into an SMP/E CSI that contains a Universal Products 3.2.0 package.  
See [Universal Products 3.2.0 Upgrade, Existing CSI](#) for installation instructions.



#### Note

If you are upgrading from Universal Products 3.1.1 or earlier, you first must upgrade to Universal Products 3.2.0, and then upgrade to Workload Automation 5.

### SMP/E

The Workload Automation 5 for z/OS components are installed in the Workload Automation SMP/E CSI. This CSI should not be shared with any other vendor products; it should be used exclusively for Workload Automation.

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 Universal Products version 2.2.0, all Workload Automation components 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**.

### SMP/E FMIDs

The following table identifies the SMP/E FMIDs for the Workload Automation for z/OS components. (For the list of SMP/E FMIDs for the Workload Automation for z/OS USS components, see [z/OS USS Installation - Installation Requirements](#).)

Product	FMID	SMP/E Requisites
Universal Common 5.1.0	TUNV510	Supersedes and deletes FMID TUNV220, TUNV310, TUNV311, TUNV320, TUNV410, TUNV420, and TUNV430.
Universal Automation Center Agent 5.1.0	TUAG510	TUNV510 is a prerequisite.
Universal Broker 5.1.0	TUBR510	TUNV510 is a prerequisite. Supersedes and deletes FMID TUBR310, TUBR311, TUBR320, TUBR410, TUBR420, and TUBR430.
Universal Command 5.1.0	TUCM510	TUBR510 is a prerequisite. Supersedes and deletes FMID TUCM110, TUCM120,

		TUCM210, TUCM220, TUCM310, TUCM311, TUCM320, TUCM410, TUCM420, and TUCM430.
Universal Enterprise Controller 5.1.0	TUEC510	TUNV510 is a prerequisite. Supersedes and deletes FMID TUEC110, TUEC310, TUEC320, TUEC410, TUEC420, and TUEC430.
Universal Data Mover 5.1.0	TUDM510	TUBR510 is a prerequisite. Supersedes and deletes FMID TUDM110, TUDM310, TUDM311, TUDM320, TUDM410, TUDM420, and TUDM430.
SAP RFC DLL	TUSD510	TUNV510 is a prerequisite. Supersedes and deletes FMID TUSD310, TUSD311, TUSD320, TUSD410, TUSD420, and TUSD430.
Universal Connector	TUSP510	TUSD510 is a prerequisite. Supersedes and deletes FMID TUSP120, TUSP310, TUSP311, TUSP320, TUSP410, TYSP420, and TUSP430.
Universal Utilities	TUTL510	TUNV510 is a prerequisite. Supersedes and deletes FMID TUEN110, TUEN120, TUEN210, TUEN220, TUEN310, TUEN311, TUTL320, TUTL410, TUTL420, and TUTL430.
Universal Event Monitor	TUEM510	TUBR510 is a prerequisite. Supersedes and deletes FMID TUEM310, TUEM311, TUEM320, TUEM410, TUEM420, and TUEM430.
IBM XML Toolkit	TUXD510	TUNV510 is a prerequisite. Supersedes and deletes TUXD320, TUXD410, TUXD420, and TUXD430.



## zOS Installation - New Install, New CSI

### New Install, New CSI

The New Install, New CSI installation process describes how to install the Workload Automation 5 package in a newly allocated SMP/E CSI.

Use this installation process for either of these environments:

- Installing a Workload Automation 5 package for the first time.
- Installing a Workload Automation 5 package in a different SMP/E CSI than other Workload Automation 5 components.

The installation JCL referenced by the following installation steps is created by the **#SETUP** member in the Workload Automation 5 **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.

<b>Step 1</b>	<p>Edit and submit the JCL in member <b>#SETUP</b>.</p> <p>The <b>#SETUP</b> 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 <b>INSTALL</b> library.</p> <p>If you want to change the customizations after the job is executed, edit the <b>#SETUP</b> member with the new customizations and resubmit the job. All <b>INSTALL</b> library members will be replaced.</p> <p>All steps must end with a return code 0.</p>
<b>Step 2</b>	<p>Submit the JCL in member <b>UNVIN01</b>. The JCL allocates the SMP/E CSI data sets.</p> <p>All steps must end with a return code 0.</p>
<b>Step 3</b>	<p>Submit the JCL in member <b>UNVIN02</b>. The JCL initializes the SMP/E CSI.</p> <p>All steps must end with a return code 0.</p>
<b>Step 4</b>	<p>Submit the JCL in member <b>UNVIN03</b>. The JCL allocates Workload Automation 5 target and distribution data sets and adds SMP/E <b>DDEF</b> definitions to the Workload Automation 5 CSI zones.</p> <p>All steps must end with a return code 0.</p>
<b>Step 5</b>	<p>Submit the JCL in member <b>UNVIN04</b>. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets.</p> <p>All steps must end with a return code 0.</p>
<b>Step 6</b>	<p>Submit the JCL in member <b>UNVIN05</b>. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs.</p> <p>Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions:</p> <ul style="list-style-type: none"> <li>• Step ends with condition code 0.</li> <li>• Step ends with condition code 4, and message GIM42001W is written in ddname <b>SMPOUT</b>.</li> <li>• Step ends with condition code 12, and message GIM24801S is written in ddname <b>SMPOUT</b>.</li> </ul>
<b>Step 7</b>	<p>Submit the JCL in member <b>UNVIN06</b>. The JCL member performs an SMP/E ACCEPT of the product FMIDs and any applied PTFs.</p> <p>Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions:</p> <ul style="list-style-type: none"> <li>• Step ends with condition code 0 or 4.</li> <li>• Step ends with condition code 12, and message GIM24801S is written in ddname <b>SMPOUT</b>.</li> </ul>
<b>Step 8</b>	<p>Submit the JCL in member <b>UNVIN07</b>. The JCL member allocates product non-SMP/E data sets and databases, and formats the zFS data sets.</p> <p>All steps must end with a return code 0.</p>

<b>Step 9</b>	<p>The product databases are allocated as z/OS UNIX zFS file systems. Additional product configuration is required to utilize zFS data sets. See <a href="#">Workload Automation 5 Database Configuration</a> for details.</p> <p>If you prefer to use HFS file systems, HFS data sets must be allocated. If you prefer to use the zFS data sets, skip this step.</p> <p>Submit the JCL in member <b>UNVINHF</b>. The JCL renames the zFS data sets and defines their HFS counterparts.</p> <p>All steps must end with return code 0.</p>
<b>Step 10</b>	<p>Submit the JCL in member <b>UNVIN08</b>. The JCL copies sample configuration members to the configuration libraries.</p> <p>All steps must end with a return code 0.</p>
<b>Step 11</b>	<p>Edit and submit the JCL in member <b>UNVIN09</b>. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL.</p> <p>The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library.</p> <p>All steps must end with a return code 0.</p>
<b>Step 12</b>	<p>Perform required z/OS configuration steps as described in <a href="#">z/OS Installation - Configuration</a>.</p>

## z/OS Installation - Stonebranch Solutions 4.x Upgrade, Existing CSI

### Stonebranch Solutions 4.x Upgrade, Existing CSI

The Stonebranch Solutions 4.x Upgrade, Existing CSI installation process describes how to upgrade an existing Stonebranch Solutions 4.x package in an existing SMP/E CSI.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Workload Automation 5 **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.

<p><b>Step 1</b></p>	<p>Edit and submit the JCL in member <b>#SETUP</b>.</p> <p>The <b>#SETUP</b> 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 <b>INSTALL</b> library.</p> <p>To change customizations after the job is executed, edit the <b>#SETUP</b> member with the new customizations and resubmit the job. All <b>INSTALL</b> library members will be replaced.</p> <p>All steps must end with a return code 0.</p>
<p><b>Step 2</b></p>	<p>Submit the JCL in member <b>UNVUG4XX</b>. The JCL removes the UAR configuration and component definition members in the non-SMP/E UNVCONF and UNVCOMP libraries, respectively, and allocates the UAG model data set.</p> <p>All steps must end with return code 0.</p>
<p><b>Step 3</b></p>	<p>Submit the JCL in member <b>UNVIN04</b>. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets.</p> <p>All steps must end with a return code 0.</p>
<p><b>Step 4</b></p>	<p>Submit the JCL in member <b>UNVIN05</b>. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs.</p> <p>Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions:</p> <ul style="list-style-type: none"> <li>• Step ends with condition code 0.</li> <li>• Step ends with condition code 4, and message GIM42001W is written in ddname <b>SMPOUT</b>.</li> <li>• Step ends with condition code 12, and message GIM24801S is written in ddname <b>SMPOUT</b>.</li> </ul>
<p><b>Step 5</b></p>	<p>Submit the JCL in member <b>UNVIN06</b>. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs.</p> <p>Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions:</p> <ul style="list-style-type: none"> <li>• Step ends with condition code 0 or 4.</li> <li>• Step ends with condition code 12, and message GIM24801S is written in ddname SMPOUT.</li> </ul>
<p><b>Step 6</b></p>	<p>Submit the JCL in member <b>UNVIN08</b>. The JCL copies sample configuration members to the configuration libraries.</p> <p>All steps must end with a return code 0.</p>
<p><b>Step 7</b></p>	<p>Edit and submit the JCL in member <b>UNVIN09</b>. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL.</p> <p>The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library.</p> <p>All steps must end with a return code 0.</p>
<p><b>Step 8</b></p>	<p>Perform required z/OS configuration steps as described in <a href="#">z/OS Installation - Configuration</a>.</p>

## zOS Installation - Universal Products 3.2.0 Upgrade, Existing CSI

### 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 Workload Automation 5 **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.

<p><b>Step 1</b></p>	<p>Edit and submit the JCL in member <b>#SETUP</b>.</p> <p>The <b>#SETUP</b> 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 <b>INSTALL</b> library.</p> <p>To change customizations after the job is executed, edit the <b>#SETUP</b> member with the new customizations and resubmit the job. All <b>INSTALL</b> library members will be replaced.</p> <p>All steps must end with a return code 0.</p>
<p><b>Step 2</b></p>	<p>Submit the JCL in member <b>UNVUG320</b>. The JCL upgrades the SMP/E Universal Products configuration from 3.2.0 to 5.1.0, and allocates the UAG model data set.</p> <p>The following changes are made:</p> <ul style="list-style-type: none"> <li>• Allocates a new AUNVHBIN SMP/E distribution data set, if necessary.</li> </ul>
<p><b>Step 3</b></p>	<p>Submit the JCL in member <b>UNVIN04</b>. The JCL performs an SMP/E RECEIVE of the product FMIDs and available PTFs from the distribution data sets.</p> <p>All steps must end with a return code 0.</p>
<p><b>Step 4</b></p>	<p>Submit the JCL in member <b>UNVIN05</b>. The JCL performs an SMP/E APPLY of the product FMIDs and any received PTFs.</p> <p>Step APYFMID must end with a condition code of 0. Step APYPTFS is considered successful under any of the following conditions:</p> <ul style="list-style-type: none"> <li>• Step ends with condition code 0.</li> <li>• Step ends with condition code 4, and message GIM42001W is written in ddname <b>SMPOUT</b>.</li> <li>• Step ends with condition code 12, and message GIM24801S is written in ddname <b>SMPOUT</b>.</li> </ul>
<p><b>Step 5</b></p>	<p>Submit the JCL in member <b>UNVIN06</b>. The JCL member performs an SMP/E ACCEPT of product FMIDs and any applied PTFs.</p> <p>Step ACCFMID must end with a condition code of 0. Step ACCPTFS is considered successful under any of the following conditions:</p> <ul style="list-style-type: none"> <li>• Step ends with condition code 0 or 4.</li> <li>• Step ends with condition code 12, and message GIM24801S is written in ddname <b>SMPOUT</b>.</li> </ul>
<p><b>Step 6</b></p>	<p>Submit the JCL in member <b>UNVIN08</b>. The JCL copies sample configuration members to the configuration libraries.</p> <p>All steps must end with a return code 0.</p>
<p><b>Step 7</b></p>	<p>Edit and submit the JCL in member <b>UNVIN09</b>. The JCL requires modifications as listed in the MODIFICATIONS section of the comments at the top of the JCL.</p> <p>The JCL copies the Universal Enterprise Controller and Universal Broker started procedure JCL to a system procedure library.</p> <p>All steps must end with a return code 0.</p>
<p><b>Step 8</b></p>	<p>Perform required z/OS configuration steps as described in <a href="#">z/OS Installation - Configuration</a>.</p>

## z/OS Installation - Cumulative PTF Maintenance

- z/OS Installation - Cumulative PTF Maintenance
- Installation Procedures
- Cumulative PTF Maintenance Distribution File
  - Distribution File Content
- Transferring a Cumulative PTF File to z/OS
  - Express/OS Method
  - Manual Method
- Installing a Cumulative PTF File

## z/OS Installation – Cumulative PTF Maintenance

This page describes the procedures for installing a cumulative PTF maintenance for Workload Automation for z/OS.

### Installation Procedures

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

<b>Step 1</b>	Download the cumulative PTF maintenance distribution file.
<b>Step 2</b>	Transfer the cumulative PTF file from Windows to z/OS. <ol style="list-style-type: none"> <li>1. Extract the cumulative PTF file from the distribution file.</li> <li>2. Allocate a data set for the PTF file.</li> <li>3. Transfer the PTF file to the data set on z/OS.</li> <li>4. Unpack the data set.</li> </ol>
<b>Step 3</b>	Install the cumulative PTF file.

### Cumulative PTF Maintenance Distribution File

To obtain a cumulative PTF maintenance, you must download the corresponding distribution file from the Stonebranch [Customer Portal](#).



**Note**

A customer user name and password — provided by Stonebranch, Inc. — are required to access the Customer Portal.

### Distribution File Content

The following table lists the files included in a cumulative PTF maintenance distribution file.

File Name	Description
UNV510xx.XMT	Cumulative PTF file for Workload Automation 5 level xx.
README.TXT	Documentation on the package and transfer methods.

If you do not have these files, contact Stonebranch, Inc. Customer Support for a complete distribution file.

### 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:

<b>Step 1</b>	Unpacks the XMT PTF file into a specified directory.
<b>Step 2</b>	Transfers the XMT PTF file to the z/OS system using FTP in passive mode.
<b>Step 3</b>	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](#), below.

### PTF HOLDS

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 HOLDS 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( TUNV510, TUTL510, TUBR510, TUCM510, TUEC510, TUDM510, TUAG510, TUXD510, TUSD510, TUSP510,
  TUEM510, UUNV510, UUBR510, UUCM510, UUDM510, UUEM510, UUTL510 ).
```

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

```
ACCEPT PTFS
  BYPASS( HOLDSYSTEM( ACTION ) )
  FORFMID( TUNV510, TUTL510, TUBR510, TUCM510, TUEC510, TUDM510, TUAG510, TUXD510, TUSD510, TUSP510,
  TUEM510, UUNV510, UUBR510, UUCM510, UUDM510, UUEM510, UUTL510 ).
```

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.

<b>Step 1</b>	Allocate an z/OS data set to receive the PTF file as: <ul style="list-style-type: none"> <li>• DSORG=PS</li> <li>• RECFM=FB</li> <li>• LRECL=80</li> <li>• BLKSIZE=3120</li> </ul>
<b>Step 2</b>	Transfer the PTF file - using a binary transfer from your local workstation - to the XMIT data set allocated on z/OS in Step 1.
<b>Step 3</b>	Enter the following TSO command to RECEIVE the XMT file: <pre>TSO RECEIVE INDA('cum PTF data set name allocated in step 1')</pre>
<b>Step 4</b>	Press <Enter> at the INMR906A message prompt.
<b>Step 5</b>	Continue with the installation of the maintenance (see <a href="#">Installing a Cumulative PTF File</a> , below).

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

## 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.

<b>Step 1</b>	Member <b>UNVMTRCV</b> performs an SMP/E RECEIVE of the PTFs. Change the data set allocated on ddname <b>SMPPTFIN</b> to the PTF data set. Modify the JCL to meet local requirements, and submit the job.
<b>Step 2</b>	Member <b>UNVMTAPL</b> 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 <b>SMPOUT</b> . In this case, see <a href="#">PTF HOLDS</a> , above.
<b>Step 3</b>	Fully test the maintenance. Only after all testing is completed should you proceed to the next step.
<b>Step 4</b>	Member <b>UNVMTACC</b> 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 <b>SMPOUT</b> . In this case, see <a href="#">PTF HOLDS</a> , above.

## **zOS Installation - Configuration**

Error formatting macro: redirect: java.lang.NullPointerException



## zOS Configuration - Overview

### z/OS Configuration - Overview

These pages describe the z/OS configuration steps required for the Universal Broker, Universal Enterprise Controller, and Universal Automation Center Agent.

The Universal Broker component is a required component. Universal Enterprise Controller is optional. Universal Automation Center Agent is required to manage workload with Opwise Automation Center.

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, adding it to program control, and adding specific programs to the LNKLST.
- Installation of SMF exits.
- Configuration of Universal Spool and Universal Enterprise Controller HFS or zFS databases.

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

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

## z/OS Configuration - Started Tasks

- z/OS Configuration - Started Tasks
- Started Task Security
- Universal Broker
  - Universal Broker User and Group Profiles
  - Universal Broker Data Access
- Universal Enterprise Controller
  - Universal Enterprise Controller User and Group Profiles
  - Universal Enterprise Controller Data Access
- Associate Started Tasks with User and Group Profiles
  - STARTED Class Profile
  - Started Procedures Table

### z/OS Configuration - Started Tasks

The Workload Automation 5 solution consists of two z/OS started tasks:

1. Universal Broker is a required started task that provides a number of services for manager and server components.
2. Universal Enterprise Controller is an optional started task that provides monitoring and administration services.

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

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

The JCL procedures are copied to a system procedure library by installation job **UNVIN09**. If this was not done, copy the JCL members to the appropriate procedure library for your local environment.

The started task programs utilize both z/OS UNIX System Services (USS) and MVS services. As a result of using USS services, the programs execute as USS processes. As do all USS processes, the Universal Broker and Universal Enterprise Controller processes must execute with user profiles that have a properly defined OMVS segments. Additionally, the user profiles must be permitted security access to privileged USS services in order for them to perform specific functions.

This page describes the following configuration tasks:

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

### Started Task Security

z/OS UNIX System Services (USS) operates in one of two different security modes. If the resource profile BPX.DAEMON is not defined, USS is operating in UNIX-level security mode; otherwise, USS is operating in z/OS UNIX security mode.

UNIX-level security provides few options to control access to USS services. A user account that requires access to privileged services must be defined with a UID value of 0, which is referred to as superuser.

z/OS UNIX security provides better access controls to USS services using a SAF security package, such as RACF. A user account can be defined with a non-zero UID and granted permissions to specific resource profiles that protect USS services. Superuser access is granted not with UID 0 but with READ access to the resource BPX.SUPERUSER in the FACILITY class.

### Universal Broker

The Universal Broker started task provides services for local and remote Workload Automation managers, such as Universal Command managers or Universal Data Mover managers. Locally executed managers register with the local Universal Broker for monitoring, configuration data, and event recording. Remotely executed managers communicate with the local Universal Broker over a TCP/IP network connection and request execution of server components in order to process local services, such as execution of commands or transfer of data.

Server components initiated by the Universal Broker started task execute as child processes of the Universal Broker process. As such, the server components inherit the user identifier of the Universal Broker process. Some server components, such as Universal Command server, can switch the user identifier with which the work is executed. Switching a user identifier is a privileged operation. The Universal Broker user profile requires sufficient security access for itself and the server components to perform their services.

### Universal Broker User and Group Profiles

The security requirements of the Universal Broker depend upon which services are being utilized. By default, all services are configured to be

used. Some services can be disabled to reduce the amount of authority the Universal Broker user profile requires. The following table lists the USS privileged services for each component and how to disable the service so that security access to the service is not required.

Service	Description	Disabling
Change directory ownership	Universal Broker dynamically mounts its USS file systems. Once the file systems are mounted, the Broker will initialize them. Initialization consists of changing the ownership of the file systems root directories to the Broker user identifier.	Initialize the file system ownership manually. The Broker will not dynamically mount the file systems and initialize them if they are already mounted and initialized.
Create external links owned by UID 0	Universal Broker dynamically creates external links on the USS file system to the server component MVS programs in its STEPLIB ddname allocation. The external links are required for the USS <i>spawn</i> function used by the Universal Broker to execute the server components. The external links must be owned by UID 0 when they link to a MVS program that resides in an APF authorized library and are link edited with AC=1. Both the UAGSRV program and UDMSRV programs are link edited with AC=1. Creation of the external links so that they are owned by UID 0 requires superuser access.	Create the external links manually at a permanent location in the USS file system. The external links must be owned by UID 0. Specify the external link absolute path name in the component definition START_COMMAND option for both UAGSRV and UDMSRV, which is located in UNVCOMP(UAGCMP00) and UNVCOMP(UDSCMP00), respectively.
Switch user ID and group associations	Universal Command, Universal Data Mover, and Universal Control switch their user IDs with which a work request is executed. The user ID is first authenticated before switching unless there is a Universal Access Control List (UACL) entry that turns authentication off for the request.	Set the Universal Command, Universal Data Mover, and Universal Control servers configuration SECURITY option to a value of NONE. With SECURITY set to NONE, all work requests are executed with the Universal Broker's user ID.
Change server component job name	Universal Broker will set the job name of child server processes to the appropriate component name. For example, when the Universal Broker starts a Universal Command server component, the job name is set to UCMSRV.	There is no product configuration option to disable this. By simply not permitting the Broker to the resource profile that protects it, all server components will run with the Universal Broker job name with a numeric value appended to it (for example, UBROKER2).

How to configure the Universal Broker started task user profile to meet security requirements depends on whether USS is running with UNIX-level security or z/OS UNIX security. The following sections describe how to configure the Universal Broker user profile to perform the privileged services listed above for both USS UNIX-level security and z/OS UNIX security configurations.

### UNIX-level Security

UNIX-level security refers to a USS security environment where the resource profile BPX.DAEMON is not defined to the FACILITY class.

The only method of permitting a user profile access to privileged services is to define the user profile with a UID value of 0 (superuser). The Universal Broker user profile must be defined with UID 0 to perform any privileged service.

The following steps define the Universal Broker user profile for a UNIX-level security environment:

<b>Step 1</b>	<p>Add the Universal Broker group profile <b>UBRGRP</b> using the following RACF command:</p> <pre>ADDGROUP UBRGRP OWNER(SYS1) OMVS(GID(5001))</pre> <p>Change the GID value 5001 to a value suitable for your local USS environment. The GID value must be unique among all group profiles.</p>
<b>Step 2</b>	<p>Add the Universal Broker user profile <b>UBRUSR</b> using the following RACF command:</p> <pre>ADDUSER UBRUSR DFLTGRP(UBRGRP) OWNER(SYS1) NOPASSWORD OMVS(UID(0))</pre>

### z/OS UNIX Security

z/OS UNIX security refers to a USS security environment where the resource profile BPX.DAEMON is defined to the FACILITY class.

The Universal Broker user profile must be defined with a valid OMVS segment with a non-zero, unique UID value. The user profile security requirements are listed in the following table for each privileged service.

Service	Requirement
Change directory ownership	READ access to BPX.SUPERUSER resource profile in the FACILITY class.
Create external links owned by UID 0	READ access to BPX.SUPERUSER resource profile in the FACILITY class.
Switch user ID and group associations	READ access to BPX.DAEMON and BPX.SUPERUSER resource profiles in the FACILITY class.
Change server component job name	READ access to the BPX.JOBNAME profile in the FACILITY class.

The security requirements can be lifted if the feature that utilizes the service is disabled as described above.

The following steps configure the Universal Broker user profile for a z/OS UNIX security environment:

<b>Step 1</b>	<p>Add the Universal Broker group profile <b>UBRGRP</b> using the following RACF command:</p> <pre>ADDGROUP UBRGRP OWNER(SYS1) OMVS(GID(5001))</pre> <p>Change the GID value 5001 to a value suitable for your local USS environment. The GID value must be unique among all group profiles.</p>
<b>Step 2</b>	<p>Add the Universal Broker user profile <b>UBRUSR</b> using the following RACF command:</p> <pre>ADDUSER UBRUSR DFLTGRP(UBRGRP) OWNER(SYS1) NOPASSWORD OMVS(UID(5001))</pre> <p>Change the UID value 5001 to a value suitable for your local USS environment. The value must be unique among all user profiles.</p>
<b>Step 3</b>	<p>Permit the Universal Broker user profile READ access to the resource profiles required for enabled services. The following RACF commands permit the user profile to the resources required for all privileged services:</p> <pre>PE BPX.DAEMON CLASS(FACILITY) ID(UBRUSR) ACCESS(READ) PE BPX.SUPERUSER CLASS(FACILITY) ID(UBRUSR) ACCESS(READ) PE BPX.JOBNAME CLASS(FACILITY) ID(UBRUSR) ACCESS(READ) SETR RACLIST(FACILITY) REFRESH</pre>

### Universal Broker Data Access

The Universal Broker user profile **UBRUSR** requires the following access to the data sets allocated in the Universal Broker started task, **UBROKER**:

ddname	Access	Description
STEPLIB	READ	Program library
UNVCONF	ALTER	Product configuration data
UNVRFC	READ	Universal Connector SAP RFC file
UNVCOMP	ALTER	Product component definition data
UNVNLS	READ	Product national language support data
UNVCREF	READ	Universal Command command reference definitions
UNVDB	UPDATE	Universal Broker HFS component database
UNVSPool	UPDATE	Universal Broker HFS spool database

UNVTMPL	READ	Universal Broker configuration templates
UNVTRACE	UPDATE	Universal Broker application trace PDSE
UNVTRMDL	ALTER	Universal Broker application model trace data set
UNVLOG	UPDATE	Universal Broker log data set
UNVAGMDL	ALTER	Universal Automation Center Agent (UAG) model log data set

## Universal Enterprise Controller

The Universal Enterprise Controller started task provides services for monitoring and administering Universal Agents distributed throughout the computer network. GUI clients connect to Universal Enterprise Controller to perform tasks and view component activity and statuses.

### Universal Enterprise Controller User and Group Profiles

The security requirements of the Universal Enterprise Controller depend upon which services are being utilized. By default, all services are configured to be used. Some services can be disabled to reduce the amount of authority the Universal Enterprise Controller user profile requires. The following table lists the USS privileged services and how to disable the service so that security access to the service is not required.

Service	Description	Disabling
Mount file system	Universal Enterprise Controller dynamically mounts its USS file system. Mounting a file system requires APF authorization or superuser access. Universal Enterprise Controller is not APF authorized.	Statically mount the Universal Enterprise Controller file system.
Change directory ownership	Universal Enterprise Controller will initialize its file system if it detects initialization has not been completed. Initialization consists of changing the ownership of the file system root directory to the Universal Enterprise Controller user identifier.	Initialize the file system ownership manually.

How to configure the Universal Enterprise Controller started task user profile to meet security requirements depend on whether USS is running with UNIX-level security or z/OS UNIX security. The following sections describe how to configure the Universal Enterprise Controller user profile to perform the privileged services listed above for both USS UNIX-level security and z/OS UNIX security configurations.

### UNIX-level Security

UNIX-level security refers to a USS security environment where the resource profile BPX.DAEMON is not defined to the FACILITY class.

The only method of permitting a user profile access to privileged services is to define the user profile with a UID value of 0 (superuser). The Universal Enterprise Controller user profile must be defined with UID 0 to perform any privileged service.

The following steps define the Universal Enterprise Controller user profile for a UNIX-level security environment:

<b>Step 1</b>	<p>Add the Universal Enterprise Controller group profile <b>UECGRP</b> using the following RACF command:</p> <pre>ADDGROUP UECGRP OWNER(SYS1) OMVS(GID(5002))</pre> <p>Change the GID value 5002 to a value suitable for your local USS environment. The GID value must be unique among all group profiles.</p>
<b>Step 2</b>	<p>Add the Universal Enterprise Controller user profile <b>UECUSR</b> using the following RACF command:</p> <pre>ADDUSER UECUSR DFLTGRP(UECGRP) OWNER(SYS1) NOPASSWORD OMVS(UID(0))</pre>

### z/OS UNIX Security

z/OS UNIX security refers to a USS security environment where the resource profile BPX.DAEMON is defined to the FACILITY class.

The Universal Enterprise Controller user profile must be defined with a valid OMVS segment with a non-zero, unique UID value. The user profile security requirements are listed in the following table for each privileged service.

Service	Requirement
Mount file system	READ access to BPX.SUPERUSER resource profile in the FACILITY class.
Change directory ownership	READ access to BPX.SUPERUSER resource profile in the FACILITY class.

The security requirements can be lifted if the feature that utilizes the service is disabled as described above.

The following steps configure the Universal Enterprise Controller user profile for a z/OS UNIX security environment:

<b>Step 1</b>	<p>Add the Universal Enterprise Controller group profile <b>UECGRP</b> using the following RACF command:</p> <pre>ADDGROUP UECGRP OWNER(SYS1) OMVS(GID(5002))</pre> <p>Change the GID value 5002 to a value suitable for your local USS environment. The GID value must be unique among all group profiles.</p>
<b>Step 2</b>	<p>Add the Universal Enterprise Controller user profile <b>UECUSR</b> using the following RACF command:</p> <pre>ADDUSER UECUSR DFLTGRP(UECGRP) OWNER(SYS1) NOPASSWORD OMVS(UID(5002))</pre> <p>Change the UID value 5002 to a value suitable for your local USS environment. The value must be unique among all user profiles.</p>
<b>Step 3</b>	<p>Permit the Universal Enterprise Controller user profile READ access to the resource profiles required for enabled services. The following RACF commands permit the user profile to the resources required for all privileged services:</p> <pre>PE BPX.SUPERUSER CLASS(FACILITY) ID(UECUSR) ACCESS(READ) SETR RACLIST(FACILITY) REFRESH</pre>

### Universal Enterprise Controller Data Access

The Universal Enterprise Controller user profile **UECUSR** requires the following access to the data sets allocated in the Universal Enterprise Controller started task, **UECTLR**:

Ddname	Access	Description
STEPLIB	READ	Program library
UNVCONF	READ	Product configuration data
UNVNLS	READ	Product national language support data
UNVDB	UPDATE	Universal Enterprise Controller HFS 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

### Associate Started Tasks with User and Group Profiles

The started tasks must be associated with their user and group profiles defined above. 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.

<b>Step 1</b>	<p>Define a STARTED class profile for the Universal Enterprise Controller started procedure with the following TSO command:</p> <pre style="border: 1px solid black; padding: 5px;">RDEFINE STARTED UECTLR.* STDATA(USER(UECUSR) GROUP(UECGRP))</pre> <p>The started procedure member name used in the above command is <b>UECTLR</b>. If this has been change, the name in the REDEFINE command must also be changed to match.</p>
<b>Step 2</b>	<p>Define a STARTED class profile for the Universal Broker started procedure with the following TSO command:</p> <pre style="border: 1px solid black; padding: 5px;">RDEFINE STARTED UBROKER.* STDATA(USER(UBRUSR) GROUP(UBRGRP))</pre> <p>The started procedure member name used in the above command is <b>UBROKER</b>. If this has been change, the name in the REDEFINE command must also be changed to match.</p>
<b>Step 3</b>	<p>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.</p> <pre style="border: 1px solid black; padding: 5px;">SETROPTS RACLIST(STARTED) REFRESH</pre>

## 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.

<b>Step 1</b>	<p>Add the following entry to the <b>ICHRIN03</b> table. The table is an assembly language program that is assembled and link edited into a system LPA library.</p> <pre style="border: 1px solid black; padding: 5px;"> 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' </pre>
<b>Step 2</b>	<p>Increment the table count field by two. (The count field is the first 2 bytes of the table.)</p>
<b>Step 3</b>	<p>Assemble and link edit the ICHRIN03 table. IBM provides a sample ICHRIN03 table and the JCL to assemble and link edit it in <b>SYS1.SAMPLIB(RACTABLE)</b>.</p>
<b>Step 4</b>	<p>IPL the system with the CLPA option.</p>

## zOS Configuration - Load Library

- [Overview](#)
- [APF Authorization](#)
- [Program Control](#)
- [LNKLST](#)

### Overview

All Workload Automation programs are installed into the SUNVLOAD PDSE library. The SUNVLOAD library installation requirements are:

- Must be APF authorized.
- Must be defined to RACF Program Control.
- Specific programs or SUNVLOAD must be added to the LNKLST only if Universal Automation Center Agent is utilized.

The following sections describe the z/OS configuration steps to satisfy these requirements.

### 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 only 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 IEAAPFxx PARMLIB member is documented in the IBM MVS Initialization and Tuning Reference manual.

#### SETPROG APF Command

The SETPROG command temporarily adds a library to the APF list. Use one of the other methods to add the load library permanently to the APF list.

The SETPROG APF console command to add data set #HLQ.SUNVLOAD on volume #SMPVOL to the APF list is illustrated below. Change #HLQ to the appropriate high-level qualifier for your installation and #SMPVOL to the volume on which the library is allocated. SUNVLOAD APF authorization should be permanently established based on local site requirements.

```
SETPROG APF,ADD,DSNAME=#HLQ.SUNVLOAD,VOLUME=#SMPVOL
```

#### PROGxx PARMLIB Member

The data set can be added permanently to the APF list using the appropriate PROGxx PARMLIB member. The appropriate PROGxx member is referenced in the IEASYSxx PARMLIB member. The PROGxx member can be activated dynamically with the SET PROG=xx console command.

The APF statement of the PROGxx PARMLIB member is illustrated below. The APF statement adds data set #HLQ.SUNVLOAD on volume #SMPVOL to the APF list. Change #HLQ to the appropriate high-level qualifier for your installation and #SMPVOL to the volume on which the library is allocated.

```
APF ADD DSNAME( #HLQ.SUNVLOAD ) VOLUME( #SMPVOL )
```

### 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.

The following table 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
UAGSRV	Universal Automation Center Agent

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

1. Use of the **\_passwd** C function to authenticate user ID's
2. Use of the **setuid** C function to switch to the authenticated user ID's

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. Change #HLQ to the appropriate high-level qualifier for your installation.

```
RDEF PROGRAM ** ADDMEM(' #HLQ.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.

Some Workload Automation components 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.

## LNKLST

The following Universal Automation Center Agent programs and their aliases must be added to the LNKLST for them to be available to all batch jobs and TSO users:

- UAGRERUN (and its alias OPSRERUN) provides batch job rerun capability. The program must be available to all jobs managed by Automation Center.
- UAGCMDZ (and its alias OPSCMDZ) provides a TSO command interface to Opwise Automation Center. The TSO command interface is optional.

The programs are located in the SUNVLOAD library.

If the Universal Automation Center Agent is not used in your installation, this z/OS configuration step can be skipped.

The UAGRERUN and UAGCMDZ programs and their aliases should be added to the LNKLST using one of the following methods:

- Copy the programs and their aliases to a PDSE library that is already in the LNKLST.
- 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 LNKLSTxx PARMLIB member and perform an IPL. The LNKLSTxx PARMLIB member is documented in the IBM MVS Initialization and Tuning Reference manual.

### Copying Programs and Aliases

The UAGRERUN and UAGCMDZ and their aliases may be copied to a PDSE data set that is already in the LNKLST.



The SUNVLOAD library is a PDSE and the programs are program objects using program management features that are not supported in a PDS. For this reason, the programs must be copied to a PDSE in the LNKLIST and not to a PDS.

The JCL below executes IEBCOPY to copy the programs and their aliases to a LNKLIST library. Change #HLQ to the appropriate high-level qualifier for your installation and change the data set allocated to ddname OUT to the target PDSE in the LNKLIST. The job must end with return code 0.

```
//jobname JOB CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID,COND=(0,NE)
//STEP1 EXEC PGM=IEBCOPY,PARM='RC4NOREP'
//IN DD DISP=SHR,DSN=#HLQ.SUNVLOAD
//OUT DD DISP=SHR,DSN=lnklist.pdse
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPYGRP I=((IN,R)),O=OUT
S M=UAGRERUN
S M=UAGCMDZ
/*
```

### SETPROG LNKLIST Command

The SETPROG LNKLIST command dynamically modifies the LNKLIST. The LNKLIST changes remain in effect until the next IPL. Use one of the other methods to add the load library permanently to the LNKLIST.

LNKLIST's are managed as sets. A LNKLIST set has a name and consists of an order number of data sets or libraries. To dynamically change the current LNKLIST, a LNKLIST set must be created, updated, and activated. A newly activated LNKLIST set will be active for new address spaces, but existing address spaces are not updated.

The SETPROG LNKLIST console command is used to create, update, and activate a LNKLIST set. The SETPROG commands to add a data set to the LNKLIST are illustrated below. The commands define a LNKLIST set name *lnklistname* based on the currently active LNKLIST set. It then adds the SUNVLOAD library at the top of the data set list. Lastly, it activates the new LNKLIST set. Change #HLQ to the appropriate high-level qualifier for your installation and the LNKLIST set name to meet your local installation requirements. Ensure each command completes successfully before executing the next.

```
SETPROG LNKLIST,DEFINE,NAME=lnklistname,COPYFROM=CURRENT
SETPROG LNKLIST,ADD,NAME=lnklistname,DSNAME=#HLQ.SUNVLOAD,ATOP
SETPROG LNKLIST,ACTIVATE,NAME=lnklistname
```

### PROGxx PARMLIB Member

The SUNVLOAD library can be added permanently to the LNKLIST using the appropriate PROGxx PARMLIB member. The appropriate PROGxx member is referenced in the IEASYSxx PARMLIB member.

The LNKLIST statement of the PROGxx PARMLIB member is illustrated below. The LNKLIST statement adds data set #HLQ.SUNVLOAD to the LNKLIST set *lnklistname*. Change #HLQ to the appropriate high-level qualifier for your installation and *lnklistname* to the appropriate LNKLIST set name for your installation.

```
LNKLIST ADD NAME(lnklistname) DSNAME(#HLQ.SUNVLOAD)
```

## zOS Configuration - SMF Exits

- [Overview](#)
- [SMF Exit Introduction](#)
- [Universal Command Server](#)
  - [Configuring SMF](#)
  - [Installing SMF Exit Routines](#)
- [Universal Automation Center Agent](#)
  - [Configuring SMF](#)
  - [Installing SMF Exit Routines](#)

### Overview

SMF exits are utilized by the following Workload Automation components to provide their z/OS services:

- Universal Command Server uses one SMF exit to provide support for starting Started Tasks (STC) using Universal Command Manager. If Universal Command STC support is not required, the SMF exit does not need to be installed.
- Universal Automation Center Agent uses a number of SMF exits to provide a range of services available in Opwise Automation Center.

The following sections describe how to install the SMF exits for each component. The SMF exits are only required if the component or component services are required.

### SMF Exit Introduction

MVS System Management Facilities (SMF) collects and records system events in the form of SMF records. An SMF record is identified with a numeric record type. In addition to recording system events, SMF provides exit points in the control program from which system or application provided exit routines or programs can be called at the time work is being processed. The SMF exit routines can take appropriate actions based on the point at which the exit is called and the information provided in the SMF record. The SMF exits allow for systems and applications to monitor and augment the processing of MVS workload. For additional information on SMF, refer to the IBM manuals *MVS System Management Facilities (SMF)*, *MVS Installation Exits*, and *MVS Initialization and Tuning Reference*.

SMF configuration is performed with PARMLIB member SMFPRMxx. SMF configuration relevant for product installation is the SMF record types being recorded and the SMF exits defined. The SMFPRMxx parameters SYS and SUBSYS define these SMF recording options. The SYS parameter specifies system wide recording options for all subsystems (JES2, TSO, STC, etc.). The SUBSYS parameter specifies subsystem specific recording options. The SUBSYS parameter options override the equivalent options specified on the SYS parameter. Any SUBSYS parameter option not specified will default to the SYS parameter option.

The following example illustrates how SMF SYS and SUBSYS parameters work together.

```
SYS( EXITS(IEFU83,IEFU84) )
SUBSYS( STC, EXITS(IEFU83) )
```

The SYS parameter EXITS option defines the two SMF exits IEFU83 and IEFU84. The SUBSYS parameter for the STC subsystem also specifies the EXITS option, overriding the SYS EXITS option. The SUBSYS parameter EXIT option defines one SMF exit IEFU83 for the STC subsystem. Consequentially, only the IEFU83 exit is called for STC subsystem workload. The IEFU84 exit is not called for STC subsystem workload. For all workload other than the STC subsystem, the SMF exits IEFU83 and IEFU84 are called.

Once SMFPRMxx has defined the appropriate SMF record types to collect and the SMF exits to activate, the product SMF exit routines are installed. IBM provides the dynamic exit facility to add and remove exit routines dynamically and at IPL. IBM has defined all the SMF exits to the dynamic exit facility. The exit routines are added or removed from SMF exits using the PROGxx PARMLIB members. The PROGxx EXIT statements used to add product exit routines are listed in the appropriate sections below.

The exit names used in the dynamic exit facility are formatted as SYS<sub>ssn</sub>.exitname, where ssn is the subsystem name, or blank if it is a system wide exit, and exitname is the name of the exit, such as IEFU83. The following table provides some example dynamic exit facility exit names.

Name	Description
SYS.IEFU83	The default SMF exit IEFU83 for all system workload types.
SYSSTC.IEFU83	The SMF exit IEFU83 for STC subsystem workload.
SYSJES2.IEFU83	The SMF exit IEFU83 for JES2 subsystem workload.

### Universal Command Server

The Universal Broker STC establishes the environment to support STC execution by Universal Command Server. Part of the environment

consists of adding SMF exit routine UNVACTRT to SMF exit point SYSSTC.IEFACTRT.

The Universal Broker `UCMD_STC_SUPPORT` option specifies whether or not the environment for STC support should be established:

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

## Configuring SMF

Configuring SMF consists of defining SMF exit point SYSSTC.IEFACTRT for the STC subsystem.

The exit point is defined in the SMFPRMxx PARMLIB member with the SUBSYS STC EXITS parameter either implicitly, by excluding the EXITS parameter, or explicitly.

The following SUBSYS statement illustrates how to define SYSSTC.IEFACTRT:

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

Once the SMFPRMxx PARMLIB member has been modified, the SMFPRMxx member can be activated dynamically by restarting SMF with the following command:

```
SET SMF=xx
```

The following console command displays the active SMF options:

```
D SMF,0
```

## Installing SMF Exit Routines

The SMF exit routine UNVACTRT is provided as part of the Workload Automation 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 is permanently added to SYSSTC.IEFACTRT 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 #HLQ.SUNVLOAD load library to the SMF exit point SYSSTC.IEFACTRT. Change #HLQ to the appropriate high-level qualifier for your installation. The exit is no longer given control if it encounters an ABEND.

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

The UNVACTRT exit routine is added when the system is IPL'ed and the PROGxx member is processed. The SMF exit can be added dynamically without an IPL by activating PROGxx member with the SET console command. The following SET command activates PROGxx member:

```
SET PROG=xx
```

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.

The following console command displays the exit routines installed for exit point SYSSTC.IEFACTRT:

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

## Universal Automation Center Agent

The Universal Automation Center Agent (UAG) establishes the environment to support Opwise Automation Center services on z/OS. Part of the environment consists of establishing the following SMF exit routines:

Exit Point	Exit Routine
SYS.IEFUJI	UAGUJI
SYS.IEFUSI	UAGUSI
SYS.IEFUJV	UAGUJV
SYS.IEFU83	UAGU83
SYS.IEFU84	UAGU84
SYSSTC.IEFU83	UAGU83

## Configuring SMF

Configuring SMF consists of specifying required SMF record types to collect and the required exit points to define.

The SMF configuration must meet the following requirements:

- SMF record types 14, 15, 16, 17, 30, and 70 must be collected.
- System exit points IEFUJI, IEFUSI, IEFUJV, IEFU83, and IEFU84 must be defined.
- Assuming the Universal Command SMF exit is installed or there is already a SUBSYS EXITS option for STC, the subsystem STC exit point IEFU83 must be defined.

The record types and exit points are defined in the SMFPRMxx PARMLIB member. The following SMFPRMxx statements illustrate how to define the record types to collect and exit points.

The SYS statement uses the NOTYPE parameter to exclude specific record types. This is one of many ways the SYS statement can be specified to meet the SMF record type requirements.

```
SYS (NOTYPE(18:19,62:69,99),EXITS(IEFU83,IEFU84,IEFUJV,IEFUSI,IEFUJI,...))
SUBSYS(STC,EXITS(IEFU83,...))
```

Once the SMFPRMxx PARMLIB member has been modified, the SMFPRMxx member can be activated dynamically by restarting SMF with the following command:

```
SET SMF=xx
```

The following console command displays the active SMF options:

```
D SMF,0
```

## Installing SMF Exit Routines

The SMF exit routines are installed using the appropriate PROGxx PARMLIB member. The EXIT statements for the PROGxx PARMLIB member are illustrated below. The EXIT statements add all the exit routines from the #HLQ.SUNVLOAD load library to the appropriate exit points. Change #HLQ to the appropriate high-level qualifier for your installation.

```
EXIT ADD EXITNAME(SYS.IEFUJI) MODNAME(UAGUJI) DSNAME(#HLQ.SUNVLOAD)
EXIT ADD EXITNAME(SYS.IEFUSI) MODNAME(UAGUSI) DSNAME(#HLQ.SUNVLOAD)
EXIT ADD EXITNAME(SYS.IEFUJV) MODNAME(UAGUJV) DSNAME(#HLQ.SUNVLOAD)
EXIT ADD EXITNAME(SYS.IEFU83) MODNAME(UAGU83) DSNAME(#HLQ.SUNVLOAD)
EXIT ADD EXITNAME(SYS.IEFU84) MODNAME(UAGU84) DSNAME(#HLQ.SUNVLOAD)
EXIT ADD EXITNAME(SYSSTC.IEFU83) MODNAME(UAGU83) DSNAME(#HLQ.SUNVLOAD)
```

The exit routines are added when the system is IPL'ed and the PROGxx member is processed. The SMF exit routines can be added dynamically without an IPL by activating PROGxx member with the SET console command. The following SET command activates PROGxx member.

```
SET PROG=xx
```

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.

The following console command displays the exit routines installed for exit point SYS.IEFUJI:

```
D PROG,EXIT,EN=SYS.IEFUJI,DIAG
```

## zOS Configuration - JES SYSOUT Processing

- [Overview](#)
- [Configuration](#)
  - [Multiple SYSOUT Applications](#)
  - [UAG SYSOUT Management](#)
  - [Default Configuration](#)



### Note

The information in this section applies to UAG release 5.1.0.12 and above.

### Overview

This page describes how to configure Universal Automation Center Agent (UAG) to reliably process JES SYSOUT data sets. If UAG is not used, this information can be skipped.

The batch jobs that UAG submits create JES SYSOUT data sets. UAG processes the following SYSOUT data sets produced by the jobs:

- UAGRERUN report produced by the first step of every batch job UAG submits. The SYSOUT data set is written by step name OPSSTP00 to ddname SYSPRINT.
- JES system data sets produced by every batch job. The JES system data sets include the JESMSG LG, JESJCL and JESYSMSG ddnames.

UAG processes the SYSOUT data sets for the following purposes:

1. Retrieving the UAGRERUN report for storage in the Opwise Controller.
2. Analyzing JCL errors that occur during JCL conversion or at run-time.
3. Optionally, retrieving the JES system data sets for storage in the Opwise Controller.

JES SYSOUT data sets cannot be processed by more than one application at a time. Applications, such as spool archivers and JES offloaders, must process SYSOUT data sets one at a time. If multiple applications process SYSOUT data sets simultaneously, some SYSOUT data sets potentially will not be processed.

For UAG to reliably process SYSOUT data sets, a JES held class must be dedicated exclusively to UAG. The JES class should be defined to JES with an OUTDISP value of HOLD,HOLD for both the JOBCLASS and OUTCLASS class definitions.

### Configuration

UAG provides a flexible SYSOUT processing configuration to accommodate a variety of environments. The best configuration depends on your existing SYSOUT management practices and processes.

There are two UAG configuration options for SYSOUT processing:

- [JES\\_SYSOUT\\_CLASS](#) specifies the JES held class dedicated to UAG SYSOUT processing.
- [JES\\_SYSOUT\\_DISP](#) specifies the disposition of the SYSOUT data sets once UAG has completed SYSOUT processing.

The following sections discuss some possible UAG SYSOUT configurations. The SYSOUT options allow UAG SYSOUT processing to be customized to meet almost any SYSOUT management requirement. The following configurations are the most common.

#### Multiple SYSOUT Applications

This configuration addresses environments that have existing SYSOUT applications that process job SYSOUT data sets. These applications are often called - for example - spool archivers, output management systems, or job log archivers. They typically read all SYSOUT data sets from one or more JES held classes and archive the SYSOUT in a database. Once they have finished processing the SYSOUT in the JES spool, the SYSOUT is typically deleted from the spool.

UAG must be properly configured to work with other SYSOUT processing applications. Since SYSOUT data sets cannot be processed simultaneously by multiple SYSOUT applications, each application must process the SYSOUT data sets one at a time.

The first step is to select a JES held class to dedicate to UAG. The class cannot be shared with any other SYSOUT application. Specify the class with the UAG [JES\\_SYSOUT\\_CLASS](#) configuration option. By specifying a [JES\\_SYSOUT\\_CLASS](#) value, UAG will modify the JOB statement of the jobs it submits to JES. The JOB statement MSGCLASS parameter is changed to the [JES\\_SYSOUT\\_CLASS](#) value. If no MSGCLASS parameter is present on the JOB statement, it will be added. The JOB statement MSGCLASS parameter specifies the JES class where the job's JES system data sets are spooled.

Once UAG has completed processing the SYSOUT data sets, it performs final disposition on them as specified by the UAG [JES\\_SYSOUT\\_DISP](#) configuration option. The SYSOUT data sets must be moved to the JES class or classes used by the next SYSOUT application to process them.

There are two ways to specify the JES class to which to move the SYSOUT data sets:

- If the original JOB statement MSGCLASS value correctly specified the class, then a `JES_SYSOUT_DISP` value of **KEEP,\*** will keep the SYSOUT data sets and move them to the original MSGCLASS class. This configuration requires no JCL changes nor changes to the other SYSOUT application.
- The class can be specified directly. For example, a `JES_SYSOUT_DISP` value of **KEEP,H** will keep the SYSOUT data sets and move them to held class H.

## UAG SYSOUT Management

UAG z/OS task definitions may be defined to retrieve the job log (that is the job's JES system data sets) once the job completes processing. The job logs are transmitted to the Opwise Controller and stored in the database. The job logs can be viewed from the Opwise Controller web interface. In this configuration, once UAG has completed retrieving the job log, the SYSOUT data sets can be deleted from the JES spool. It is assumed that there are no other SYSOUT applications processing the SYSOUT data sets other than UAG.

The `JES_SYSOUT_CLASS` option is not required in this configuration. By default, UAG will process a job's SYSOUT data sets from any JES class in which it finds them.

Once UAG has completed processing the SYSOUT data sets, it performs final disposition on them as specified by the UAG `JES_SYSOUT_DISP` option. Since UAG is the one and only SYSOUT application, the SYSOUT data sets can be deleted. A `JES_SYSOUT_DISP` value of **DELETE** will remove the SYSOUT data sets from the JES spool.

## Default Configuration

The UAG default configuration is backwardly compatible with previous UAG releases. By default UAG will process SYSOUT data sets in the JES spool regardless of their disposition (held or not) or class. Consequentially, if the SYSOUT data sets are processed by another application, the results are unreliable. Once UAG completes processing of the SYSOUT data sets, it leaves them in the JES spool without changing their disposition or class.



## Workload Automation 5 Database Configuration

- [z/OS UNIX File System Introduction](#)
- [zFS Configuration](#)
- [HFS Configuration](#)
- [Mounting and Unmounting the Databases](#)
  - [Dynamic Mounts](#)
  - [Manual Mounts](#)
  - [TSO Commands](#)
  - [Console Commands](#)
  - [BPXPRMxx](#)
  - [Data Set Initialization](#)
- [Memory Management](#)

### z/OS UNIX File System Introduction

The z/OS implementation of Workload Automation 5 databases utilizes the z/OS UNIX file system. HFS or zFS data sets are used by Universal Broker and Universal Enterprise Controller started tasks.

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 one 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.

### zFS Configuration

zFS data sets are created by the installation JCL. The zFS data sets are used by Universal Broker and Universal Enterprise Controller.

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.

Workload Automation 5 uses compatibility mode aggregates only.

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 or Universal Enterprise Controller started tasks, the data sets cannot be allocated to the **UNVDB** or **UNVSPPOOL** ddnames.

The Universal Broker zFS data set names must be specified with the **UNIX\_DB\_DATA\_SET** and **UNIX\_SPOOL\_DATA\_SET** Universal Broker configuration options. The Universal Enterprise Controller zFS data set name must be specified with the **UNIX\_DB\_DATA\_SET** Universal Enterprise Controller configuration option. 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.

### HFS Configuration

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

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 sets to the ddnames.

When using HFS data sets instead of zFS data sets, uncomment the **UNVDB** and **UNVSPPOOL** ddname statements in the Universal Broker started task procedure and uncomment the **UNVDB** ddname statement in the Universal Enterprise Controller started task procedure.

The **DBHLQ** parameter in the Universal Broker started task procedure must also be uncommented.

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

## Mounting and Unmounting the Databases

When the Universal Broker and Universal Enterprise Controller started tasks are started, they check if their HFS or zFS 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 the Universal Broker `MOUNT_POINT` configuration option, 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 BPXOINIT,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 **MOUNT\_POINT\_MODE** configuration option value, 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 write the temporary cache files to the UEC database directory:

<b>Step 1</b>	Mount the <b>UECDB</b> HFS or zFS data set.
<b>Step 2</b>	Inside the mount point, create a text file named <b>DB_CONFIG</b> .

<b>Step 3</b>	Inside the <b>DB_CONFIG</b> file, add the following string: <code>set_tmp_dir *dbpath*</code> Where <b>dbpath</b> is the path to the location in which the database files reside.
<b>Step 4</b>	Start / restart UEC.

## zOS Installation - Customization

- Overview
- Universal Broker Customization
  - Universal Broker Configuration
  - Universal Broker JCL Procedure
- Universal Automation Center Agent Customization
  - Universal Automation Center Agent Configuration
- Universal Certificate Customization
  - Universal Certificate JCL Procedure
- Universal Command Manager Customization
  - Universal Command Manager Configuration
  - Universal Command Manager JCL Procedure
- Universal Command Server Customization
  - Universal Command Server Configuration
- Universal Connector Customization
  - Universal Connector Configuration
  - Universal Connector SAP RFC Configuration
  - Universal Connector JCL Procedure
- Universal Control Manager Customization
  - Universal Control Manager Configuration
  - Universal Control Manager JCL Procedure
- Universal Control Server Customization
  - Universal Control Server Configuration
- Universal Data Mover Manager Customization
  - Universal Data Mover Manager Configuration
  - Universal Data Mover Manager JCL Procedure
- Universal Data Mover Server Customization
  - Universal Data Mover Server Configuration
- Universal Enterprise Controller Customization
  - Universal Enterprise Controller Configuration
  - Universal Enterprise Controller JCL Procedure
- Universal Event Monitor Manager Customization
  - Universal Event Monitor Manager Configuration
  - Universal Event Monitor Manager JCL Procedure
- Universal Query Customization
  - Universal Query Configuration

### Overview

This page provides the following information for the customization of Workload Automation 5 components:

- Configuration
- JCL procedure

(For information on applying product licenses to installed Workload Automation 5 for z/OS components, see [z/OS Installation - Licensing](#).)

### Universal Broker Customization

#### Universal Broker Configuration

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

See the [Universal Broker 5.1.0 Reference Guide](#) for details on configuring Universal Broker.

#### Universal Broker 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 [z/OS Installation - Time Zone Environment Variable](#) for more information the TZ environment variable.

## Universal Automation Center Agent Customization

### Universal Automation Center Agent Configuration

There are two files used in defining the Universal Automation Center Agent (UAG) configuration:

- UAG component definition file
- UAG configuration file

The execution of UAG is managed by Universal Broker. The component definition file defines UAG attributes to the Broker. The Broker uses the component definition file to manage the execution of UAG. 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 **UAGCMP00**.

The UAG configuration file defines system-wide customizations for UAG 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 UAG when it is started by the Broker. The UAG component definition file defines which member of the **UNVCONF** library to use as the configuration file. The default member name is **UAGCFG00**.

See the [Universal Automation Center Agent 5.1.0 Reference Guide](#) for details on configuring UAG.

## Universal Certificate Customization

### Universal Certificate 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 [z/OS Installation - 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
```

## Universal Command Manager Customization

### Universal Command Manager Configuration

Universal Command Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is member **UCMCFG00** in the **UNVCONF** library allocated to ddname **UNVCONF** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

See the [Universal Command 5.1.0 Reference Guide](#) for details on configuring Universal Command Manager.

### Universal Command Manager 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 [z/OS Installation - 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
```

## Universal Command Server Customization

### Universal Command Server 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 5.1.0 Reference Guide](#) for details on configuring Universal Command Server.

## Universal Connector Customization

### Universal Connector Configuration

Universal Connector for z/OS uses a configuration file for system-wide customizations. The configuration file is member **USPCFG00** in the **UNVCONF** library allocated to ddname **UNVRFC** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

See the [Universal Connector 5.1.0 Reference Guide](#) for details on configuring Universal Connector.

### Universal Connector SAP RFC Configuration

Universal Connector utilizes SAP's RFC interface. The RFC interface must be configured to meet your local SAP environment. The RFC configuration file is member **USPRFC00** in the **UNVCONF** library allocated to ddname **UNVCONF** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

### Universal Connector 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 [z/OS Installation - 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
```

## Universal Control Manager Customization

### Universal Control Manager Configuration

Universal Control Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is member **UCTCFG00** in the **UNVCONF** library allocated to ddname **UNVCONF** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

See the [Workload Automation Utilities 5.1.0 Reference Guide](#) for details on configuring Universal Control Manager.

### 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 [z/OS Installation - 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
```

## Universal Control Server Customization

### Universal Control Server 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 Universal Broker, which uses the 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 Universal Broker passes the data set name to the Server when it starts the Server. 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 [Workload Automation Utilities 5.1.0 Reference Guide](#) for details on configuring Universal Control Server.

## Universal Data Mover Manager Customization



## Universal Data Mover Manager Configuration

Universal Data Mover Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is member **UDMCFG00** in the **UNVCONF** library allocated to ddname **UNVCONF** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

See the [Universal Data Mover 5.1.0 Reference Guide](#) for details on configuring Universal Data Mover.

## Universal Data Mover Manager 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 [z/OS Installation - 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
```

## Universal Data Mover Server Customization

### Universal Data Mover Server 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 Universal Broker, which uses the 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 Universal Broker passes the data set name to the Server when it starts the Server. 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 5.1.0 Reference Guide](#) for details on configuring Universal Data Mover Server.

## Universal Enterprise Controller Customization

### Universal Enterprise Controller Configuration

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

See the [Universal Enterprise Controller 5.1.0 Reference Guide](#) for details on configuring Universal Data Mover.

### Universal Enterprise Controller 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 [z/OS Installation - Time Zone Environment Variable](#) for more information the TZ environment variable.

## Universal Event Monitor Manager Customization

### Universal Event Monitor Manager Configuration

Universal Event Monitor Manager for z/OS uses a configuration file for system-wide customizations. The configuration file is member **UEMCFG00** in the **UNVCONF** library allocated to ddname **UNVCONF** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

See the [Universal Event Monitor 5.1.0 Reference Guide](#) for details on configuring Universal Event Monitor Manager.

### 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 [z/OS Installation - 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
```

## Universal Query Customization

### Universal Query Configuration

Universal Query for z/OS uses a configuration file for system-wide customizations. The configuration file is member **UQRCFG00** in the **UNVCONF** library allocated to ddname **UNVCONF** in the Universal Broker's started task JCL procedure.

Any changes to the configuration member requires the Broker's configuration cache to be refreshed by either restarting the Broker started task or using the Universal Control utility refresh option.

See the [Workload Automation Utilities 5.1.0 Reference Guide](#) for details on Universal Query configuration options.

## zOS Installation - Licensing

- Licensing Workload Automation 5 for z/OS Components
- Product License File
  - Format
  - Sample
- Entering License Information
- Restart Universal Broker

### Licensing Workload Automation 5 for z/OS Components

After Workload Automation 5 for z/OS has been installed, you must configure the following components with product licenses before they can be used:

- Universal Command Manager
- Universal Data Mover Manager
- Universal Connector
- Universal Enterprise Controller

### Product License File

For each component, product license information (license parameter keywords and their values) is contained in a separate text file provided by your Stonebranch, Inc. account representative.

#### Format

The format of the product license file name is: *<component name>\_<customer name>\_<operating system>\_<schedule or solution>.txt*. For example: **Indesca\_Stonebranch\_MVS\_A1.txt**.

(For Universal Command Manager, **Indesca** is used as the *<component name>* in the product license file name and as the name of the product in the file itself; for Universal Data Mover Manager, **Infitran** is used as the *<component name>* in the product license file name and as the name of the product in the file itself - see below.)

#### Sample

The following is a sample Workload Automation 5 for z/OS product license file (for Universal Command Manager):

```
License_Product "INDESCA"
License_Customer "STONEBRANCH"
License_OS_Type "MVS"
License_Type "PERPETUAL"
License_Expiration_Date 2029.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

### Entering License Information

In the **UNVCONF** product library:

- Enter the Universal Command Manager license parameters into the Universal Command Manager configuration file, member **UCMCFG00**.
- Enter the Universal Data Mover Manager license parameters into the Universal Data Mover Manager configuration file, member **UDMCFG00**.
- Enter the Universal Connector license parameters into the Universal Connector configuration file, member **USPCFG00**.
- Enter the Universal Enterprise Controller license parameters into the Universal Enterprise Controller configuration file, member **UECCFG00**.

It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)

## Restart Universal Broker

For Universal Broker to read the license information, you must stop and restart it.

Stop Universal Broker	<pre>STOP UBROKER</pre>
Start Universal Broker	<pre>START UBROKER[ ,UPARM='options']</pre>

## zOS Installation - Time Zone Environment Variable

- Time Zone Environment Variable
- North American Values
- European Values

### Time Zone Environment Variable

Workload Automation components 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.

## zOS Installation - TCPIP Configuration

- Overview
- Specifying TCP/IP Affinity
- TCP/IP Resource Protection
  - Stack Access Control
  - Port Access Control
  - Network Access Control
  - Socket Option Access Control

### Overview

This page describes optional Workload Automation and TCP/IP configuration topics. Whether these steps are required or desirable depends on your local TCP/IP configuration.

### Specifying TCP/IP Affinity

Workload Automation 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 Workload Automation 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 '
```

### 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 Workload Automation executes. Refer to the IBM Communications Server: IP Configuration Guide for complete details on TCP/IP resource protection. The TCP/IP resource profiles and the Workload Automation 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 Workload Automation 5 executes 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 Workload Automation executes 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 Workload Automation programs use the `SO_BROADCAST` socket option, so no user profiles require access.

## zOS Installation - SAP RFC DLL

- [Overview](#)
- [Background](#)
- [Output in the USS File System](#)
- [Workload Automation Components](#)

### Overview

The SAP RFC DLL is a remote function call library provided by SAP AG for applications to interface with the SAP system. A number of Workload Automation components make use of the SAP RFC DLL to provide SAP-related functionality.

The SAP RFC DLL interacts with the z/OS Language Environment (LE) and the z/OS UNIX System Services (USS), resulting in residual files being created by LE in the USS file system.

### Background

A z/OS user profile must have a properly defined OMVS segment in order to run a program that utilizes USS services. The OMVS segment specifies, among other attributes, a home directory in the USS file system. If no home directory is specified, it defaults to **/tmp**.

The z/OS Language Environment (LE) produces output under certain conditions. When a LE batch job or started task is executed, the LE MSGFILE option specifies the ddname to which LE output is written. By default, MSGFILE specifies the SYSOUT ddname.

Additionally, for diagnostic purposes, LE can write LE dumps and traces. As a batch job or started task, LE writes the dumps and traces to ddname CEEDUMP. When a LE program is run from the USS environment, the LE MSGFILE option defaults to standard error, and LE dumps and traces are written to a file created in the current working directory with a name starting with CEEDUMP.

The SAP RFC DLL is loaded dynamically at run time when a Workload Automation 5 batch job or started task needs to use SAP RFC functions. When the DLL is loaded, RFC initialization is performed by the DLL. As part of the RFC initialization, a USS **popen** function is called from within the DLL that creates two USS processes that run as child processes of the Workload Automation 5 program. The child processes run for a very brief amount of time (1-2 seconds).

A USS child process inherits a number of attributes from its parent, which in this case is a Workload Automation 5 program. Among the attributes inherited are the user ID profile, including the OMVS segment, as well many of the LE options specified in the batch job or started task. Among the attributes that a child process does not inherit is the ddname allocations of its parent. The child processes run in a separate address space managed by z/OS Workload Manager.

### Output in the USS File System

The two child processes created by the **popen** function executed by the SAP DLL can result in LE creating certain files in the USS file system. The LE options that the batch job or started task execute with are inherited by the child processes created by the **popen** function. When the LE options produce output, this output will be created for each of the USS child processes in the USS file system.

As an example, the LE option RPTOPTS(ON) will write a report to the location specified by the LE MSGFILE option. The report lists all the LE option values and the source of the options. The two child processes created by the SAP RFC DLL inherit the LE options, and when both of the child processes end, LE writes a options report to the location specified by the LE MSGFILE option.

Since the child processes run in a USS environment, the LE MSGFILE option defaults to standard error; however, there is no standard error defined in this case, so LE writes its output to a file named SYSOUT in the current working directory. The current working directory will be the home directory specified in the user profile OMVS segment.

There are a number of conditions that cause LE to produce output. A partial list is:

- Certain LE options produce reports to the MSGFILE location. For example, RPTOPTS and RPTSTG.
- LE error messages resulting from invalid options or run-time errors are written to the MSGFILE location.
- LE dumps and traces produced due to run-time exceptions or the LE TRACE option. Dumps and traces are written to either the CEEDUMP ddname or to a USS file name starting with CEEDUMP.

### Workload Automation Components

The following Workload Automation components use the SAP RFC DLL:

- Universal Broker
- Universal Enterprise Controller
- Universal Connector



When these components are executed with LE options that product output, a USS file named SYSOUT or a file starting with CEEDUMP will be created in the home directory of the user profile executing the component.

If two or more instances of a component executing with different user profiles share the same home directory, security violations may occur. The first component creates a USS file named SYSOUT owned by its user ID and the second component executing with a different user profile attempts to append to it resulting in a security violation.

To avoid the potential security violation when attempting to append to the LE SYSOUT file on the USS file system, define each z/OS user profile with a unique home directory.

## **zOS Installation - Configuration of zOS System SSL**

Error formatting macro: redirect: java.lang.NullPointerException

## Configuration of z/OS System SSL - Overview

- Configuration of z/OS System SSL
  - SSL Benefits
  - Required Conditions for Using SSL

### Configuration of z/OS System SSL

Workload Automation 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 the Workload Automation `SSL_IMPLEMENTATION` configuration option.

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

#### SSL Benefits

System SSL provides the following benefits:

- Utilizes any cryptographic hardware features available reducing the amount CPU resources used by Workload Automation.
- 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

#### Required Conditions for Using SSL

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

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

## Integrated Cryptographic Service Facility (ICSF)

### 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.

## Universal Broker Digital Certificate (RACF) Set-up

### 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 Workload Automation administrator. The instructions provided on this page 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.

<b>Step 1</b>	<p>Create a Certificate Authority (CA) certificate and private key using the following RACDCERT command:</p> <pre style="border: 1px solid black; padding: 10px;"> 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')</pre> <p>Change the subject and label names to meet local requirements.</p>
<b>Step 2</b>	<p>Create a certificate for the Universal Broker STC and sign it with the CA certificate created in Step 1 using the following RACDCERT command:</p> <pre style="border: 1px solid black; padding: 10px;"> 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'))</pre> <p>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.</p>
<b>Step 3</b>	<p>Create a certificate key ring for the user profile UBRUSR with the following RACDCERT command:</p> <pre style="border: 1px solid black; padding: 10px;"> RACDCERT ID(UBRUSR) ADDRING(BROKER)</pre>

<b>Step 4</b>	<p>Connect the CA certificate and the Universal Broker certificate to the key ring with the following RACDCERT command:</p> <pre>RACDCERT ID(UBRUSR) CONNECT(CERTAUTH LABEL('Company CA') + RING(BROKER) RACDCERT ID(UBRUSR) CONNECT(LABEL('Broker') RING(BROKER) DEFAULT)</pre> <p>Change the labels to match the values used in previous steps.</p>
<b>Step 5</b>	<p>If the resource profile <b>IRR.DIGTCERT.LISTRING</b> in the FACILITY class is not defined, define it with the following RDEFINE command:</p> <pre>RDEFINE FACILITY (IRR.DIGTCERT.LISTRING) UACC(NONE)</pre>
<b>Step 6</b>	<p>Permit the Broker user profile UBRUSR READ access to the RACF profile <b>IRR.DIGTCERT.LISTRING</b> in the FACILITY class using the following PERMIT command:</p> <pre>PE IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(UBRUSR) ACCESS(READ)</pre>
<b>Step 7</b>	<p>Modify the Universal Broker configuration member <b>UBRCFG00</b> as follows:</p> <pre>ssl_implementation    system saf_key_ring          BROKER</pre>
<b>Step 8</b>	<p>The CA certificate must be distributed to the remote systems from which Workload Automation 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:</p> <pre>RACDCERT CERTAUTH EXPORT (LABEL('Company CA')) + DSN(TEST.CA.CERT) FORMAT(CERTB64)</pre> <p>Change the label to match the value used in previous steps. The <b>tsoprefix.TEST.CA.CERT</b> 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.</p>

## **zOS Installation - Configuration of Security**

Error formatting macro: redirect: java.lang.NullPointerException

## **zOS Installation - Configuration of Security Overview**

### **z/OS Installation – Configuration of Workload Automation Security**

Some Workload Automation components utilize z/OS security services to control access to product functions and resources. These pages describe the installation steps to activate product security and define resource controls.



**Note**

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

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



## Configuration of Security - RACF Class

### RACF Class

All components use the Workload Automation RACF class **\$UNV**. The class name can be changed if local requirements require it.

To install the Workload Automation RACF class, perform the following steps:

<p><b>Step 1</b></p>	<p>Modify your local RACF class descriptor table by adding the following entry to <b>ICHRRCDE</b>:</p> <pre style="border: 1px solid black; padding: 10px; margin: 10px 0;"> ICHERCDE CLASS=\$UNV,           +       CASE=ASIS,                +       FIRST=ANY,                 +       OTHER=ANY,                 +       MAXLNTH=246,               +       DFTUACC=NONE,              +       ID=128,                     +       POSIT=128</pre>
<p><b>Step 2</b></p>	<p>Modify the ID and POSIT values to meet local requirements.</p> <p>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 <a href="#">Universal Command Standard I/O Access Profile</a> for details.</p> <p>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.</p> <p>Sample assemble and link edit JCL is provided in member <b>UNVRRCDE</b> in the <b>INSTALL</b> library. IBM provides a sample ICHRCDE table and the JCL to assemble and link edit it in <b>SYS1.PARMLIB(RACTABLE)</b>.</p>
<p><b>Step 3</b></p>	<p>Modify your local RACF routing table by adding the following entry to <b>ICHRFR01</b>.</p> <pre style="border: 1px solid black; padding: 10px; margin: 10px 0;"> ICHRFR01 CLASS=\$UNV,           +       ACTION=RACF</pre> <p>Sample assemble and link edit JCL is provided in member <b>UNVFR01</b> in the <b>INSTALL</b> library.</p> <p>IBM provides a sample ICHRCDE table and the JCL to assemble and link edit it in <b>SYS1.PARMLIB(RACTABLE)</b>.</p>
<p><b>Step 4</b></p>	<p>If you changed the default RACF class name <b>\$UNV</b> 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.)</p> <ol style="list-style-type: none"> <li>1. Member <b>UCMCMVS</b> in the <b>INSTALL</b> 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.</li> <li>2. Member <b>UCTCMVS</b> in the <b>INSTALL</b> 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.</li> <li>3. Member <b>UEMCMVS</b> in the <b>INSTALL</b> 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.</li> </ol> <p>All steps must end with return code 0.</p>
<p><b>Step 5</b></p>	<p>IPL the z/OS system to install the new class descriptor and routing tables.</p>

**Step 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)
```

## Configuration of Security - ACF2 Class

### ACF2 CLASS

All components use the Workload Automation ACF2 class **\$UNV**. The class name may be changed if local requirements require it.

To install the Workload Automation ACF2 class, perform the following steps:

<b>Step 1</b>	Create a CLASMAP with the following ACF2 command: <pre>SET CONTROL(GSO) INSERT CLASMAP.\$UNV RESOURCE(\$UNV) RSRCTYPE(UNV) ENTITYLN(246)</pre>
<b>Step 2</b>	Rebuild and refresh the directories with the following console command: <pre>F ACF2,REFRESH(CLASMAP),SYSID(*sysid*),CLASS(C),TYPE(GSO)</pre>
<b>Step 3</b>	Define the following resource rules for Universal Command and Universal Control with the following commands: <pre>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</pre>

## Configuration of Security - Universal Command Security

- [Overview](#)
- [Universal Command Remote Access Profile](#)
- [Universal Command Standard I/O Access Profile](#)
- [Universal Command Security Profile Definitions](#)
  - [Example 1](#)
  - [Example 2](#)
  - [Example 3](#)
  - [Example 4](#)

### Overview

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 Workload Automation. 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.Ipaddress.Pport.userid
```

The following table identifies the fields that comprise the profile name.

Field	Description
ipaddress	Numeric dotted-form IP address of the remote host as specified by the <code>REMOTE_HOST</code> 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 <b>256.10.2.123</b> is formatted as <b>256010002123</b> .
port	Numeric port number on which Universal Broker is listening as specified by the <code>REMOTE_PORT</code> option. The default Universal Broker port number is 7887. The format of the port number is a five-digit number. For example, port <b>7887</b> is formatted as <b>07887</b> .
userid	Remote user identifier with which Universal Command Manager will execute the remote command as specified by the <code>USER_ID</code> 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.

### 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, **UNVOU** 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`

The following table 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"> <li>• STDIN for the standard input file.</li> <li>• STDOUT for the standard output file.</li> <li>• STDERR for the standard error file.</li> </ul>
ipaddress	Numeric dotted-form IP address of the remote host as specified by the <code>REMOTE_HOST</code> 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 <code>256.10.2.123</code> is formatted as <code>256010002123</code> .
port	Numeric port number on which Universal Broker is listening as specified by the <code>REMOTE_PORT</code> option. The default Universal Broker port number is 7887. The format of the port number is a five-digit number. For example, port <code>7887</code> is formatted as <code>07887</code> .
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: <ul style="list-style-type: none"> <li>• A data set allocation is represented by <code>Ddsn</code>, where <code>dsn</code> is the data set name. For example, a ddname allocation of <code>PROD.APPL.DATA</code> is formatted as <code>DPROD.APPL.DATA</code>.               <ul style="list-style-type: none"> <li>• If allocating a PDS and a member name is specified, do not include the member name in the profile name, only the PDS name.</li> <li>• If allocating a relative generation data set, do not include the relative number in the profile name, only the GDG name.</li> </ul> </li> <li>• A JES SYSIN, SYSOUT or SUBSYS= allocation is represented by the value <code>SUBSYS</code>.</li> <li>• A temporary data set allocation is represented by the value <code>TEMPORARY</code>.</li> <li>• A DUMMY or NULLFILE allocation is represented by the value <code>NULLFILE</code>.</li> <li>• A USS path name allocation is represented by <code>Upath</code>, where <code>path</code> is the USS path name. For example, a ddname allocation of <code>/prod/app/data</code> is formatted as <code>U/prod/app/data</code>.               <ul style="list-style-type: none"> <li>• 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 <code>UNKNOWNUSSPATHNAME</code>.</li> <li>• USS path names are case sensitive. The Workload Automation 5 class must be defined with the <code>CASE=ASIS</code> parameter to support case sensitive profile names.</li> </ul> </li> </ul> <p>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.</p>

## Universal Command Security Profile Definitions

These examples illustrate sample RACF commands that can 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)
```

## Configuration of Security - Universal Control Security

- [Overview](#)
- [Universal Control Remote Access Profile Format](#)
- [Universal Control Security Profile Definition](#)
  - [Example 1](#)

### Overview

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 Workload Automation. 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.Iipaddress.Pport.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 <code>REMOTE_HOST</code> 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 <code>256.10.2.123</code> is formatted as <code>256010002123</code> .
port	Numeric port number on which Universal Broker is listening as identified by the <code>REMOTE_PORT</code> option. The default Universal Broker port number is 7887. The format of the port number is a five-digit number. For example, port <code>7887</code> is formatted as <code>07887</code> .
command	Universal Control command that the Manager is requesting execution. Possible command values are <code>START</code> , <code>STOP</code> , and <code>REFRESH</code> .

### Universal Control Security Profile Definition

This example illustrates sample RACF commands that can 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)
```





## Configuration of Security - Universal Event Monitor Security

- [Overview](#)
- [Universal Event Monitor Remote Access Profile Format](#)
- [Universal Event Monitor Security Profile Definition](#)
  - [Example 1](#)

### Overview

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 Workload Automation. The remote system is identified by three elements:

1. System's IP address
2. Port number that the Manager uses to connect 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.Ipaddress.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 <code>REMOTE_HOST</code> 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 <b>256.10.2.123</b> is formatted as <b>256010002123</b> .
Port	Numeric port number on which Universal Broker is listening as identified by the <code>REMOTE_PORT</code> configuration option. The default Universal Broker port number is 7887. The format of the port number is a five-digit number. For example, port <b>7887</b> is formatted as <b>07887</b> .
Userid	ID of a remote user account, which was specified by the Manager with the <code>USER_ID</code> option. If the Universal Event Monitor Server is configured to not require a user ID, the value for this field is <BLANK>. The value <code>userid</code> 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.

### Universal Event Monitor Security Profile Definition

This example illustrates sample RACF commands that can 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)
```

## zOS Installation - Performance Guidelines

- [Overview](#)
- [UNIX System Services and Language Environment](#)
- [Workload Automation Managers](#)
- [Universal Broker and Workload Automation 5 Servers](#)
- [Universal Enterprise Controller](#)

### Overview

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

Workload Automation 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 Workload Automation for z/OS components.

### UNIX System Services and Language Environment

All Workload Automation 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.

Workload Automation components do not attempt to use the USS **setpriority**, **chpriority**, or **nice** functions to adjust their performance group or service class.

### Workload Automation Managers

Workload Automation 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 Workload Automation 5 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 Workload Automation 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 Workload Automation manager.

The effect of a network time-out condition depends on whether or nor the Workload Automation 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 both of the following options:

1. Increase the manager's 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 Workload Automation 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.

### Universal Broker and Workload Automation 5 Servers

The Universal Broker started task is the center of activity on each system on which Workload Automation 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 Workload Automation servers that are performing work on z/OS on behalf of remote Workload Automation managers. Workload Automation 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.

## **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 or zFS database that is mounted in the z/OS UNIX 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.

## zOS Installation - Converting STC User Profiles to a Non-Zero UID

- [Overview](#)
- [Converting Universal Broker User Profile to Non-Zero UID](#)
- [Converting Universal Enterprise Controller \(UEC\) User Profile to Non-Zero UID](#)

### Overview

Prior to Stonebranch Solutions 4.2.0, the Universal Broker and Universal Enterprise Controller (UEC) started task user profiles were required to have an OMVS UID value of 0. As of 4.2.0, the products were enhanced to execute with a user profile defined with a non-zero UID value to improve upon the product security features.

A Workload Automation 5 installation that already has a user profile with UID 0 in use can convert the user profile from UID 0 to a non-zero UID value. There are a number of concerns when changing a user profiles UID value. The UID value identifies the user profile in the z/OS UNIX (USS) environment.

The following sections describe how to convert a Universal Broker or Universal Enterprise Controller user profile UID value from 0 to non-zero.

### Converting Universal Broker User Profile to Non-Zero UID

The conversion steps assume the following:

- The UID value is being changed from 0 to 5001. If a UID value of 5001 does not work in your local environment, change all references to 5001 in the following steps to a unique, non-zero UID value suitable for your local environment. Note that the UID value must be unique among all user profiles.
- The Universal Broker user profile name is **UBRUSR**. If the Universal Broker STC in your local environment uses a different user profile name, change all references to **UBRUSR** in the following steps to the user profile name used in your local environment.
- The user ID used to execute the commands requires an OMVS segment, and the user ID must have either UID 0 or READ access to the BPX.SUPERUSER profile in the FACILITY class.
- The Universal Broker HFS or zFS data sets must be mounted and their mount point known. The console system command **D OMVS,F** or the USS shell command **df** can be used to display all mounted USS data sets.

<b>Step 1</b>	Stop the Universal Broker STC if it is running.
<b>Step 2</b>	Change the user profile <b>UBRUSR</b> UID value to 5001 with the following command: <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre>ALTUSER UBRUSR OMVS (UID(5001))</pre> </div>

**Step 3** Permit the user profile **UBRUSR** READ access to the required resource profiles with the following commands:

```
PE BPX.SUPERUSER CLASS(FACILITY) ID(UBRUSR) ACCESS(READ)
PE BPX.JOBNAME CLASS(FACILITY) ID(UBRUSR) ACCESS(READ)
SETR RACLIST(FACILITY) REFRESH
```

The user profile **UBRUSR** should already have READ access to **BPX.DAEMON** in the FACILITY class based on Universal Broker installation requirements prior to 4.2.0. If **UBRUSR** does not have READ access to **BPX.DAEMON**, the following commands will permit appropriate access:

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

For detailed information regarding Universal Broker security requirements, see [zOS Configuration - Started Tasks](#).

**Step 4** Universal Broker databases are maintained in USS HFS or zFS data sets. The database files have an owner attribute that is based on the UID value of the Universal Broker STC user profile. The database files, the root directory, and administration files must have their owner attribute changed from UID 0 to the new non-zero UID value 5001.

By default, the Universal Broker STC will dynamically mount the USS data sets in the `/tmp` directory. Assuming the USS data set names are `UNV.UNVDB` and `UNV.UNVSPool`, their mount point would be directory `/tmp/UNV.UNVDB` and `/tmp/UNV.UNVSPool`. If the mount point is different for your local environment, adjust the commands below to the appropriate directory names.

From the z/OS UNIX shell prompt, execute the following commands:

```
su
cd /tmp/UNV.UNVDB
chown -R 5001 *
chown 5001 .
chown 5001 .initd
cd /tmp/UNV.UNVSPool
chown -R 5001 *
chown 5001 .
chown 5001 .initd
exit
```

The first command, `su`, changes to the superuser ID. The user ID used to execute the above commands will need either a UID of 0 or READ access to the **BPX.SUPERUSER** profile in the FACILITY class. If the user ID has UID 0, the `su` command is not necessary.

**Step 5** Start the Universal Broker STC.

## Converting Universal Enterprise Controller (UEC) User Profile to Non-Zero UID

The conversion steps assume the following:

- The UID value is being changed from 0 to 5002. If a UID value of 5002 does not work in your local environment, change all references to 5002 in the following steps to a unique, non-zero UID value suitable for your local environment. Note that the UID value must be unique among all user profiles.
- The UEC user profile name is **UECUSR**. If the UEC STC in your local environment uses a different user profile name, change all references to **UECUSR** in the following steps to the user profile name used in your local environment.
- The user ID used to execute the commands requires an OMVS segment, and the user ID must have either UID 0 or READ access to the **BPX.SUPERUSER** profile in the FACILITY class.
- The UEC HFS or zFS data set must be mounted and its mount point known. The console system command `D OMVS,F` or the USS shell command `df` can be used to display all mounted USS data sets.

<b>Step 1</b>	Stop the UEC STC if it is running.
<b>Step 2</b>	<p>Change the user profile UECUSR UID value to 5002 with the following command:</p> <pre>ALTUSER UECUSR OMVS(UID(5002))</pre>
<b>Step 3</b>	<p>Permit the user profile <b>UECUSR</b> READ access to the required resource profiles with the following commands:</p> <pre>PE BPX.SUPERUSER CLASS(FACILITY) ID(UECUSR) ACCESS(READ) SETR RACLIST(FACILITY) REFRESH</pre>
<b>Step 4</b>	<p>UEC databases are maintained in a USS HFS or zFS data set. The database files have an owner attribute that is based on the UID value of the UEC STC user profile. The database files, the root directory, and administration files must have their owner attribute changed from UID 0 to the new non-zero UID value 5002.</p> <p>By default, the UEC STC will dynamically mount the USS data set in the <code>/tmp</code> directory. Assuming the USS data set name is <code>UNV.UECDB</code>, its mount point would be directory <code>/tmp/UNV.UECDB</code>. If the mount point is different for your local environment, adjust the commands below to the appropriate directory names.</p> <p>From the z/OS UNIX shell prompt, execute the following commands:</p> <pre>su cd /tmp/UNV.UECDB chown -R 5002 * chown 5002 . chown 5002 .inited exit</pre> <p>The first command, <code>su</code>, changes to the superuser ID. The user ID used to execute the above commands will need either a UID of 0 or READ access to the <b>BPX.SUPERUSER</b> profile in the FACILITY class. If the user ID has UID 0, the <code>su</code> command is not necessary.</p>
<b>Step 5</b>	Start the UEC STC.

## zOS Installation - Data Set Inventory

- [Types of Data Sets](#)
- [SMP/E Data Sets](#)
- [Non-SMP/E Data Sets](#)

### Types of Data Sets

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

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

### SMP/E Data Sets

The following table lists the SMP/E data sets — and their space requirements — that are allocated and cataloged as part of the Workload Automation 5 for z/OS installation.

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

Data Set Name	Space (tracks)	Description
UNV.AUNVLOAD	6500	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	60	SMP/E distribution library for configuration template files
UNV.SUNVLOAD	6500	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	60	SMP/E target library for configuration template file.
UNV.SMP.CSI		SMP/E CSI VSAM cluster for Workload Automation 5 SMP/E zones.
UNV.SMP.CSI.DATA	75	SMP/E CSI VSAM data for Workload Automation 5 SMP/E zones.
UNV.SMP.CSI.INDEX	15	SMP/E CSI VSAM index for Workload Automation 5 SMP/E zones.
UNV.SMP.SMPLOG	30	SMP/E log file.



<b>UNV.SMP.SMPLOGA</b>	30	SMP/E backup log file.
<b>UNV.SMP.SMPLTS</b>	100	SMP/E target library for base versions of load modules using a SYSLIB allocation.
<b>UNV.SMP.SMPMTS</b>	30	SMP/E target library for macros existing only in the distribution libraries.
<b>UNV.SMP.SMPPTS</b>	4500	SMP/E temporary SYSMOD storage.
<b>UNV.SMP.SMPSCDS</b>	30	SMP/E zone backup library.
<b>UNV.SMP.SMPSTS</b>	30	SMP/E target library for source existing only in the distribution libraries.

## Non-SMP/E Data Sets

The following table lists the non-SMP/E data sets – and their space requirements – that are allocated and cataloged as part of the Workload Automation 5 for z/OS package installation.

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

<b>Data Set Name</b>	<b>Space (tracks)</b>	<b>Description</b>
<b>UNV.UECDB</b>	4500	Universal Enterprise Controller HFS or zFS databases.
<b>UNV.MDL</b>	1	Universal Broker sequential trace data set allocation model.
<b>UNV.UNVCREF</b>	75	Universal Command Server command reference library.
<b>UNV.UNVDB</b>	150	Universal Broker HFS or zFS database.
<b>UNV.UCRDB</b>	15	Universal Certificate database.
<b>UNV.UNVSPool</b>	3000	Workload Automation 5 HFS or zFS spool database.
<b>UNV.UNVTRACE</b>	150	Universal Broker PDS/E trace data set.
<b>UNV.UAG</b>	1	Universal Automation Center Agent logging data set allocation model.
<b>UNV.V5R1M0.INSTALL</b>	30	Workload Automation 5 package installation and maintenance JCL.
<b>UNV.UNVCONF</b>	15	Workload Automation 5 configuration library.

<b>UNV.UNVCOMP</b>	15	Workload Automation 5 component definition library.
--------------------	----	---

## **zOS USS Installation**

Error formatting macro: redirect: java.lang.NullPointerException

## z/OS USS Installation - Overview

- [z/OS USS Installation](#)
- [Installation Procedures](#)

### z/OS USS Installation

These pages describe the installation of Stonebranch, Inc.'s Workload Automation 5 for z/OS UNIX System Services (USS) on a z/OS operating system. Unless otherwise specified, all references to Workload Automation for z/OS USS refer to version 5.1.0.



**Note**

Starting with the 3.2.0 release of Universal Products, a Universal Broker must run on all systems on which a Workload Automation 5 component is running, including manager components. The Broker maintains product configuration data for all components that have a configuration file.

### Installation Procedures

All Workload Automation for z/OS USS components are provided in the Workload Automation for z/OS SMP/E package.

The z/OS components in that package must be installed before you can install the z/OS USS components. (See [z/OS Installation Procedures](#) for information on completing the z/OS installation.)

After the z/OS components have been installed, you can install the z/OS USS components (see [SMPE Installation](#)).

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**

Please read the following sections before installing Workload Automation for z/OS USS:

- [z/OS USS Installation - Installation Requirements](#)
- [z/OS USS Installation - Installation Upgrades](#)

## zOS USS Installation - Components

- [Workload Automation 5 for z/OS USS Components](#)
- [Component Compatibility](#)

### Workload Automation 5 for z/OS USS Components

Workload Automation 5 for z/OS USS contains the following Workload Automation components:

- Universal Command Manager 5.1.0
- Universal Control Manager 5.1.0
- Universal Data Mover Manager 5.1.0
- Universal Encrypt 5.1.0
- Universal Event Monitor Manager 5.1.0
- Universal Message to Exit Code Translator 5.1.0
- Universal Query 5.1.0
- Universal Certificate 5.1.0
- Universal WTO 5.1.0
- Universal Copy 5.1.0

### Component Compatibility

The following table identifies the compatibility of Workload Automation 5 for z/OS USS components with previous component / product versions.

Component	Compatibility
Universal Command 5.1.0	Universal Command 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Control 5.1.0	Universal Control 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Data Mover 5.1.0	Universal Data Mover 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, and 3.1.0.
Universal Encrypt 5.1.0	Universal Encrypt 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Query 5.1.0	Universal Broker 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Event Monitor 5.1.0	Universal Event Monitor 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, and 3.1.0.

The component references pertain to all supported platforms for that version.

## zOS USS Installation - Installation Requirements

- [Overview](#)
- [Data Set Space Requirements](#)
- [SMP/E](#)
- [Platform Requirements](#)

### Overview

Workload Automation for z/OS USS require the following software releases:

- z/OS 1.10 or above.
- IBM Language Environment for z/OS 1.10 or above.
- IBM Communication Server for z/OS 1.10 or above.
- SMP/E 3.5 or above.
- Minimum 200 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 Workload Automation for z/OS USS components are provided in the Workload Automation for z/OS SMP/E package. The z/OS components must be installed before the z/OS USS components. See [z/OS Installation](#) for information on completing the z/OS installation.

### Data Set Space Requirements

As part of the Workload Automation 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 [z/OS USS Installation - Data Set Inventory](#).

### SMP/E

Workload Automation for z/OS USS are installed using SMP/E. The components are installed in the Workload Automation SMP/E CSI.

The following table identifies the SMP/E FMIDs for the Workload Automation for z/OS USS components.

Product	FMID	SMP/E Requisites
USS Universal Common 5.1.0	UUNV510	No prerequisites. Supersedes and deletes FMID UUNV320, UUNV410, UUNV420, and UUNV430.
USS Universal Broker Utilities 5.1.0	UUBR510	UUNV510 and TUBR510 are prerequisites. Supersedes and deletes FMID UUBR320, UUBR410, UUBR420, and UUBR430.
USS Universal Command Manager 5.1.0	UUCM510	UUNV510 and TUBR510 are prerequisites. Supersedes and deletes FMID UUCM320, UUCM410, UUCM420, and UUCM430.
USS Universal Data Mover Manager 5.1.0	UUDM510	UUNV510 and TUBR510 are prerequisites. Supersedes and deletes FMID UUDM320, UUDM410, UUDM420, and UUDM430.

USS Universal Utilities 5.1.0	UUTL510	UUNV510 is a prerequisite. Supersedes and deletes FMID UUTL320, UUTL410, UUTL420, and UUTL430.
USS Universal Event Monitor Manager 5.1.0	UUEM510	UUNV510 and TUBR510 are prerequisites. Supersedes and deletes FMID UUEM320, UUEM410, UUEM420, and UUEM430.

## Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opwise Automation Center 5.1.1](#) and [Indesca-Infitran 5.1.0](#) page to make sure that your platform is supported before performing an installation.

## zOS USS Installation - Installation Upgrades

### Overview

This page describes changes in the z/OS USS installation that have occurred with new versions of the package. If a particular version is absent from the list, no change occurred.

### Universal Products for z/OS USS 3.2.0

Universal Products for z/OS USS 3.2.0 was 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.

### Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opwise Automation Center 5.1.1](#) and [Indesca-Infitran 5.1.0](#) page to make sure that your platform is supported before upgrading.



## **zOS USS Installation - Distribution File**

### **Workload Automation for z/OS USS – Product Distribution File**

The Workload Automation for z/OS USS installation files are distributed with the Workload Automation for z/OS distribution file.

See [z/OS Installation](#) for information on downloading the Workload Automation for z/OS distribution file.

## **zOS USS Installation - SMPE Installation**

Error formatting macro: redirect: java.lang.NullPointerException

## zOS USS SMPE Installation - Overview

### Workload Automation for z/OS USS – SMP/E Installation

The installation steps describe how to perform the SMP/E install of Workload Automation 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.

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

1. Installing any Workload Automation package from Stonebranch, Inc. for the first time, or installing a Workload Automation for z/OS USS package in a new SMP/E CSI.  
See [z/OS USS Installation - New Install, New CSI](#) for installation instructions.
2. Upgrading a Stonebranch Solutions 4.x for z/OS USS package install. In this case, the Workload Automation for z/OS USS 5.1.0 package is installed into an SMP/E CSI that contains a Stonebranch Solutions 4.x package.  
See [z/OS USS Installation - 4.x Upgrade, Existing CSI](#) for installation instructions.
3. Upgrading a Universal Products 3.2.0 for z/OS USS package install. In this case, the Workload Automation for z/OS USS 5.1.0 package is installed into an SMP/E CSI that contains a Universal Products 3.2.0 package.  
See [z/OS USS Installation - 3.2.0 Upgrade, Existing CSI](#) for installation instructions.

## zOS USS Installation - New Install, New CSI

### New Install, New CSI

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

Use this installation process for either of these environments:

- Installing a Workload Automation for z/OS USS package for the first time.
- Installing a Workload Automation for z/OS USS package in a different SMP/E CSI than other Workload Automation for z/OS USS components.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Workload Automation **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.

<b>Step 1</b>	<p>Submit the JCL in member <b>UNVUN01</b>. 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.</p> <p>Step <b>ALLOCUSS</b> creates the z/OS UNIX directories required by the USS components. The path prefix where the directories are created was defined when the <b>#SETUP</b> member of the <b>INSTALL</b> library was modified and run as part of the z/OS installation. IBM recommends the path prefix to be <b>/usr/lpp</b>. Review the path prefix used in the <b>ALLOCUSS</b> step PARM value to be sure it meets local requirements.</p> <p>The z/OS UNIX directory where the Workload Automation directories are created must be mounted in read/write mode. The user ID used to run this job must have write access to the directory.</p> <p>If the z/OS UNIX directories already have been created, the <b>ALLOCUSS</b> step will indicate this in its report and end successfully.</p>
<b>Step 2</b>	<p>Submit the JCL in member <b>UNVUN02</b>. 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.</p>
<b>Step 3</b>	<p>Submit the JCL in member <b>UNVUN03</b>. The JCL performs a SMP/E APPLY of the USS product FMIDs and any received PTFs. Step <b>APYFMID</b> must end with a condition code of 0.</p> <p>Step <b>APYPTFS</b> is considered successful under any of the following conditions:</p> <ul style="list-style-type: none"> <li>• Step ends with condition code 0.</li> <li>• Step ends with condition code 4, and message GIM42001W is written in ddname SMPDOUT.</li> <li>• Step ends with condition code 12, and message GIM24801S is written in ddname SMPDOUT.</li> </ul> <p>The user ID used to run the job must have READ access to <b>BPX.FILEATTR.APF</b> profile in the <b>FACILITY</b> 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.</p>
<b>Step 4</b>	<p>Submit the JCL in member <b>UNVUN04</b>. The JCL member performs a SMP/E ACCEPT of the USS product FMIDs and any applied PTFs. Step <b>ACCFMID</b> must end with a condition code of 0.</p> <p>Step <b>ACCPYPTFS</b> is considered successful under any of the following conditions:</p> <ul style="list-style-type: none"> <li>• Step ends with condition code 0 or 4.</li> <li>• Step ends with condition code 12, and message GIM24801S is written in ddname SMPDOUT.</li> </ul>
<b>Step 5</b>	<p>Submit the JCL in member <b>UNVUN05</b>. The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.</p>

## zOS USS Installation - 4.x Upgrade, Existing CSI

### Stonebranch Solutions for z/OS USS 4.x Upgrade, Existing CSI

The Stonebranch Solutions 4.x for z/OS USS Upgrade, Existing CSI installation process describes how to upgrade an existing Stonebranch Solutions 4.x package in an existing SMP/E CSI.



#### Note

The Stonebranch Solutions 4.x for z/OS package first must be upgraded to Workload Automation 5 before the Stonebranch Solutions 4.x for z/OS USS package is upgraded.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Workload Automation 5 **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.

<b>Step 1</b>	Submit the JCL in member <b>UNVUN02</b> . 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.
<b>Step 2</b>	Submit the JCL in member <b>UNVUN03</b> . 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: <ul style="list-style-type: none"> <li>• Step ends with condition code 0.</li> <li>• Step ends with condition code 4, and message GIM42001W is written in ddname <b>SMPOUT</b>.</li> <li>• Step ends with condition code 12, and message GIM24801S is written in ddname <b>SMPOUT</b>. The user ID used to run the job must have READ access to <b>BPX.FILEATTR.APF</b> profile in the <b>FACILITY</b> 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.</li> </ul>
<b>Step 3</b>	Submit the JCL in member <b>UNVUN04</b> . 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: <ul style="list-style-type: none"> <li>• Step ends with condition code 0 or 4.</li> <li>• Step ends with condition code 12, and message GIM24801S is written in ddname <b>SMPOUT</b>.</li> </ul>
<b>Step 4</b>	Submit the JCL in member <b>UNVUN05</b> . The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

## zOS USS Installation - 3.2.0 Upgrade, Existing CSI

### 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.



#### Note

The Universal Products 3.2.0 for z/OS package must be upgraded first to Workload Automation 5 before the Universal Products 3.2.0 for z/OS USS package is upgraded.

The installation JCL referenced by the installation steps is created by the **#SETUP** member in the Workload Automation 5 **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.

<b>Step 1</b>	Submit the JCL in member <b>UNVUN02</b> . 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.
<b>Step 2</b>	Submit the JCL in member <b>UNVUN03</b> . 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: <ul style="list-style-type: none"> <li>• Step ends with condition code 0.</li> <li>• Step ends with condition code 4, and message GIM42001W is written in ddname <b>SMPOUT</b>.</li> <li>• Step ends with condition code 12, and message GIM24801S is written in ddname <b>SMPOUT</b>. The user ID used to run the job must have READ access to <b>BPX.FILEATTR.APF</b> profile in the <b>FACILITY</b> 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.</li> </ul>
<b>Step 3</b>	Submit the JCL in member <b>UNVUN04</b> . 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: <ul style="list-style-type: none"> <li>• Step ends with condition code 0 or 4.</li> <li>• Step ends with condition code 12, and message GIM24801S is written in ddname <b>SMPOUT</b>.</li> </ul>
<b>Step 4</b>	Submit the JCL in member <b>UNVUN05</b> . The JCL copies sample configuration members to the configuration library. All steps must end with a return code 0.

## z/OS USS Installation - Customization

- Overview
- Universal Command Manager for z/OS USS Customization
  - Configuration
- Universal Control Manager for z/OS USS Customization
  - Configuration
- Universal Data Mover Manager for z/OS USS Customization
  - Configuration
- Universal Event Monitor Manager for z/OS USS Customization
  - Configuration
- Universal Query for z/OS USS Customization
  - Configuration

### Overview

The product executable files intended for command line use are located in the directory **#USSPRE/universal/bin**, where **#USSPRE** 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.

(For information on applying product licenses to installed Workload Automation 5 for z/OS USS components, see [z/OS USS Installation - Licensing](#).)

### 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 5.1.0 Reference Guide](#) for details on configuring Universal Command Manager.

### 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 [Workload Automation Utilities 5.1.0 Reference Guide](#) for details on configuring Universal Control Manager.

### 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 5.1.0 Reference Guide](#) for details on configuring Universal Data Mover.

### 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 5.1.0 Reference Guide](#) for details on configuring Universal Event Monitor Manager.

## 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 [Universal Query](#) for details on configuring Universal Query.



## zOS USS Installation - Licensing

- Licensing Workload Automation 5 for z/OS USS Components
- Product License File
  - Format
  - Sample
- Entering License Information
- Restart Universal Broker

### Licensing Workload Automation 5 for z/OS USS Components

After Workload Automation 5 for z/OS USS has been installed, you must configure the following components with product licenses before they can be used:

- Universal Command Manager
- Universal Data Mover Manager

### Product License File

For each component, product license information (license parameter keywords and their values) is contained in a separate text file provided by your Stonebranch, Inc. account representative.



#### Note

Product license information for Workload Automation 5 for z/OS USS components are contained in the Workload Automation 5 for z/OS product license files.

### Format

The format of the product license file name is: *<component name>\_<customer name>\_<operating system>\_<schedule or solution>.txt*. For example: **Indesca\_Stonebranch\_MVS\_A1.txt**.

(For Universal Command Manager, **Indesca** is used as the *<component name>* in the product license file name and as the name of the product in the file itself; for Universal Data Mover Manager, **Infitran** is used as the *<component name>* in the product license file name and as the name of the product in the file itself - see below.)

### Sample

The following is a sample Workload Automation 5 for z/OS product license file (for Universal Command Manager):

```
License_Product "INDESCA"
License_Customer "STONEBRANCH"
License_OS_Type "MVS"
License_Type "PERPETUAL"
License_Expiration_Date 2029.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

### Entering License Information

In the **UNVCONF** product library:

- Enter the Universal Command Manager license parameters into the Universal Command Manager configuration file, member **UCMCFGU0**.
- Enter the Universal Data Mover Manager license parameters into the Universal Data Mover Manager configuration file, member

**UDMCFGU0.**

It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)

**Restart Universal Broker**

For Universal Broker to read the license information, you must stop and restart it.

Stop Universal Broker	<pre>STOP UBROKER</pre>
Start Universal Broker	<pre>START UBROKER[ ,UPARM='options' ]</pre>

## z/OS USS Installation - Data Set Inventory

### SMP/E Data Sets

A number of SMP/E data sets are allocated and cataloged as part of the Workload Automation for z/OS USS installation.

The following table 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.

### z/OS UNIX Directories

A number of z/OS UNIX directories are created as part of the Workload Automation for z/OS USS installation.

The following table lists the directories and their space requirements.

Directory Name	Space (MB)	Description
universal/bin/IBM	80	SMP/E target directory for the SUNVHBIN DDDEF.
universal/bin	0	USS Workload Automation 5 program files.
universal/nls/IBM	0.5	SMP/E target directory for the SUNVHMLS DDDEF.
universal/nls	0	USS Workload Automation 5 national language support files.
universal/ucmdmgr	0	Universal Command Manager installation directory.
universal/uctlmgr	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.

## Windows Installation

Error formatting macro: redirect: java.lang.NullPointerException

## Windows Installation - Overview

- Windows Installation
- Installation Packages
  - Distribution File Format
- Pre-Installation Guidelines
  - Installation Paths
- Installation Summary
- Windows Installer
  - Windows Installer Interfaces
  - Installing Windows Installer with a Workload Automation Package
  - Installing Windows Installer Separately from a Workload Automation 5 Package

## Windows Installation

These pages provide information on the installation of Stonebranch, Inc.'s Workload Automation 5 on Microsoft Windows operating systems. Unless otherwise specified, all references to Workload Automation for Windows refer to version 5.1.0.

## Installation Packages

Workload Automation 5 is installed with one required installation package and three optional installation packages.

To obtain a Workload Automation package, you must download the corresponding product distribution file from the Stonebranch [Customer Portal](#).



### Note

A customer user name and password – provided by Stonebranch, Inc. – are required to access the Customer Portal.

The following table lists the packages available, their supported platforms, distribution file names, and whether or not the package is required.

Package	Distribution File	Required
Workload Automation 5	sb-5.1.0. <i>LEVEL</i> -windows-i386.exe (32-bit) <b>OR</b> sb-5.1.0. <i>LEVEL</i> -windows-x64.exe (64-bit)	Yes
Universal Enterprise Controller	sb-UECtrlr-5.1.0. <i>LEVEL</i> -windows-i386.exe	No
Universal Enterprise Controller Clients	sb-UEClient-5.1.0. <i>LEVEL</i> -windows-i386.exe	No
Workload Automation 5 for SOA	sb-soa-5.1.0. <i>LEVEL</i> -windows-i386.exe	No

The word *LEVEL* in each distribution file name is a numeric value representing the product maintenance level contained in the distribution file.

## Distribution File Format

The product distribution files are in a Windows self-extracting archive file format.

The names of the distribution files have the following format:

*PROD-VRML-OS-HW.exe*

<i>PROD</i>	Package content
<i>VRML</i>	Version, Release, Modification level, and Maintenance level (for example, 5.1.0.3)
<i>OS</i>	Operating system supported
<i>HW</i>	Hardware type supported

## Pre-Installation Guidelines

Before starting any Workload Automation for Windows installation, it is recommended that you close all other Windows programs. Active

programs may be using system files that the Workload Automation 5 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.



You must be able to write to the directory from which the installation is launched.

## Installation Paths

### 32-Bit Systems

On 32-bit Windows systems, the default installation path is:

- "C:\Program Files"

### 64-Bit Systems

On 64-bit Windows systems, the default installation paths are:

- "C:\Program Files (x86)" (for the 32-bit Workload Automation for Windows package)
- "C:\Program Files" (for the 64-bit Workload Automation for Windows package)

## Installation Summary

<b>Step 1</b>	Download the distribution file from the Stonebranch <a href="#">Customer Portal</a> .
<b>Step 2</b>	Log on to Windows using an account with the privileges noted above.
<b>Step 3</b>	Exit all running programs to reduce the likelihood of a reboot.
<b>Step 4</b>	Execute the installation file.
<b>Step 5</b>	Follow installation instructions presented by the graphical installation program.

## Windows Installer

The installation of each Workload Automation for Windows package requires Windows Installer (**msiexec.exe**) 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 Workload Automation for Windows package provides version **3.1.4000.1823** of Windows Installer.

### Windows Installer Interfaces

Windows Installer provides two interfaces that you can use to install Workload Automation: graphical and command line.

### Installing Windows Installer with a Workload Automation Package

If Windows Installer is not present on your Windows operating system, the Workload Automation package being installed will install it before any of its own files are copied.

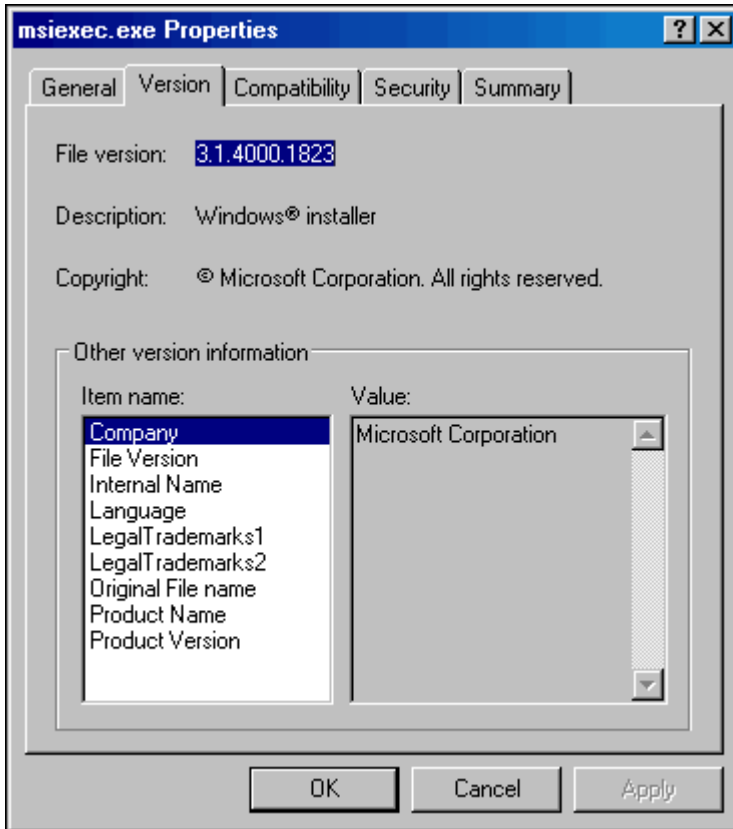
If Windows Installer is present on your Windows operating system, but it is a version prior to **3.1.4000.1823**, the installation will upgrade it.

### Determining if Windows Installer will be Installed or Upgraded

To determine if the installation of a Workload Automation package also will install or upgrade Windows Installer, perform the following steps:

<b>Step 1</b>	From Windows Explorer, go to the 32-bit Windows system directory (for example, <b>\\Windows\System32</b> , <b>\\Windows\SysWow64</b> ).
---------------	---

- Step 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 Workload Automation 5 installation will install it.
- Step 3** Right-click on the **msiexec.exe** file name to display a pop-up menu.
- Step 4** Click **Properties** to display the Properties dialog for **msiexec.exe**.
- Step 5** Click the **Version** tab. File Version: identifies the currently installed version of Windows Installer (see the following figure). If the version is prior to **3.1.4000.1823**, the package installation will upgrade it.



(In this figure, 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 Workload Automation 5 Package

The Windows Installer version provided with each Workload Automation 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:

<b>Step 1</b>	Execute the package distribution file downloaded from the Stonebranch [Customer Portal <a href="https://stonebranch.zendesk.com/hc/en-us">https://stonebranch.zendesk.com/hc/en-us</a> ]. Make note of the directory into which the installation files are extracted. After extraction of the files is complete, the installation will begin.
<b>Step 2</b>	Cancel the installation.
<b>Step 3</b>	Go to the directory where the extracted files reside.
<b>Step 4</b>	Execute <b>WindowsInstaller-KB893803-x86.exe</b> . <ul style="list-style-type: none"><li>• If Windows Installer was upgraded, reboot the system.</li><li>• If Windows Installer was installed new, no reboot is required.</li></ul>



## Windows Installation - Licensing

- Licensing Workload Automation 5 for Windows Components
- Product License File
  - Format
  - Sample
- Entering License Information
  - Licensing via Universal Configuration Manager
- Restart Universal Broker
  - Via Windows Services
  - Via Universal Configuration Manager

### Licensing Workload Automation 5 for Windows Components

After Workload Automation 5 for Windows has been installed, you must configure the following components with product licenses before they can be used:

- Universal Command Manager
- Universal Data Mover Manager
- Universal Connector
- Universal Event Monitor Server
- Universal Enterprise Controller
- Universal Application Container Server



#### Note

Universal Enterprise Controller 5.1.0 for Windows and Universal Application Container Server (as a component of Universal Command Agent for SOA 5.1.0 for Windows) are packaged, and licensed, separately.

### Product License File

For each component, product license information (license parameter keywords and their values) is contained in a separate text file provided by your Stonebranch, Inc. account representative.

#### Format

The format of the product license file name is: *<component name>\_<customer name>\_<operating system>\_<schedule or solution>.txt*. For example: **Indesca\_Stonebranch\_NT\_A1.txt**.

(For Universal Command Manager, **Indesca** is used as the *<component name>* in the product license file name and as the name of the product in the product license file itself; for Universal Data Mover Manager, **Inftran** is used as the *<component name>* in the product license file name and as the name of the product in the product license file itself - see below.)

#### Sample

The following is a sample Universal Command Manager for Windows product license file:

```
License_Product "INDESCA"
License_Customer "STONEBRANCH"
License_OS_Type "NT"
License_Type "PERPETUAL"
License_Expiration_Date 2029.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

## Entering License Information

Workload Automation 5 for Windows components can be configured with product licenses either by:

1. Entering the information into their configuration file. It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)
  - Universal Command Manager: **ucmd.conf**
  - Universal Data Mover Manager: **udm.conf**
  - Universal Connector: **usap.conf**
  - Universal Event Monitor Server: **uems.conf**
  - Universal Enterprise Controller: **uec.conf**
  - Universal Application Container Server: **uacs.conf**
2. Specifying the information on the Universal Configuration Manager License Information page for that component, either by.
  - a. Entering the information specified in the license file.
  - b. Importing the license file.



**Note**

Universal Configuration Manager is installed during the Workload Automation 5.1.0 for Windows installation.

## Licensing via Universal Configuration Manager

To enter license information via the Universal Configuration Manager, perform the following steps:

<b>Step 1</b>	On the Windows Control Panel, double-click the Universal Configuration Manager icon to display the Universal Configuration Manager screen.
<b>Step 2</b>	In the Installed Components tree, click + next to the component that you want to configure with license information.
<b>Step 3</b>	Click License Information to display the License Information page.
<b>Step 4</b>	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.

<b>Step 5</b>	If you want to import the license file: <ol style="list-style-type: none"><li>1. Click the Import button.</li><li>2. On the Select License File dialog that displays, select the license file to import.</li><li>3. Click the Open button. The information is imported to the License information page.</li></ol>
<b>Step 6</b>	Click the OK button to save the license information.

## Restart Universal Broker

For Universal Broker to read the license information, you must start / restart it.

### Via Windows Services

<b>Step 1</b>	Go to Windows Services.
<b>Step 2</b>	Locate and select Universal Broker.
<b>Step 3</b>	If Universal Broker is not running, click <b>Start</b> .  If Universal Broker is running, click <b>Restart</b> .

### Via Universal Configuration Manager

If you enter license information via the Universal Configuration Manager (see [above](#)), Universal Broker is restarted automatically, and the license information is read, when you click OK to save the license information.

## Workload Automation 5 for Windows

Error formatting macro: redirect: java.lang.NullPointerException

## Workload Automation 5 for Windows - Overview

### Overview

The following information is provided for the installation of Workload Automation 5 for Windows:

- [Workload Automation 5 for Windows - Installation Package](#)
- [Workload Automation 5 for Windows - Installation Requirements](#)
- [Workload Automation 5 for Windows - Installation Procedures](#)
- [32-Bit Workload Automation 5 for Windows on 64-Bit Windows Systems](#)
- [Migrating an Opwise Agent to UAG for Workload Automation 5 for Windows](#)
- [Workload Automation 5 for Windows - File Inventory Lists](#)

(For licensing information, see [Windows Installation - Licensing](#).)

## Workload Automation 5 for Windows - Installation Package

### Components

The Workload Automation 5.1.0 for Windows package includes the following product components:

- Universal Broker 5.1.0
- Universal Automation Center Agent 5.1.0
- Universal Certificate 5.1.0
- Universal Command Manager and Server 5.1.0
- Universal Connector 5.1.0
- Universal Control Manager and Server 5.1.0
- Universal Data Mover Manager and Server 5.1.0
- Universal Encrypt 5.1.0
- Universal Event Log Dump 5.1.0
- Universal Event Monitor Manager and Server 5.1.0
- Universal Message to Exit Code Translator 5.1.0
- Universal Query 5.1.0



#### Note

Universal Enterprise Controller 5.1.0, Universal Enterprise Controller 5.1.0 Client Applications, and Workload Automation 5 for SOA 5.1.0 are packaged separately (see [Universal Enterprise Controller for Windows - Package](#), [UEC Client Applications - Package](#), and [Workload Automation 5 for SOA for Windows - Package](#)).

### Component Compatibility

The following table identifies the compatibility of Workload Automation 5 for Windows components with previous component / product versions.

Component	Compatibility
Universal Broker 5.1.0	Stonebranch Solutions / Universal Products releases 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0.
Universal Automation Center Agent 5.1.0	Universal Automation Center Agent is compatible with Automation Center 5.1.0 and above.
Universal Command 5.1.0	Universal Command 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0.
Universal Control 5.1.0	Universal Control 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0.
Universal Data Mover 5.1.0	Universal Data Mover 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1 and 3.1.0.
Universal Encrypt 5.1.0	Universal Encrypt 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0.
Universal Query 5.1.0	Universal Broker 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, 2.1.0, and 1.2.0.
Universal Event Monitor 5.1.0	Universal Event Monitor 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, and 3.1.0.

The component references pertain to all supported platforms for that version.

## Workload Automation 5 for Windows - Installation Requirements

- System Requirements
- Platform Requirements
- Installation Account
- Universal Broker Service
  - UBrokerService: Default Universal Broker Service Account
  - Using a Windows Domain Account to Execute Universal Broker
- Spool Directory
  - Universal Command Server
  - UAG Cache
- UAG Crossgrade

### System Requirements

- One of the supported Windows operating systems. Currently, the following Windows operating systems are supported by Workload Automation 5:
  - Windows Server 2003 SP1 and higher
  - Windows Server 2003 R2
  - Windows XP SP3
  - Windows Vista
  - Windows 7
  - Windows Server 2008
  - Windows Server 2008 R2
  - Windows Server 2012
- An account with administrative privileges.
- Possible reboot: a reboot is required if the Windows Installer service is not installed, a version of the Windows Installer prior to 3.1.4000.1823 is installed, or if required files are in use at the time of the installation.
- TCP/IP.
- About 75 megabytes of disk space.



#### Note

The Workload Automation 5 (x64) for Windows package requires a 64-bit edition of the Windows versions listed above, running on an Intel-based system. For a 64-bit Itanium-based (IA64) system, the 32-bit package must be used.

### Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opwise Automation Center 5.1.1](#) and [Indesca-Infitran 5.1.0](#) page to make sure that your platform is supported before performing an installation.

### Installation Account

The account used to install the package must have administrative privileges.

### Universal Broker Service

The Universal Broker service runs either with an Administrative account or with the Local System account.

If you are using an Administrative account, the account must have the following privileges:

- Act as part of the operating system
- Adjust memory quotas for a process
- Bypass traverse checking
- Debug programs
- Deny access to this computer from the network
- Deny log on as batch job
- Deny log on locally
- Deny log on through Terminal Services
- Log on as a service
- Impersonate a client after authentication
- Increase scheduling priority
- Replace a process level token
- Take ownership of files and other objects

### UBrokerService: Default Universal Broker Service Account

The Workload Automation 5 for Windows installation will create an account with the privileges listed above, if the account does not already exist. The name of this account is **UBrokerService**. A default password provided for the account can be obtained by contacting Stonebranch support. We strongly suggest that you set this password at install time via the installation dialogs or via the **BROKERPWD** command line option.



**Note**

If the **UBrokerService** account already exists, its password cannot be changed from the install.

See [Installing Workload Automation 5 via the Graphical Interface](#) for detailed information about setting up the Universal Broker service account at install time.

## Using a Windows Domain Account to Execute Universal Broker

The Universal Broker Service may be configured to execute with an existing Windows domain account, provided that account has been granted the privileges listed in this section, above, on the local system and the account has the file system permissions described in [Service Security](#).

## Spool Directory

The spool directory is used to store the following types of information:

- Execution information for Workload Automation 5 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 Workload Automation 5 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`.

## Universal Command Server

### 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

You must have approximately 50 megabytes of disk space for the installation.

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:

**MAX-PROCESSES x AVERAGE-STDIO-SIZE x 2 = 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.

For more information on the Manager Fault Tolerant feature, spooling of redirected standard I/O files, and Universal Configuration Manager, see the [Indesca 5.1.0 User Guide](#) and [Infitran 5.1.0 User Guide](#).

## Security

Universal Broker and Universal Command Server require read/write access to the spool directory. No other Workload Automation 5 components



access the spool directory directly.

## UAG Cache

UAG cache is used by Universal Automation Center Agent (UAG) for storing standard I/O files.

### Space

Cache files are located, by default, in directory `.\Universal\UagSrv\cache`.

Cache files are created for each job that is run by an Agent. They remain in the cache until they are purged by an automated purge process scheduled nightly by the Opswise Automation Center Controller. You can configure the number of days that files remain in the cache using the user interface module Automation Center Administration->Properties->Agent Cache Retention Period In Days (for more information, see [Opswise Properties](#)).

The amount of disk space required for the cache directory depends on:

1. Number of jobs you estimate will run during the cache retention period you specified.
2. These files remain until they are purged by the automated cache purge process scheduled by the Opswise Automation Center Controller daily at midnight.
3. Average size of the user processes standard output and error files.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

**(RETENTION PERIOD x MAX-JOBS) x (AVERAGE-STDOUT-SIZE + AVERAGE-STDERR-SIZE) = required disk space.**

For example:

If the files are purged every 7 days, and you run 1200 jobs on that agent server daily, and the average size of the STDOUT + STDERR files is 3,000 bytes, the average amount of required disk space is 25MB (7 x 1,200 x 3000).

UAG automatically redirects the standard output and standard error files to the cache directory with no required user input.

### Security

Universal Automation Center Agent (UAG) requires read/write access to the UAG cache directory. No other Workload Automation 5 components access the cache directory. No general user access is required.

## UAG Crossgrade

The Universal Automation Center Agent (UAG) provided by the Workload Automation 5 for Windows installation replaces all previous versions (1.7 and earlier) of the Opswise Automation Center Agent.

If you are upgrading from OpsWise 1.7 and earlier to OpsWise 5.1, you must perform a UAG crossgrade (see [Migrating an Opswise Agent to UAG for Workload Automation 5 for Windows](#)).

## Workload Automation 5 for Windows - Installation Procedures

Error formatting macro: redirect: java.lang.NullPointerException

## **Workload Automation 5 for Windows - Installation Procedures Overview**

### **Workload Automation 5 for Windows - Installation Procedures**

The following procedures are provided for the installation and modification of Workload Automation 5 for Windows:


- [Installing Workload Automation 5 via the Graphical Interface](#)
- [Modifying a Workload Automation 5 Installation via the Graphical Interface](#)
- [Installing Workload Automation 5 via the Command Line Interface](#)
- [Modifying a Workload Automation 5 Installation via the Command Line Interface](#)
- [Migrating between 32- and 64-bit Workload Automation 5 for Windows Installs](#)
- [Migrating an Opswise Agent to UAG for Workload Automation 5 for Windows](#)

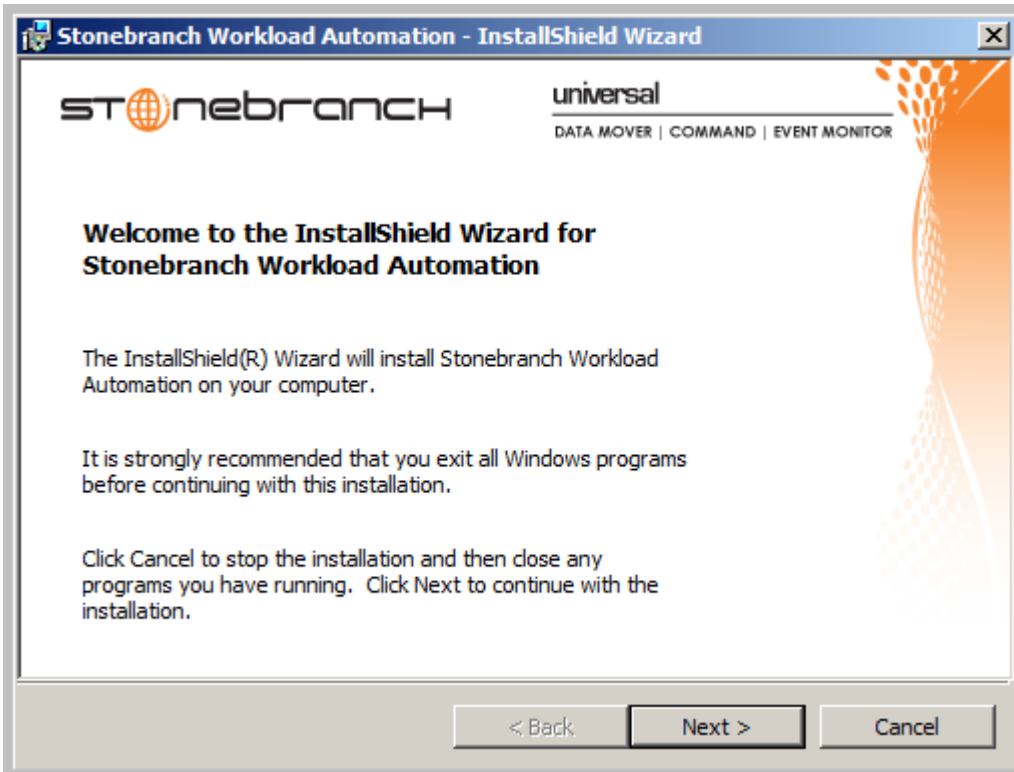
## Installing Workload Automation 5 via the Graphical Interface

- [Installing Workload Automation 5 for Windows via the Windows Installer Graphical Interface](#)
  - [Welcome Dialog \(32-bit Install\)](#)
  - [Welcome Dialog \(64-bit Install\)](#)
- [File Locations](#)
  - [32-bit Workload Automation 5 for Windows Package](#)
  - [64-bit Workload Automation 5 for Windows Package](#)

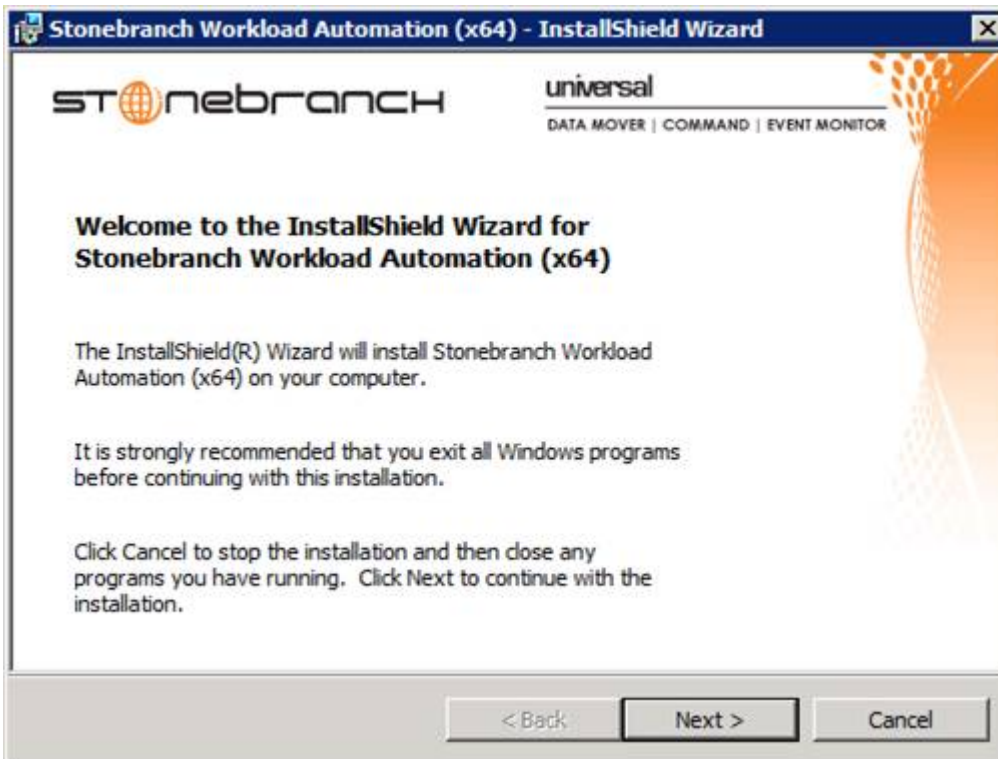
### Installing Workload Automation 5 for Windows via the Windows Installer Graphical Interface

To install Workload Automation 5 for Windows using the Windows Installer graphical interface, perform the following steps:

<b>Step 1</b>	<p>Download the desired Workload Automation 5 for Windows product distribution file to your work station:</p> <ul style="list-style-type: none"> <li>• <code>sb-5.1.0.&lt;level&gt;-windows-i386.exe</code>, the 32-bit Workload Automation 5 for Windows distribution file.</li> <li>• <code>sb-5.1.0.&lt;level&gt;-windows-x64.exe</code>, the 64-bit Workload Automation 5 for Windows distribution file (for supported 64-bit versions of Windows only).</li> </ul>
<b>Step 2</b>	<p>Execute the distribution file to begin the installation. When you execute the distribution file, the installation process determines whether a Windows Installer update is needed (see <a href="#">Windows Installer</a>).</p> <p>If no updates are necessary, the process extracts the Workload Automation Windows Installer Package (.msi) file to the locations listed below. The .msi file is all that's needed to install Workload Automation 5 for Windows, and its smaller size might make it a more convenient option for distributing and installing across several machines. The complete distribution file and the stand-alone .msi file accept the same command line properties, so either may be used to drive installs from the command line (see <a href="#">Installing Workload Automation 5 via the Command Line Interface</a>).</p> <p>The .msi file is extracted to the following locations:</p> <ul style="list-style-type: none"> <li>• 32-bit Workload Automation for Windows package: <code>&lt;LocalAppData&gt;\StonebranchWorkloadAutomation\&lt;packagecode&gt;\UCmd.msi</code></li> <li>• 64-bit Workload Automation for Windows package: <code>&lt;LocalAppData&gt;\StonebranchWorkloadAutomationx64\&lt;packagecode&gt;\UCmdx64.msi</code></li> </ul> <p>In the paths above, <code>&lt;LocalAppData&gt;</code> represents a particular user's local Application Data folder. For example, if the installation was performed by the built-in Administrator account, <code>&lt;LocalAppData&gt;</code> would expand by default to:</p> <ul style="list-style-type: none"> <li>• <code>C:\Documents and Settings\Administrator\Local Settings\Application Data</code> on Windows XP and Server 2003.</li> <li>• <code>C:\Users\Administrator\AppData\Local</code> on Windows Vista, 7, and Server 2008.</li> </ul> <p><code>&lt;packagecode&gt;</code> is a Universally Unique Identifier (UUID) in the format {XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}, where 'X' is a hexadecimal character in the range 0 (zero) – E.</p> <p>A sample UUID might look something like this: {3B10285A-9602-4DC9-B0A5-4D701BEB5225}.</p> <p>See <a href="#">File Locations</a>, below, for a summary of the different locations for the files involved in the Workload Automation for Windows installation.</p>
<b>Step 3</b>	<p>The installation starts after the files are extracted. It first will verify that your machine meets the minimum <a href="#">system requirements</a>. If the requirements are met, a Welcome dialog displays.</p> <div style="background-color: #ffffcc; padding: 10px; margin: 10px 0;"> <p> <b>Note</b> The same steps below apply regardless of whether you are performing a 32- or 64-bit Workload Automation for Windows installation. The only difference is the "(x64)" label shown for 64-bit installs. The Welcome dialog from each install is shown below, but for simplicity, the remaining steps show only the 32-bit install dialogs.</p> </div> <p><b>Welcome Dialog (32-bit Install)</b></p>

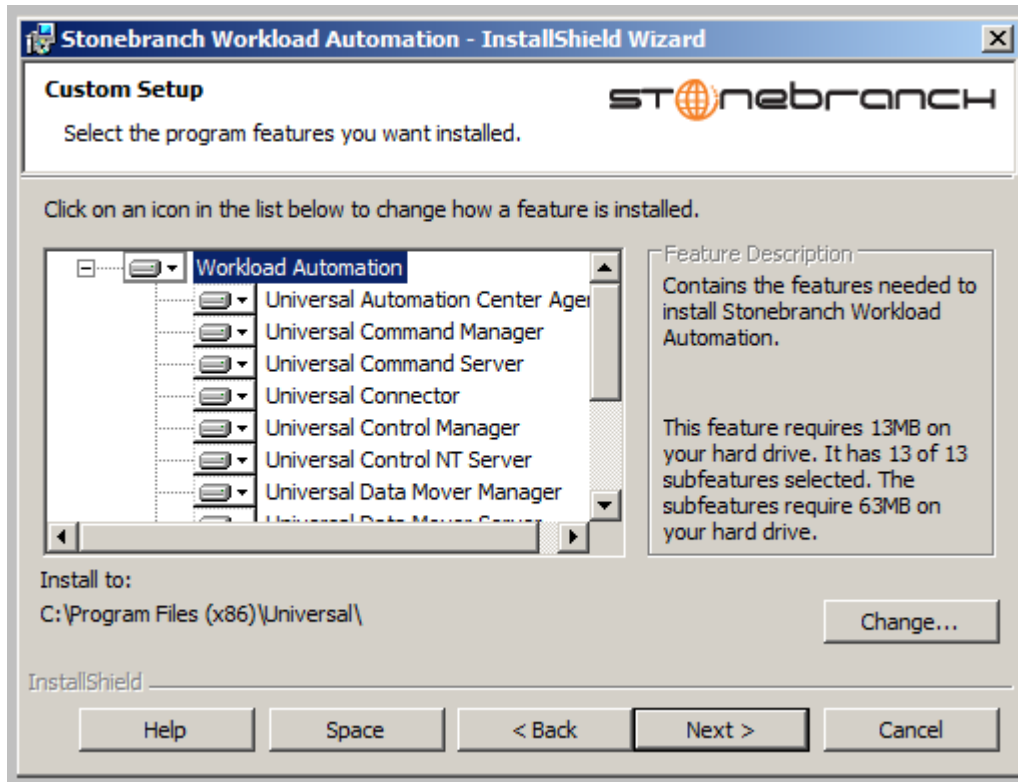


Welcome Dialog (64-bit Install)



**Step 4**

Click the **Next** button. A list of Workload Automation 5 components included in the installation package then displays. It is from this list that you can select which components to install.



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.

**A Stonebranch Tip**

If the installation detects an existing Workload Automation 5 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.

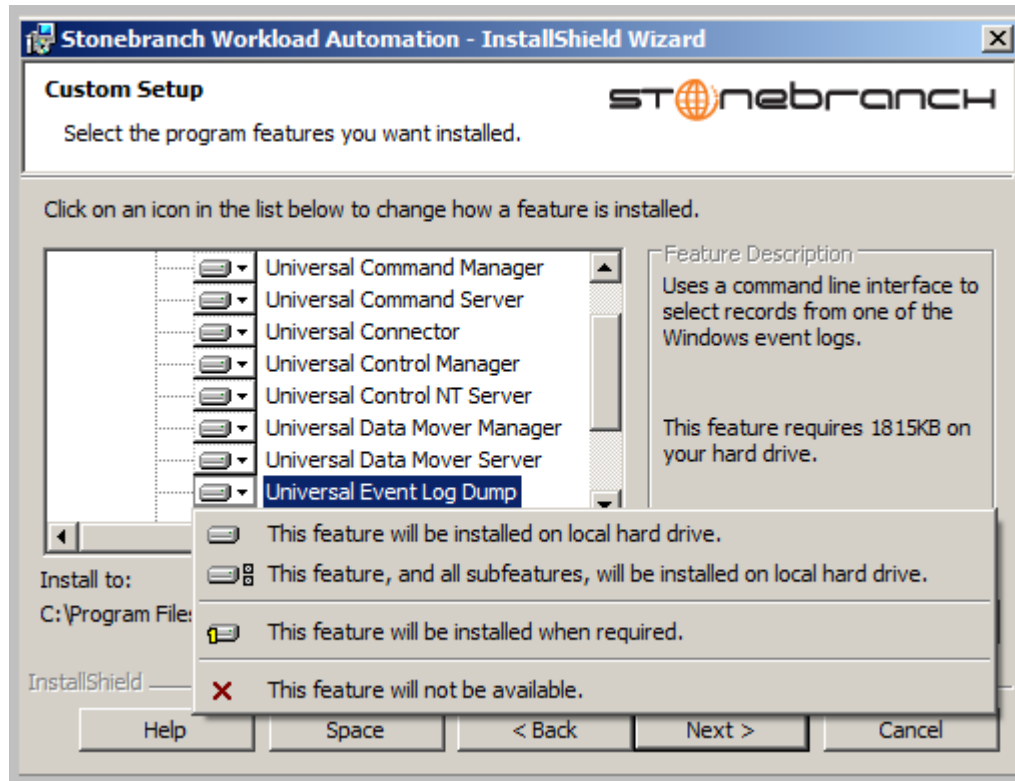
**Step 5**

The previous figure indicates that all Workload Automation 5 components will be installed in their respective directories under the **C:\Program Files\Universal** directory.

- If you want to select a different location, click the **Change...** button.
- 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.

**Step 6** If you do not want to install a component:

1. Click the drive icon next to that component name.
2. From the drop-down list that displays, select the X icon to mark the component as one not to be installed.  
For example, the following figure indicates that Universal Event Log Dump has been selected to not be installed.



**Step 7**

When you have selected the components (and their installation destinations) that you want to install, click the **Next>** button. The Universal Broker Service Account dialog then displays.

The screenshot shows a dialog box titled "Stonebranch Workload Automation - InstallShield Wizard". The main heading is "Universal Broker Service Account" with the Stonebranch logo to the right. Below the heading, it says "Select the account to use to execute the Universal Broker service." The main area contains instructions: "Choose one of the options below to specify the account that will be used to execute the Universal Broker service. An Administrative account will be created for a new user, or enter the ID and valid password of an existing Administrative account. The account currently used to execute the Broker is shown for upgrades." There are two radio button options: "Local System" (unselected) and "The user account with the ID and password specified below. (Recommended)" (selected). Below the second option are three text input fields: "User ID:" containing "UBrokerService", "Password:" containing "\*\*\*\*\*", and "Confirm Password:" containing "\*\*\*\*\*". At the bottom left is the "InstallShield" logo, and at the bottom right are three buttons: "< Back", "Next >", and "Cancel".

**Step 8**

Select an account to use to execute the Universal Broker service:

- **Local System**  
(This is the default for upgrades from releases prior to 4.3.0.1.)



- An Administrative account capable of executing the Universal Broker service. (This is the default for new installs or upgrades where the Universal Broker currently is executing with an account other than **Local System**.)

The user account ID defaults to **UBrokerService**, although any valid user name (up to 20 characters) can be specified. A domain name of up to 256 characters also can be entered for domain accounts, in the format domainname\userid.

If the account does not already exist, the install will create it with the privileges necessary to execute the Universal Broker service. See [Starting Universal Broker for Windows](#) for the privileges that an account needs to run the Universal Broker service.

A default password provided for the account can be obtained by contacting Stonebranch support, although any password can be specified.

If the user account does not exist, its password is set to the value entered here.

If the user account does exist, the installation process will validate the specified password. You must enter a valid password for the account to continue with the install.



#### Note

For upgrades, if the Universal Broker service is already configured to run with a user account, the install dialog shows only the account name and omits the password fields. The Broker service's configuration in the Windows Service Control Manager is retained across upgrades, making it unnecessary to re-enter account information during an upgrade.



The install provides the RUNBROKERASUSER, BROKERUID, and BROKERPWD command line options to override the default behavior described above.


- Specifying RUNBROKERASUSER=0 causes the install to select Local System, regardless of how the Broker is currently configured to execute.
- Specifying RUNBROKERASUSER=1 causes the install to select the user account option, even if the Broker is currently configured to run as Local System.

When RUNBROKERASUSER=1, the BROKERUID and BROKERPWD command line options are provided to override the default or currently configured user ID and password. The BROKERUID option limits user IDs to 20 characters; the BROKERPWD option for password has no character limit. (See [Windows Installer Command Line Parameters](#) for more information).

**Step 9**

When you have selected an account, click the **Next>** button. The Automation Center Connection Information dialog then displays.

**Stonebranch Workload Automation - InstallShield Wizard**

**Automation Center Connection Information** 

Enter Universal Automation Center agent connection information.

Enter a comma-separated list of one or more port and host addresses in the format port@host in the Automation Center Transports field.

The Automation Center Core specifies the queue name for the Automation Center message hub.

If no value is specified for the Automation Center Transport list, the default 4803@127.0.0.1 is used. If no value is specified for the Automation Center Core, a default of HUB01 is used.

Automation Center Transport(s):

Automation Center Core:

InstallShield

< Back    Next >    Cancel

**Note**

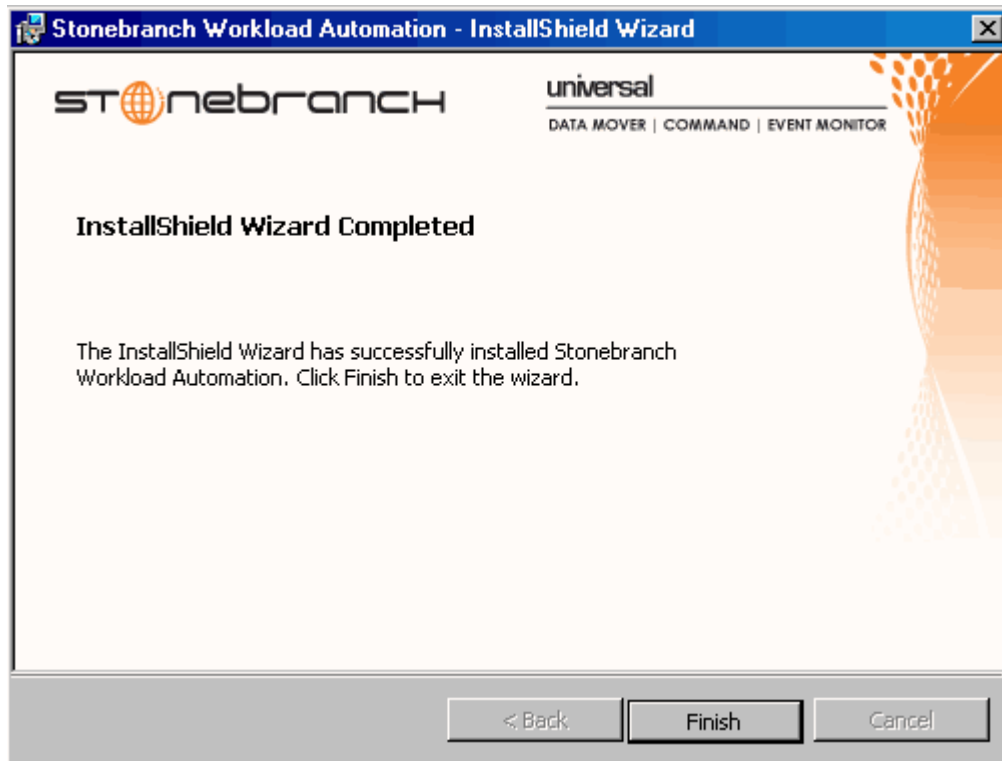
This dialog is only displayed for new Workload Automation 5 installs, upgrades to Workload Automation 5 from earlier releases, and 32-/64-bit Workload Automation crossgrades. It will not be displayed for upgrades from Workload Automation 5 to a later version.

**Step 10**

Enter values for the Automation Center Transport(s) and Automation Center Core. If you want to use the default values (as identified on this dialog), leave the fields empty.

**Step 11** 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.



**Step 12** Click the **Finish** button to exit Windows Installation.

## File Locations

This section summarizes the different locations for the files involved in the Workload Automation for Windows installation.

For purposes of this example:

- Package level is 0 (zero).
- Install is executed by the built-in Administrator user.
- Package code for the 32-bit Workload Automation for Windows package is {D16BD569-77D7-48D0-8EF0-3B0D143C44D8}.
- Package code for the 64-bit Workload Automation for Windows package is {3B10285A-9602-4DC9-B0A5-4D701BEB5225}.

### 32-bit Workload Automation 5 for Windows Package

<b>Distribution File</b>	sb-5.1.0.0-windows-i386.exe
<b>.msi File Name</b>	UCmd.msi
<b>Default .msi File Location: Windows XP, Server 2003</b>	C:\Documents and Settings\Administrator\Local Settings\Application Data\StonebranchWorkloadAutomation\{D16BD569-77D7-48D0-8EF0-3B0D143C44D8}
<b>Default .msi File Location: Windows Vista, 7, and Server 2008</b>	C:\Users\Administrator\AppData\Local\StonebranchWorkloadAutomation\{D16BD569-77D7-48D0-8EF0-3B0D143C44D8}

### 64-bit Workload Automation 5 for Windows Package

<b>Distribution File</b>	sb-5.1.0.0-windows-x64.exe
--------------------------	----------------------------

<b>.msi File Name</b>	UCmdx64.msi
<b>Default .msi File Location: Windows XP, Server 2003</b>	C:\Documents and Settings\Administrator\Local Settings\Application Data\StonebranchWorkloadAutomationx64\{3B10285A-9602-4DC9-B0A5-4D701BEB5225}
<b>Default .msi File Location: Windows Vista, 7, and Server 2008</b>	C:\Users\Administrator\AppData\Local\StonebranchWorkloadAutomationx64\{3B10285A-9602-4DC9-B0A5-4D701BEB5225}

## Modifying a Workload Automation 5 Installation via the Graphical Interface

- [Introduction](#)
- [Modifying a Workload Automation 5 Installation via the Windows Installer Graphical Interface](#)
- [Adding or Removing Workload Automation 5 Components](#)
- [Repairing a Corrupted Workload Automation 5 Installation](#)
- [Removing a Workload Automation 5 Installation](#)
  - [Un-Installed Files](#)

### Introduction

The information on this page applies to the 32- and 64-bit Workload Automation 5 for Windows installations.

The appearance of some of the screens shown below may differ between the two packages. The product name displayed for a 32-bit Workload Automation 5 installation is simply "Stonebranch Workload Automation." The "(x64)" label is displayed for a 64-bit Workload Automation 5 installation, making the product name "Stonebranch Workload Automation (x64)."

The sequence of steps to modify, repair, or remove a Workload Automation 5 install is the same for 32- and 64-bit packages.

### Modifying a Workload Automation 5 Installation via the Windows Installer Graphical Interface

This page describes how to modify a Workload Automation 5 installation via the Windows Installer graphical interface.

After Workload Automation 5 is installed, the Windows Installer can be run as many times as needed to modify the installation by:

- [Adding or Removing Workload Automation 5 Components](#)
- [Repairing a Corrupted Workload Automation 5 Installation](#)
- [Removing a Workload Automation 5 Installation](#)

### Adding or Removing Workload Automation 5 Components

To add or remove components from a Workload Automation 5 installation, perform the following steps:

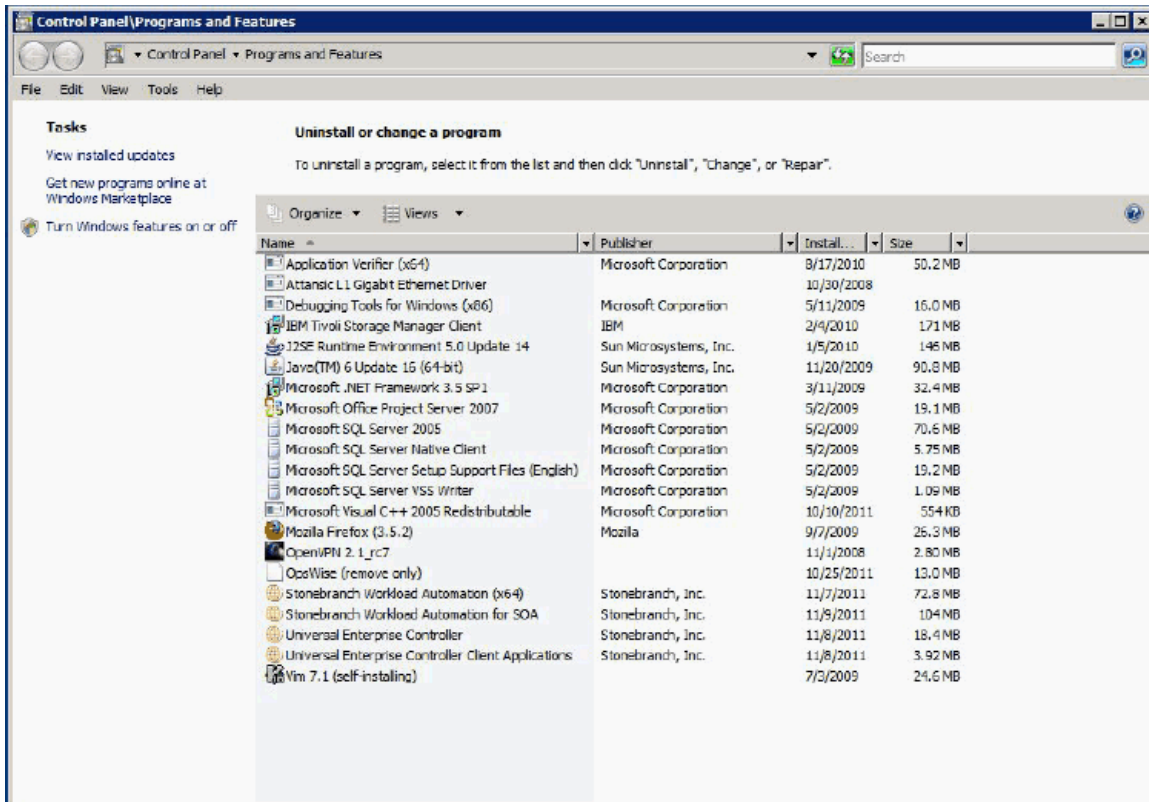
**Step 1** On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays.

**Step 2** From the list of installed programs, select **Workload Automation 5**.

Program Name	Size	Used
SoundMAX	2.01MB	
Steelray Project Viewer 3.1.3.0	79.01MB	
<b>Stonebranch Workload Automation</b>	65.00MB	rarely
Stonebranch Workload Automation for SOA	105.00MB	
Sun ODF Plugin for Microsoft Office 1.2	53.91MB	
TN3270 Plus	2.22MB	
TortoiseCVS 1.8.18	9.44MB	
Universal Enterprise Controller	22.94MB	
Universal Enterprise Controller Client Applications	3.92MB	

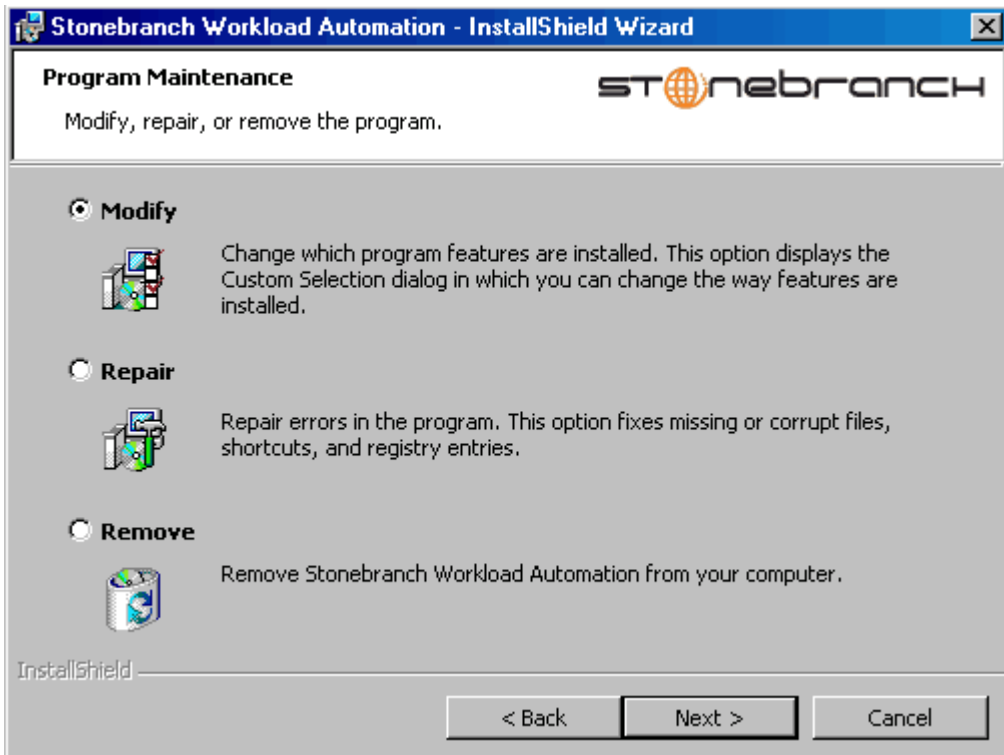
**Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**

The Programs and Features dialog, below, replaces the Add or Remove Programs dialog.

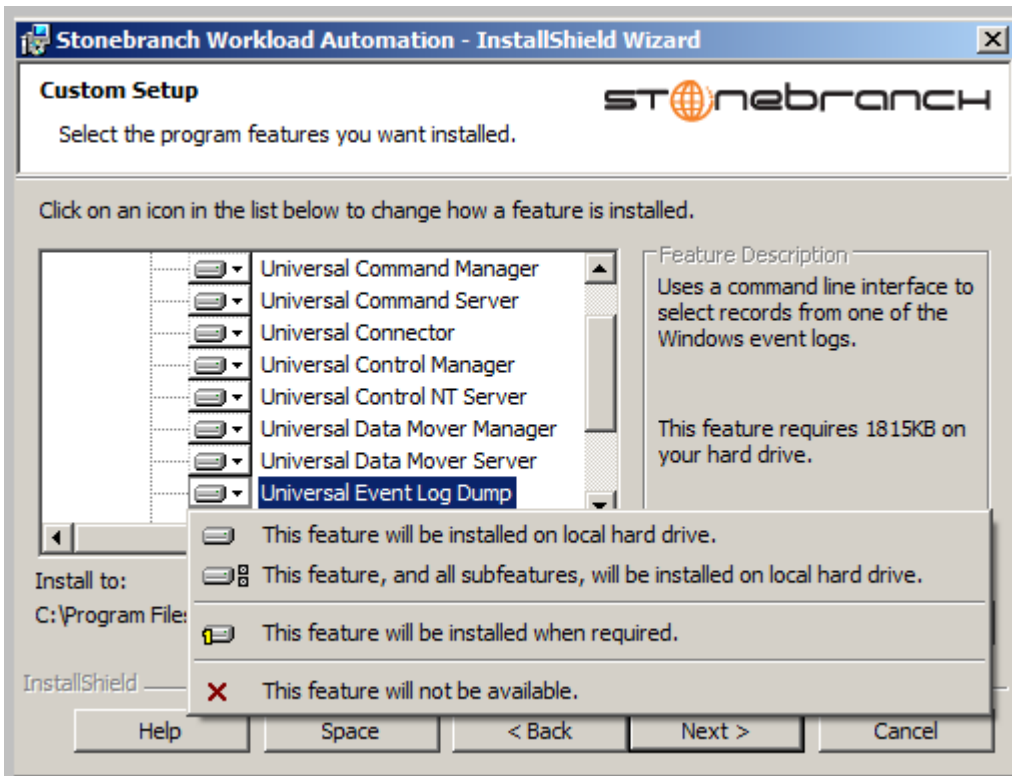


**Step 3** Click the **Change** button to start Windows Installer.

**Step 4** On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays.



**Step 5** Click the **Modify** radio button, and then the **Next>** button, to display the Custom Setup dialog.



Currently installed components are identified by a drive icon.

Uninstalled components are identified by an **X** icon.

<b>Step 6</b>	<p>To remove a currently installed component:</p> <ol style="list-style-type: none"> <li>1. Click the drive icon next to that component.</li> <li>2. Select the X icon from the drop-down list to mark the component for removal.</li> </ol>
<b>Step 7</b>	<p>To add an uninstalled component:</p> <ol style="list-style-type: none"> <li>1. Click the X icon next to that component.</li> <li>2. Select the drive icon from the drop-down list to mark the component for installation.</li> </ol>
<b>Step 8</b>	<p>Click the <b>Next&gt;</b> button to continue with the modification. When the modifications are complete, the following actions will be taken:</p> <ul style="list-style-type: none"> <li>• Components marked with a drive icon will: <ul style="list-style-type: none"> <li>• Remain installed if they already are installed.</li> <li>• Be installed if they are not already installed.</li> </ul> </li> <li>• Components marked with an <b>X</b> will: <ul style="list-style-type: none"> <li>• Remain uninstalled if they are not currently installed</li> <li>• Be removed if they currently are installed.</li> </ul> </li> </ul>

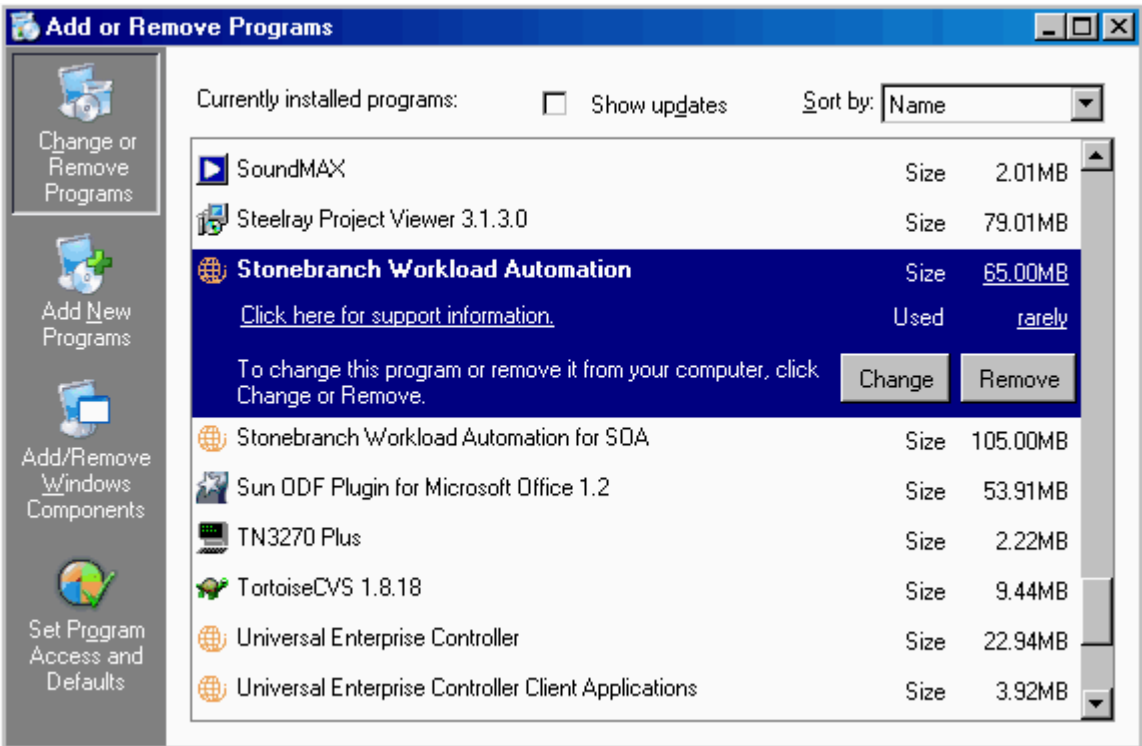
### Repairing a Corrupted Workload Automation 5 Installation

Windows Installer has the ability to recover accidentally deleted application files, configuration and component definition files, and registry entries required by Workload Automation 5. 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.

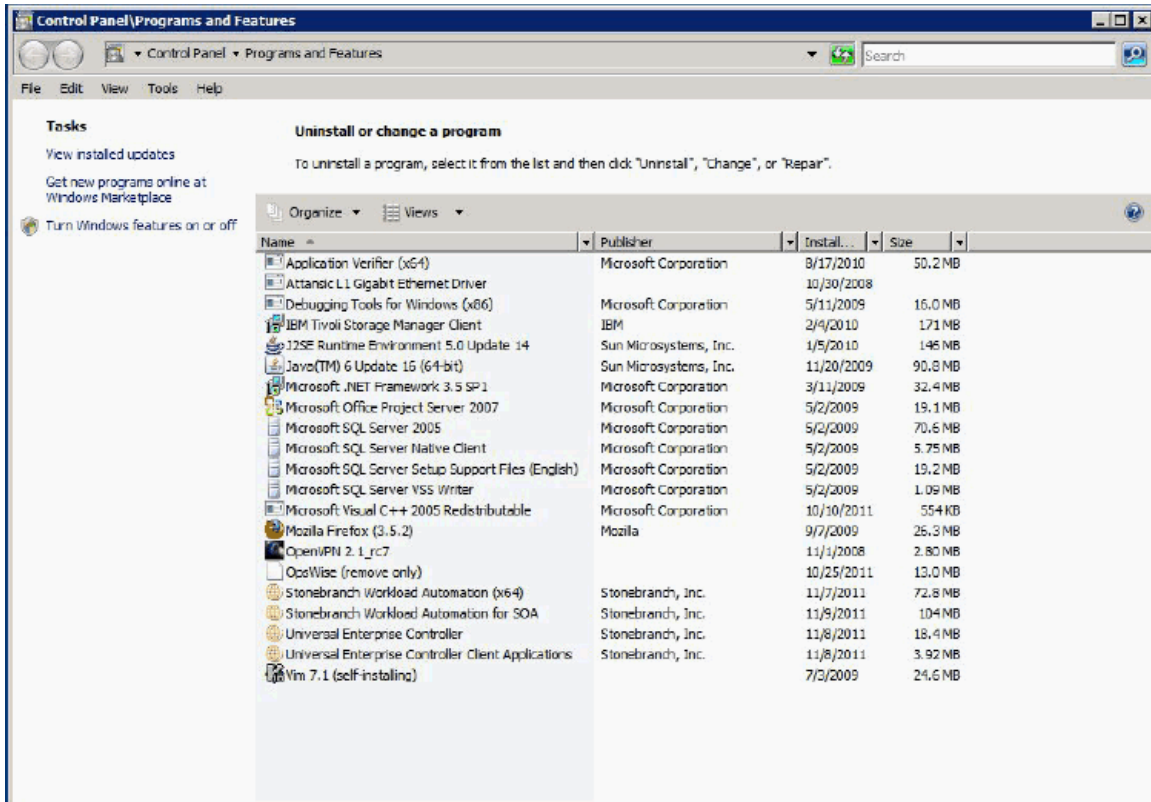
To repair an installation, perform the following steps:

**Step 1** On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays.




**Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**  
The Programs and Features dialog replaces the Add or Remove Programs dialog.



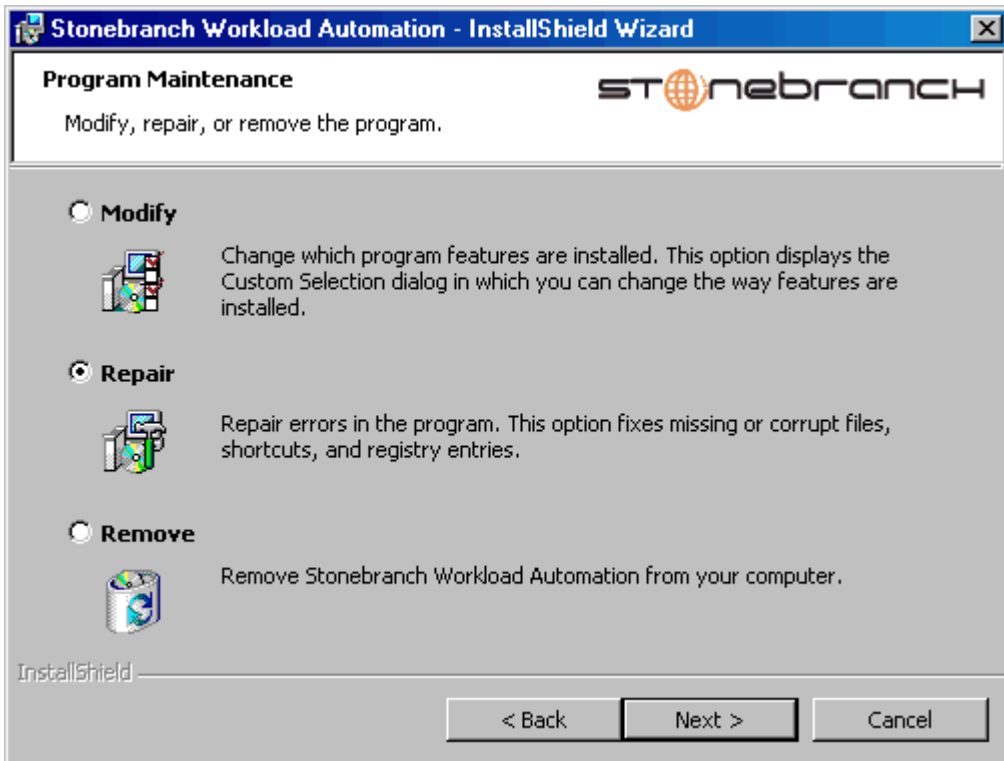


**Step 2** From the list of installed programs, select **Workload Automation 5**.

 **Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**  
To skip the remaining steps, click "Repair".

**Step 3** Click the **Change** button to start Windows Installer.

**Step 4** On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays.



**Step 5** Click the **Repair** radio button, and then the **Next>** button, to display the Ready to Repair dialog.

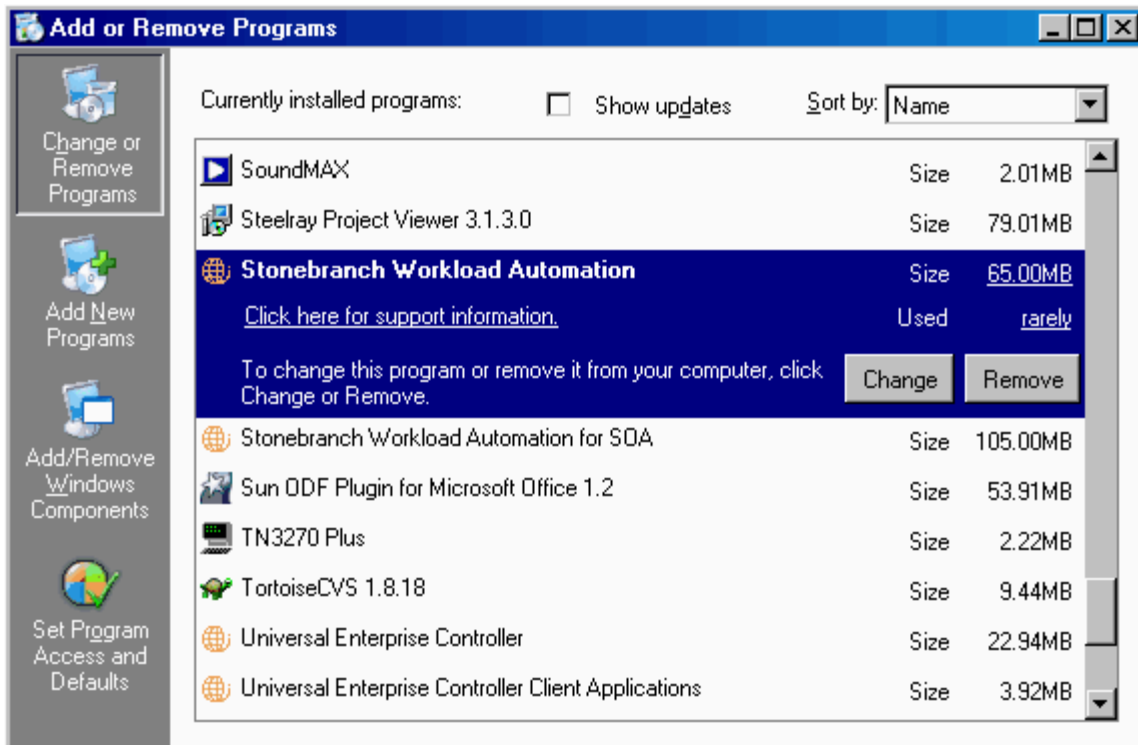
**Step 6** Follow the instructions displayed in successive dialogs to complete the repair.

## Removing a Workload Automation 5 Installation

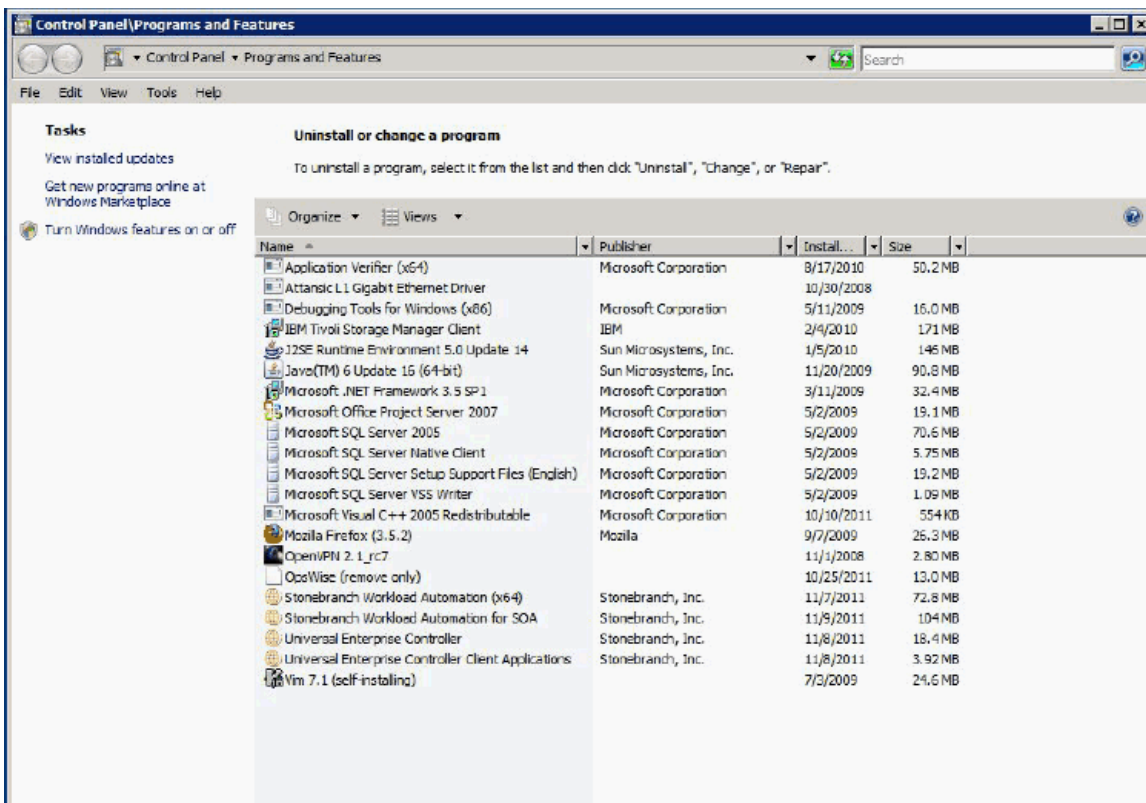
To uninstall a Workload Automation 5 installation, perform the following steps:

**Step 1**

On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays.



**Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**  
 The Programs and Features dialog replaces the Add or Remove Programs dialog.

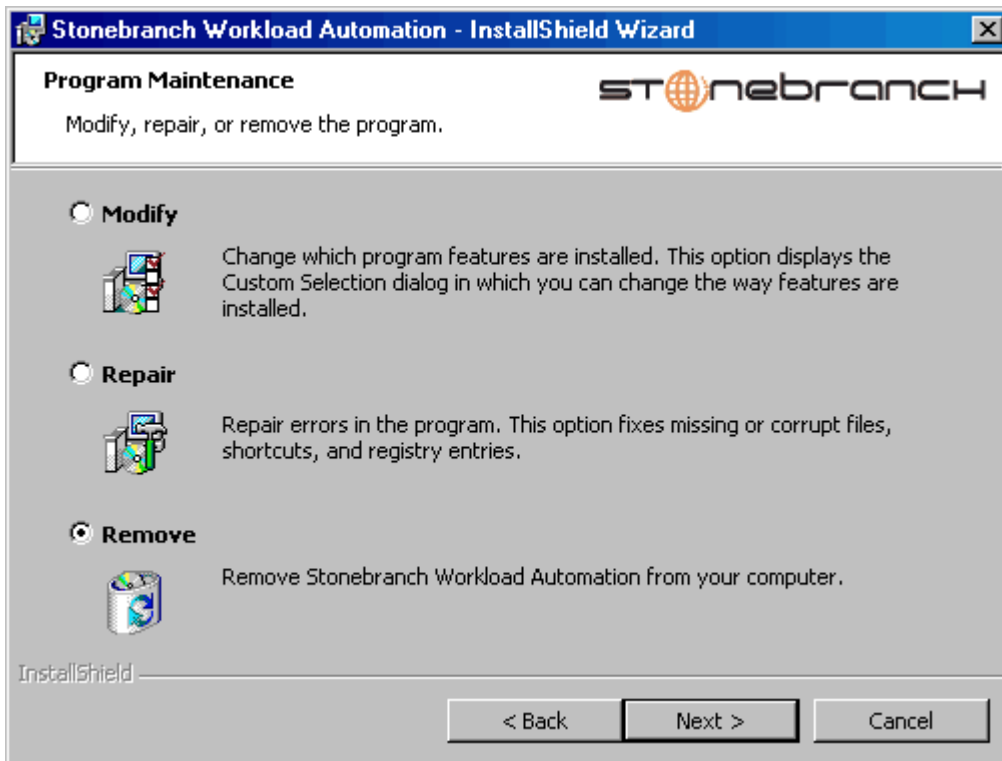


**Step 2** From the list of installed programs, select **Workload Automation 5**.

**Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**  
To skip the remaining steps, click "Uninstall".

**Step 3** Click the **Change** button to start Windows Installer.

**Step 4** On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays.



**Step 5** Select **Remove** and click the **Next>** button.

**Step 6** On the subsequent dialog, click the **Remove** button to remove the Workload Automation 5 installation.

### Un-Installed Files

The un-install process will remove only those files created during the installation. Some files stored under the **.Universal** install directory by Workload Automation 5, 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 **.Universal** directory, make sure that no other Stonebranch, Inc. products are installed there. (See [Workload Automation 5 for Windows - File Inventory Lists](#) for a list of files and directories installed with Workload Automation 5.)

In addition to those files and directories created by the Workload Automation 5 installation, there may be some shared files (for example, codepage files) and Workload Automation 5 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.

## Installing Workload Automation 5 via the Command Line Interface

- [Installing Workload Automation 5 via the Command Line Interface](#)
- [File Locations](#)
  - [32-bit Workload Automation 5 for Windows Package](#)
  - [64-bit Workload Automation 5 for Windows Package](#)
- [Workload Automation 5 Installation Command Line Parameters](#)
  - [Distribution File Command Line Syntax](#)
  - [Distribution File Command Line Switches](#)
  - [Windows Installer Command Line Syntax](#)
  - [Windows Installer Command Line Parameters](#)
- [Workload Automation 5 Command Line Installation Examples](#)
- [Detecting the Completion of a Silent Install](#)

### Installing Workload Automation 5 via the Command Line Interface

This page describes how to install Workload Automation 5 using the command line interface. Except where noted, the instructions are the same for the 32- and 64-bit Workload Automation packages.

The command line interface is useful in situations where:

- Several Workload Automation 5 installations must be deployed across many different systems.
- It is not practical or convenient to perform the graphical interface installation.
- It is necessary to generate an installation log file.

To install Workload Automation 5 for Windows using the command line interface, perform the following steps:

<b>Step 1</b>	<p>Download the desired Workload Automation 5 for Windows product distribution file to your work station:</p> <ul style="list-style-type: none"> <li>• <code>sb-5.1.0.&lt;level&gt;-windows-i386.exe</code>, the 32-bit Workload Automation 5 for Windows distribution file.</li> <li>• <code>sb-5.1.0.&lt;level&gt;-windows-x64.exe</code>, the 64-bit Workload Automation 5 for Windows distribution file (for supported 64-bit versions of Windows only).</li> </ul>
<b>Step 2</b>	<p>Execute the distribution file to begin the installation.</p> <p>When you execute the distribution file, the installation process determines whether a Windows Installer update is needed (see <a href="#">Windows Installer</a>).</p> <p>If no updates are necessary, the process extracts the Workload Automation Windows Installer Package (.msi) file to the locations listed below. The .msi file is all that is needed to install Workload Automation 5 for Windows, and its smaller size might make it a more convenient option for distributing and installing across several machines. The complete distribution file and the stand-alone .msi file accept the same command line properties, so either may be used to drive installs from the command line (see <a href="#">Windows Installer Command Line Syntax</a>).</p> <p>The .msi file is extracted to the following locations:</p> <ul style="list-style-type: none"> <li>• 32-bit Workload Automation for Windows package: <code>&lt;LocalAppData&gt;\StonebranchWorkloadAutomation\&lt;packagecode&gt;\UCmd.msi</code></li> <li>• 64-bit Workload Automation for Windows package: <code>&lt;LocalAppData&gt;\StonebranchWorkloadAutomationx64\&lt;packagecode&gt;\UCmdx64.msi</code></li> </ul> <p>In the paths above, <code>&lt;LocalAppData&gt;</code> represents a particular user's local Application Data folder. For example, if the installation was performed by the built-in Administrator account, <code>&lt;LocalAppData&gt;</code> would expand by default to:</p> <ul style="list-style-type: none"> <li>• <code>C:\Documents and Settings\Administrator\Local Settings\Application Data</code> on Windows XP and Server 2003.</li> <li>• <code>C:\Users\Administrator\AppData\Local</code> on Windows Vista, 7, and Server 2008.</li> </ul> <p><code>&lt;packagecode&gt;</code> is a Universally Unique Identifier (UUID) in the format <code>{xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}</code>, where <code>x</code> is a hexadecimal character in the range 0 (zero) – E.</p> <p>A sample UUID might look something like this: <code>{3B10285A-9602-4DC9-B0A5-4D701BEB5225}</code>.</p> <p>See <a href="#">File Locations</a>, below, for a summary of summarizes the different locations for the files involved in the Workload Automation for Windows installation.</p>
<b>Step 3</b>	<p>The installation starts after the files are extracted.</p> <p>It first will verify that your machine meets the minimum <a href="#">system requirements</a>. If the requirements are met, a Welcome dialog displays.</p>

## File Locations

This section summarizes the different locations for the files involved in the Workload Automation for Windows installation.

For purposes of this example:

- Package level is 0 (zero).
- Install is executed by the built-in Administrator user.
- Package code for the 32-bit Workload Automation for Windows package is {D16BD569-77D7-48D0-8EF0-3B0D143C44D8}.
- Package code for the 64-bit Workload Automation for Windows package is {3B10285A-9602-4DC9-B0A5-4D701BEB5225}.

### 32-bit Workload Automation 5 for Windows Package

<b>Distribution File</b>	sb-5.1.0.0-windows-i386.exe
<b>.msi File Name</b>	UCmd.msi
<b>Default .msi File Location: Windows XP, Server 2003</b>	C:\Documents and Settings\Administrator\Local Settings\Application Data\StonebranchWorkloadAutomation\{{{D16BD569-77D7-48D0-8EF0-3B0D143C44D8}}*
<b>Default .msi File Location: Windows Vista, 7, and Server 2008</b>	C:\Users\Administrator\AppData\Local\StonebranchWorkloadAutomation\{D16BD569-77D7-48D0-8EF0-3B0D143C44D8}

### 64-bit Workload Automation 5 for Windows Package

<b>Distribution File</b>	sb-5.1.0.0-windows-x64.exe
<b>.msi File Name</b>	UCmdx64.msi
<b>Default .msi File Location: Windows XP, Server 2003</b>	C:\Documents and Settings\Administrator\Local Settings\Application Data\StonebranchWorkloadAutomationx64\{3B10285A-9602-4DC9-B0A5-4D701BEB5225}
<b>Default .msi File Location: Windows Vista, 7, and Server 2008</b>	C:\Users\Administrator\AppData\Local\StonebranchWorkloadAutomationx64\{3B10285A-9602-4DC9-B0A5-4D701BEB5225}

## Workload Automation 5 Installation Command Line Parameters

There are two sets of command line parameters available when executing the distribution file.

1. The more basic of the two sets is processed directly by the distribution file to control behavior of the set-up application (see [Distribution File Command Line Syntax](#) and [Distribution File Command Line Switches](#), below).
2. The set-up application also accepts a more advanced set of parameters that it passes to the Windows Installer (**msiexec**). These parameters control the extracted .msi file's behavior during the install process (see [Windows Installer Command Line Syntax](#) and [Windows Installer Command Line Parameters](#), below).

### Distribution File Command Line Syntax

The following illustrates the syntax used to install Workload Automation 5 from the command line:

```
sb-5.1.0.<level>-windows-<platform>.exe [/v"Windows Installer parameters" [/s] [/w] [/x]
```

In this syntax:

- **<level>** is the numeric package level.
- **<platform>** is:
  - **i386** for the 32-bit Workload Automation distribution file.
  - **x64** for the 64-bit Workload Automation distribution file.

## Distribution File Command Line Switches

The following table describes the command line switches available for installing Workload Automation 5 for Windows from the command line.

Command Line Switch	Description
<b>/v</b>	Passes parameters to the Windows Installer ( <b>msiexec</b> ).  The list of parameters must be enclosed in double (") quotation marks. See <a href="#">Windows Installer Command Line Syntax</a> for available parameters.
<b>/s</b>	Instructs the set-up application to run without a graphical interface (that is, a silent install).  Use this switch along with the <b>/q</b> Windows Installer command line switch for a completely silent install.
<b>/w</b>	Instructs the set-up application to wait until the installation completes.  Use this switch when launching the installation from a script file. Without it, the set-up application may return immediately after launching Windows Installer.
<b>/x</b>	Uninstalls Workload Automation 5 for Windows.

## Windows Installer Command Line Syntax

The following figure illustrates the command line syntax when using Windows Installer to install Workload Automation 5 from the command line.

```
msiexec.exe [/i|/fom|/x] SetupPath\Ucmd[x64].msi
  [INSTALLDIR=installdir]
  [RUNBROKERASUSER={0|1}]
  [BROKERUID=BrokerAccountId]
  [BROKERPWD=BrokerAccountPassword]
  [CONVERT_OPSAGENT={yes|no}]
  [AC_TRANSPORTS=port@ipaddr[,port2@ipaddr2,...,portn@ipaddrn]]
  [AC_CORE=value]
  [UCMDSRVWORKFOLDER=WorkFolderPath]
  [UCTLSRVWORKFOLDER=WorkFolderPath]
  [UDMSRVWORKFOLDER=WorkFolderPath]
  [UEMSRVWORKFOLDER=WorkFolderPath]
  [UAGSRV={yes|no}]
  [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}]
  [USAP={yes|no}]
  [USPOOL={yes|no}]
  [/q[n|b|r]] [/L*v logfilepath]
```

The order in which the parameters are specified is significant.

The **/i**, **/fom**, or **/x** switches 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 Workload Automation 5 for Windows 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** switches.

The **/q** and **/L** switches can be specified in any order, but, if used, must come after all other parameters.

## Windows Installer Command Line Parameters

The following table describes the optional parameters that are available when using Windows Installer to install Workload Automation 5 from the command line.

Parameter	Description	Default
-----------	-------------	---------

<div style="border: 1px solid black; padding: 2px; width: fit-content;">/i</div>	<p>Installs Workload Automation 5 for Windows.</p> <p><b>/i</b> cannot be used with either the <b>/fom</b> (repair) or <b>/x</b> (remove) parameter.</p>	<p>n/a</p>
<div style="border: 1px solid black; padding: 2px; width: fit-content;">/fom</div>	<p>Repairs a Workload Automation 5 for Windows installation.</p> <p><b>om</b> (after the <b>/f</b>) are options used by the repair. There are other options available, but for behavior that matches the repair done from the graphical install, the <b>om</b> options must be used.</p> <p><b>/fom</b> cannot be used with either the <b>/i</b> (install) or <b>/x</b> (remove) parameter.</p>	<p>n/a</p>
<div style="border: 1px solid black; padding: 2px; width: fit-content;">/x</div>	<p>Removes a Workload Automation 5 for Windows installation.</p> <p><b>/x</b> cannot be used with either the <b>/i</b> (install) or <b>/fom</b> (repair) parameter.</p>	<p>n/a</p>
<div style="border: 1px solid black; padding: 2px; width: fit-content;">SetupPath \\Ucmd[x64].msi</div>	<p>Specifies the path to the <b>Ucmd.msi</b> or <b>Ucmdx64.msi</b> file. <b>SetupPath</b> is chosen automatically by the distribution file and depends on the Windows version as described in <a href="#">Step 2</a>, above.</p>	<p>(none)</p>
<div style="border: 1px solid black; padding: 2px; width: fit-content;">INSTALLDIR</div>	<p>Sets the root installation directory to <b>&lt;installdir&gt;</b>. Each component will be installed under this directory.</p> <p><b>INSTALLDIR</b> is required only if you want to install Workload Automation 5 for Windows under a directory different from the one specified by the <b>PROGRAMFILES</b> environment variable (typically <b>C:\Program Files\Universal</b>).</p> <p><b>INSTALLDIR</b> is valid only when the <b>/i</b> switch is used. Otherwise, it is ignored. If the directory contains spaces, you must use double ( " ) quotation marks around the path name.</p>	<p>(none)</p>
<div style="border: 1px solid black; padding: 2px; width: fit-content;">RUNBROKERASUSER</div>	<p>Controls whether the Universal Broker service executes as an Administrative user account or as <b>Local System</b>.</p> <p>Valid values are <b>0</b> and <b>1</b>.</p> <ul style="list-style-type: none"> <li>• If <b>RUNBROKERASUSER=1</b>, the install will perform the following steps:             <ol style="list-style-type: none"> <li>1. If the Universal Broker service is not installed or currently is configured to run as <b>Local System</b>, the install will set the service's start-up account to <b>UBrokerService</b> or the user ID specified by the <b>BROKERUID</b> command line option.</li> <li>2. If the Universal Broker service is installed and already is configured to run as a user account, the Broker's start-up account will not be changed (allows customized user accounts to be preserved during upgrades).</li> </ol> </li> <li>• If <b>RUNBROKERASUSER=0</b>, the install will perform the following steps:             <ol style="list-style-type: none"> <li>1. If the Universal Broker service is not installed or is currently configured to run as an account other than <b>Local System</b>, the install will set the service's start-up account to <b>Local System</b>.</li> </ol> </li> </ul> <div style="background-color: #ffffcc; padding: 10px; margin: 10px 0;"> <p> <b>Note</b> The Universal Broker service's properties will be set to allow Universal Broker to interact with the desktop. This is supported only when Universal Broker runs as Local System.)</p> </div> <ol style="list-style-type: none"> <li>2. If the Universal Broker service is installed and is currently configured to run as <b>Local System</b>, the Universal Broker service's properties will not be changed.</li> </ol>	<p><b>1</b>: new installs <b>0</b>: upgrades earlier than 4.3.0</p>



BROKERUID	<p>Used by the install when RUNBROKERASUSER=1 to override the default or currently configured user ID associated with the account used to execute the Universal Broker service. The install also uses <b>BROKERUID</b> to initialize the User ID shown in the Universal Broker Service Account dialog.</p> <p>Valid values are any characters. The maximum supported length of a User ID is 20 characters. A domain name of up to 256 characters also can be specified for domain accounts, in the format domainname\userid.</p> <p><b>BROKERUID</b> is ignored if RUNBROKERASUSER=0.</p>	(none)
BROKERPWD	<p>Used by the install when RUNBROKERASUSER=1 to specify the password for the account used to execute the Universal Broker service. The install also uses <b>BROKERPWD</b> to initialize the Password shown in the Universal Broker Service Account dialog.</p> <p>Valid values are any characters.</p> <p><b>BROKERPWD</b> is ignored if RUNBROKERASUSER=0.</p>	(none)
CONVERT_OPSAGENT	<p>Specification (yes or no) that causes Windows Installer to:</p> <ol style="list-style-type: none"> <li>1. Search for an existing Opwise Agent install (1.5 or later) and convert configuration options stored in the <b>agent.props</b> file to corresponding options in <b>uags.conf</b>.</li> <li>2. Stops the Opwise Agent service and change its start-up type from "Automatic" to "Manual."</li> <li>3. Assigns the ID used by the Opwise agent to UAG by moving the <b>qname</b> file from the Opwise install directory to the <b>.\UagSrv\var</b> directory.</li> </ol>	no
AC_TRANSPORTS	<p>Specifies a value for the port and network address of the Opwise Automation Center Transporter(s) used for network communication. The install uses this value to set the UAG <b>AUTOMATION_CENTER_TRANSPORTS</b> configuration option. The value specified in <b>AC_TRANSPORTS</b> will override any currently configured options, including those imported from <b>agent.props</b> if <b>CONVERT_OPSAGENT</b> is set to <b>yes</b>.</p> <p>If <b>AC_TRANSPORTS</b> is omitted from the command line, and no other value specifies the port and network address of the Transporter(s), the default is used.</p>	4803@127.0.0.1
AC_CORE	<p>Specifies a value for the queue name of the Opwise Automation Center Message Hub. The install uses this value to set the UAG <b>AUTOMATION_CENTER_CORE</b> configuration option. The value specified in <b>AC_CORE</b> overrides any currently configured options, including those imported from <b>agent.props</b> when <b>CONVERT_OPSAGENT</b> is set to <b>yes</b>.</p> <p>If <b>AC_CORE</b> is omitted from the command line, and no other value specifies the queue name of the Message Hub, the default is used.</p>	HUB01
UCMDSRVWORKFOLDER	<p>Sets the Universal Command Server working folder. This value is used only if the Universal Command Server is being installed.</p> <p>It is valid only when the <b>/i</b> switch is used. Otherwise, it is ignored. If the folder contains spaces, you must use double ( " ) quotation marks around the path name.</p>	INSTALLDIR\UcmdHome
UCTLSRVWORKFOLDER	<p>Sets the Universal Control Server working folder. This value is used only if the Universal Control Server is being installed.</p> <p>It is valid only when the <b>/i</b> switch is used. Otherwise, it is ignored. If the folder contains spaces, you must use double ( " ) quotation marks around the path name.</p>	INSTALLDIR\UcmdHome
UDMSRVWORKFOLDER	<p>Sets the Universal Data Mover Server working folder. This value is used only if the Universal Data Mover Server is being installed.</p> <p>It is valid only when the <b>/i</b> switch is used. Otherwise, it is ignored. If the folder contains spaces, you must use double ( " ) quotation marks around the path name.</p>	INSTALLDIR\UdmHome

UEMSRVWORKFOLDER	Sets the Universal Event Monitor Server working folder. This value is used only if the Universal Event Monitor Server is being installed.  It is valid only when the <i>/i</i> switch is used. Otherwise, it is ignored. If the folder contains spaces, you must use double ( " ) quotation marks around the path name.	<b>INSTALLDIR\UemHome</b>
UAGSRV	Prevents Universal Automation Center Agent (UAG) from being installed (if value is <b>no</b> ). This has the same effect as choosing not to install Universal Automation Center Agent from the Custom Setup dialog.  It is valid only when the <i>/i</i> switch is used. Otherwise, it is ignored.	<b>yes</b>
UCMDMGR	Prevents Universal Command Manager from being installed (if value is <b>no</b> ). This has the same effect as choosing not to install Universal Command Manager from the Custom Setup dialog.  It is valid only when the <i>/i</i> switch is used. Otherwise, it is ignored.	<b>yes</b>
UCMSRV	Prevents Universal Command Server from being installed (if value is <b>no</b> ). This has the same effect as choosing not to install Universal Command Server from the Custom Setup dialog.  It is valid only when the <i>/i</i> switch is used. Otherwise, it is ignored.	<b>yes</b>
UCTLMGR	Prevents Universal Control Manager from being installed (if value is <b>no</b> ). This has the same effect as choosing not to install Universal Control Manager from the Custom Setup dialog.  It is valid only when the <i>/i</i> switch is used. Otherwise, it is ignored.	<b>yes</b>
UCTLSRV	Prevents Universal Control Server from being installed (if value is <b>no</b> ). This has the same effect as choosing not to install Universal Control Server from the Custom Setup dialog.  It is valid only when the <i>/i</i> switch is used. Otherwise, it is ignored.	<b>yes</b>
UDMMGR	Prevents Universal Data Mover Manager from being installed (if value is <b>no</b> ). This has the same effect as choosing not to install Universal Data Mover Manager from the Custom Setup dialog.  It is valid only when the <i>/i</i> switch is used. Otherwise, it is ignored.	<b>yes</b>
UDMSRV	Prevents Universal Data Mover Server from being installed (if value is <b>no</b> ). This has the same effect as choosing not to install Universal Data Mover Server from the Custom Setup dialog.  It is valid only when the <i>/i</i> switch is used. Otherwise, it is ignored.	<b>yes</b>
UELD	Prevents Universal Event Log Dump from being installed (if value is <b>no</b> ). This has the same effect as choosing not to install Universal Event Log Dump from the Custom Setup dialog.  It is valid only when the <i>/i</i> switch is used. Otherwise, it is ignored.	<b>yes</b>
UEMMGR	Prevents Universal Event Monitor Manager from being installed (if value is <b>no</b> ). This has the same effect as choosing not to install Universal Event Monitor Manager from the Custom Setup dialog.  It is valid only when the <i>/i</i> switch is used. Otherwise, it is ignored.	<b>yes</b>
UEMSRV	Prevents Universal Event Monitor Server from being installed (if value is <b>no</b> ). This has the same effect as choosing not to install Universal Event Monitor Server from the Custom Setup dialog.  It is valid only when the <i>/i</i> switch is used. Otherwise, it is ignored.	<b>yes</b>
UQUERY	Prevents Universal Query from being installed (if value is <b>no</b> ). This has the same effect as choosing not to install Universal Query from the Custom Setup dialog.  It is valid only when the <i>/i</i> switch is used. Otherwise, it is ignored.	<b>yes</b>

<div style="border: 1px solid black; padding: 5px; width: fit-content;">USAP</div>	<p>Prevents Universal Connector from being installed (if value is <b>no</b>). This has the same effect as choosing not to install Universal Connector from the Custom Setup dialog.</p> <p>It is valid only when the <code>/i</code> switch is used. Otherwise, it is ignored.</p>	<p><b>yes</b></p>
<div style="border: 1px solid black; padding: 5px; width: fit-content;">USPOOL</div>	<p>Prevents Universal Spool utilities from being installed (if value is <b>no</b>). This has the same effect as choosing not to install Universal Spool utilities from the Custom Setup dialog.</p> <p>It is valid only when the <code>/i</code> switch is used. Otherwise, it is ignored.</p>	<p><b>yes</b></p>
<div style="border: 1px solid black; padding: 5px; width: fit-content;">/q[n b r]</div>	<p>Instructs Windows Installer to run without a graphical interface (silent install).</p> <p>If <code>/q</code> is omitted from the command line, the Workload Automation 5 installation is started from the command line, but run with a graphical interface. This is useful when an installation log file is desired.</p> <p>The following flags are available for <code>/q</code>:</p> <ul style="list-style-type: none"> <li>• <b>n</b> – Creates no user interface.</li> <li>• <b>b</b> – Creates a basic user interface, which displays a simple dialog box to report progress and any errors.</li> <li>• <b>r</b> – Creates a reduced user interface, which displays progress and errors in customized windows.</li> </ul> <p>See <a href="#">Windows Installer Command Line Syntax</a>, <a href="#">Workload Automation 5 Command Line Installation Examples</a>, and <a href="#">Detecting the Completion of a Silent Install</a> for additional information regarding silent installs.</p>	<p>n/a</p>
<div style="border: 1px solid black; padding: 5px; width: fit-content;">/L*v</div>	<p>Instructs Windows Installer to create an installation log file named <code>&lt;logfilepath&gt;</code> (full path name). If <code>&lt;logfilepath&gt;</code> contains spaces, you must enclose it with double ( " ) quotation marks.</p> <p><code>*v</code> are flags used to specify the level of detail (<b>verbose</b>) contained in the log file. To reduce the amount of output generated, <code>*v</code> can be omitted. However, using these options is good practice; they can assist Stonebranch Customer Support with problem determination should any errors occur during installation.</p>	<p>n/a</p>

## Workload Automation 5 Command Line Installation Examples

The following examples demonstrate different ways to install Workload Automation 5 from the command line.

To install all 32-bit Workload Automation 5 components using a full graphical interface, simply execute the 32-bit distribution file and follow the instructions:

```
sb-5.1.0.0-windows-i386.exe
```

To install all 64-bit Workload Automation 5 components silently using the 64-bit distribution file while writing an installation log to `C:\Temp\install.log`, issue the following command:

```
sb-5.1.0.0-windows-x64.exe /s /v"/L*v C:\Temp\install.log /qn"
```

To install only the Universal Broker (which always is installed), Universal Data Mover Manager, Universal Event Monitor 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 32-bit Workload Automation 5 distribution file, issue the following command:

```
sb-5.1.0.0-windows-i386.exe /v"INSTALLDIR=D:\Universal UAGSRV=no UCMDMGR=no UCMSRV=no UCTLSRV=no UELD=no USPOOL=no UDMSRV=no UEMSRV=no"
```

To install all 32-bit Workload Automation 5 components using the extracted **.msi** file and to set the Universal Automation Center Agent's (UAG) **AUTOMATION\_CENTER\_TRANSPORTS** and **AUTOMATION\_CENTER\_CORE** configuration options, issue the following command:

```
SetupPath\UCmd.msi AC_TRANSPORTS=4803@opshost AC_CORE=HUB01
```

To install all 64-bit Workload Automation 5 components using the extracted **.msi** file while setting the Universal Broker service to execute with an administrative account with the specified user ID and password, issue the following command:

```
SetupPath\UCmdx64.msi RUNBROKERASUSER=1 BROKERUID=ubradmin BROKERPWD=guesme
```

### 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 installation process ends is to execute it using the system's **start** command. Using available command line switches, **start** can be used to initiate the installation 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 perform a silent 64-bit Workload Automation 5 installation using the 64-bit distribution file and wait for it to finish:

```
start /b /wait sb-5.1.0.0-windows-x64.exe /w /s /v"/qn"
```

- The **/b** switch prevents the **start** command from opening a new window.
- The **/wait** parameter causes the **start** command to start the installation and then wait for it to finish.

This also is the recommended way to execute the installation from within a batch file.

For more information on the **start** command, go to the Windows command prompt and enter: **start /?**

## Modifying a Workload Automation 5 Installation via the Command Line Interface

- [Modifying a Workload Automation 5 Installation via the Windows Installer Command Line Interface](#)
- [Adding or Removing Workload Automation 5 Components](#)
- [Repairing a Corrupted Workload Automation 5 Installation](#)
- [Removing a Workload Automation 5 Installation](#)

### Modifying a Workload Automation 5 Installation via the Windows Installer Command Line Interface

This page describes how to modify a Workload Automation 5 installation via the Windows Installer command line interface.

After Workload Automation 5 is installed, Windows Installer can be run as many times as needed to modify the installation by:

- Adding or Removing Workload Automation 5 Components
- Repairing a Corrupted Workload Automation 5 Installation
- Removing a Workload Automation 5 Installation

(For a description of the parameters used in these procedures, see [Windows Installer Command Line Parameters](#).)

### Adding or Removing Workload Automation 5 Components

Currently, it only is possible to add or remove individual Workload Automation 5 components using the Windows Installer graphical interface (see [Modifying a Workload Automation 5 Installation via the Graphical Interface](#)).

### Repairing a Corrupted Workload Automation 5 Installation

To recover accidentally deleted files or registry entries required by Workload Automation 5 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 Workload Automation 5 distribution file.

To repair a Workload Automation 5 installation from the command line using the original distribution file, issue the following command:

```
sb-5.1.0.<level>-windows-<platform>.exe /v"/fom /q /L*v c:\temp\repair.log"
```

In this command, `<level>` is the numeric package level and `<platform>` is `i386` for the 32-bit distribution file and `x64` for the 64-bit distribution file.

To repair a 32-bit Workload Automation 5 installation using the `.msi` file that was extracted from the distribution file during the initial install, issue the following command:

```
msiexec.exe /fom SetupPath\Ucmd.msi /q /L*v c:\temp\repair.log
```

To repair a 64-bit Workload Automation 5 installation using the `.msi` file that was extracted from the distribution file during the initial install, issue the following command:

```
msiexec.exe /fom SetupPath\Ucmdx64.msi /q /L*v c:\temp\repair.log
```

In the last two examples, `SetupPath` refers to the location in which the extracted `.msi` file resides. The exact path varies, depending on the user account that performed the install and the Windows version.

For information on how the installation determines `SetupPath`, see [Step 2](#) of the Installing via Command Line procedures.

**Note**

All of these commands also use the optional `/L*v` parameter to generate a log file named `C:\temp\repair.log`. If you want to generate a log file, you can substitute the path and file name for one of your choosing. If no log file is desired, simply omit the `/L*v` option and file name.

## Removing a Workload Automation 5 Installation

To uninstall a Workload Automation 5 installation from the command line using the original distribution file, issue the following command:

```
sb-5.1.0.<level>-windows-<platform>.exe /x
```

In this command, `<level>` is the numeric package level and `<platform>` is `i386` for the 32-bit distribution file and `x64` for the 64-bit distribution file.

To uninstall a 32-bit Workload Automation 5 installation using the `.msi` file that was extracted from the distribution file during the initial install, issue the following command:

```
msiexec.exe /x SetupPath\Ucmd.msi /q
```

To uninstall a 64-bit Workload Automation 5 installation using `.msi` file that was extracted from the distribution file during the initial install, issue the following command:

```
msiexec.exe /x SetupPath\Ucmdx64.msi /q
```

In the last two examples, `SetupPath` refers to the location in which the extracted `.msi` file resides. The exact path varies depending on the user account that performed the install and the Windows version.

For information on how the installation determines `SetupPath`, see [Step 2](#) of the Installing via Command Line procedures.

## Migrating between 32- and 64-bit Workload Automation 5 for Windows Installs

- [32- and 64-bit Workload Automation for Windows Crossgrades](#)
- [Performing a 32- to 64-bit Workload Automation for Windows Crossgrade](#)
  - [Executing a 32- to 64-bit Workload Automation Crossgrade from the Command Line](#)
- [Performing a 64- to 32-bit Workload Automation for Windows Crossgrade](#)
  - [Executing a 64- to 32-bit Workload Automation Crossgrade from the Command Line](#)

### 32- and 64-bit Workload Automation for Windows Crossgrades

The Workload Automation for Windows installation supports migrations between 32- and 64-bit versions. This migration is not an upgrade in the strictest sense, because the installed location of some application files is likely to change.

However, because the migration will preserve configuration settings, it does possess some characteristics of an upgrade. Therefore, migrations from a 32-bit to a 64-bit Workload Automation installation (or from a 64-bit to a 32-bit installation) are referred to throughout this document as a 32-bit / 64-bit crossgrade.

(See [Migrating an Opwise Agent to UAG](#) for information on crossgrade from an Opwise 1.5, 1.6, or 1.7 Agent install to a Workload Automation 5 install with the new Universal Automation Center Agent.)

The Workload Automation installation supports the following 32-bit / 64-bit crossgrade scenarios:

- From any 32-bit Universal Products (3.2.0.0 and later) or Workload Automation install to a 64-bit Workload Automation install.
- From any 64-bit Workload Automation install to a 32-bit Workload Automation 5 or later install.

If a 64-bit Workload Automation package currently is installed, and you want to return to a 32-bit install prior to Workload Automation 5 (that is, 5.1.0.0), you first must remove the 64-bit Workload Automation installation and do a clean install of the 32-bit package. No configuration settings are preserved in this scenario.



#### Note

A crossgrade from a 32-bit Universal Products package prior to 3.2.0.0 is not currently supported. Configuration information for releases prior to 3.2.0.0 was stored in the Windows system registry, and crossgrades to a 64-bit Workload Automation install only preserve options that are stored in text-based configuration files.

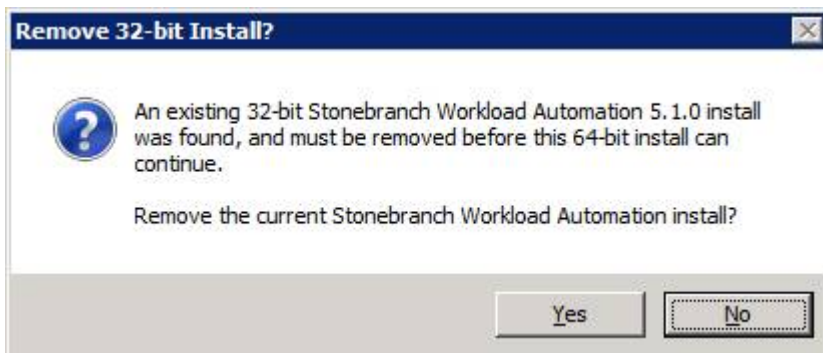
To migrate from a 32-bit Universal Products install that is older than 3.2.0.0, first do an upgrade to the 32-bit Workload Automation package (which will dump configuration options stored in the registry to our text-based .conf files), and then do the 64-bit crossgrade.

### Performing a 32- to 64-bit Workload Automation for Windows Crossgrade

To begin a 32- to 64-bit Workload Automation for Windows crossgrade:

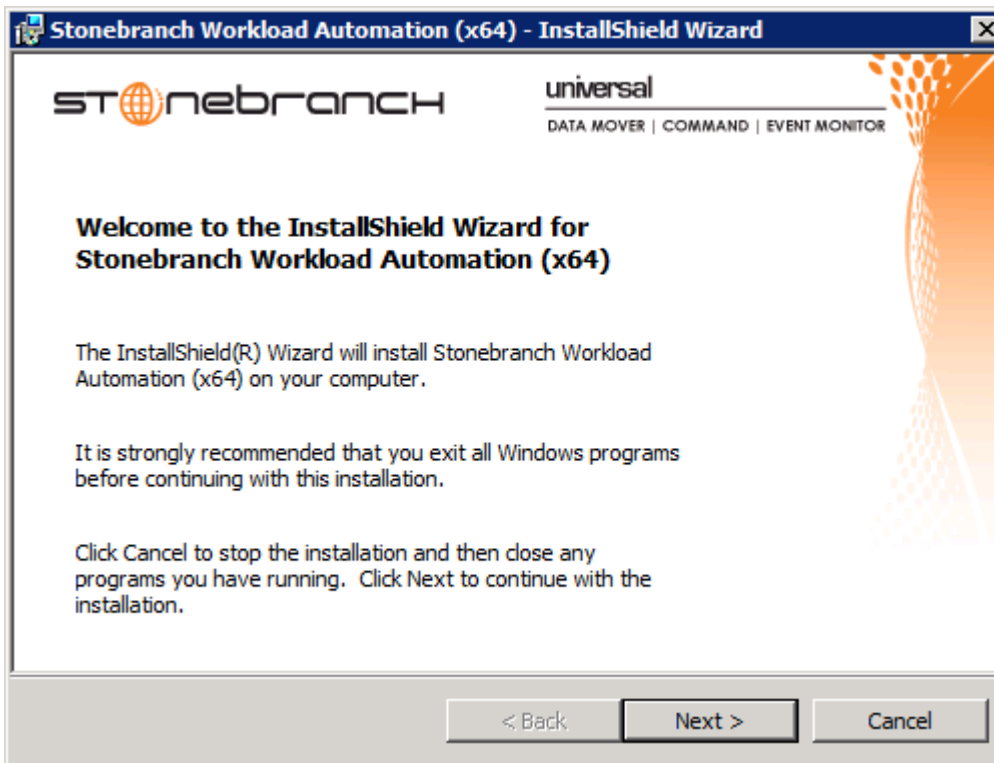
**Step 1** Execute the 64-bit install package as described in [Installing Workload Automation 5 via the Graphical Interface](#).

If you are performing an interactive install and the Workload Automation install detects an existing 32-bit Universal Products or Workload Automation installation, the install will display the following message:



**Step 2** Click **<Yes>** to save all configured settings and remove the existing 32-bit install.

Once the removal of the 32-bit install is complete, the 64-bit install will resume with the following dialog:



Notice the **(x64)** label, which identifies this as the 64-bit Workload Automation install. Because no Workload Automation install exists on the system at this point, the installation will proceed as would a clean install.

This means some information – the Universal Broker service account and Universal server component working directories, for example – will need to be reentered or specified via command line properties (see [Windows Installer Command Line Parameters](#)).

Upon completion of a 32- to 64-bit Workload Automation crossgrade:

- User-configured settings stored in Universal component **.conf** files are retained.
- Some configuration settings that are dependent upon install locations will be updated (for example, trace file directories).
- The location of the Broker spool directory is preserved. The configuration files will still point to the 32-bit install location (for example, **C:\Program Files (x86)\Universal\spool\ubroker**) so that no persistent data (for example, Universal Event Subsystem records) is lost during the crossgrade.
- Any Workload Automation components that were installed to the **%SystemRoot%\SysWow64** will now reside as native 64-bit applications in the **%SystemRoot%\System32** directory.
- The icon for the Universal Configuration Manager Control Panel applet is still compiled as a 32-bit application and will continue to appear in the 32-bit Control Panel items. The Universal Configuration Manager can continue to be used to configure 64-bit Workload Automation components.

#### Executing a 32- to 64-bit Workload Automation Crossgrade from the Command Line

A 32- to 64-bit Workload Automation crossgrade can be executed from the command line, but it must be run with at least a reduced user interface. A reduced user interface is specified using the **/qr** Windows installer command line option. A reduced user interface still displays some installation dialogs, but allows the install to run unattended (that is, without user interaction).

If a 64-bit install detects an existing 32-bit installation, it automatically will perform a crossgrade without requiring a response to the **Remove 32-bit Install?** dialog displayed for fully interactive installs.

The command to execute a 32- to 64-bit Workload Automation crossgrade might look similar to this:

```
sb-5.1.0.0-windows-x64.exe /s /v"/qr"
```

In this command:



- **/s** option instructs the Windows installer initialization to execute silently.
- **/v** option identifies the start of the Windows installer command line switches.
- **/qr** instructs the installer to execute with a reduced user interface.

Attempts to perform a completely silent crossgrade from the command line using the **/qn** or **/qb** Windows installer switch will fail.

New installs and 64-bit upgrades from older 64-bit installations still can be done silently.

See [Installing Workload Automation 5 via the Command Line Interface](#) for more information regarding command line installs.

### Performing a 64- to 32-bit Workload Automation for Windows Crossgrade

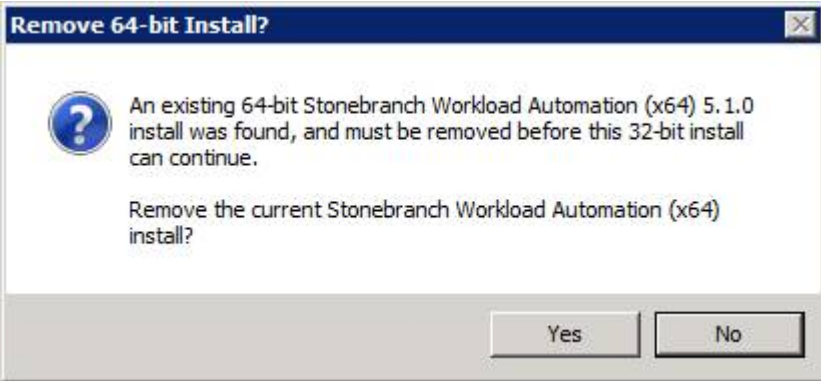
Support for 64- to 32-bit crossgrades is only available for 32-bit Workload Automation 5 or later install packages. Earlier versions of the 32-bit Universal Products install do not have the ability to detect a 64-bit Workload Automation install.

To return to a 32-bit Universal Products installation (that is, a version prior to 5.1.0.0) when a 64-bit Workload Automation package is currently installed, you must uninstall the 64-bit Workload Automation package first and then do a clean install of the earlier 32-bit package.

To begin a 64- to 32-bit Workload Automation for Windows crossgrade:

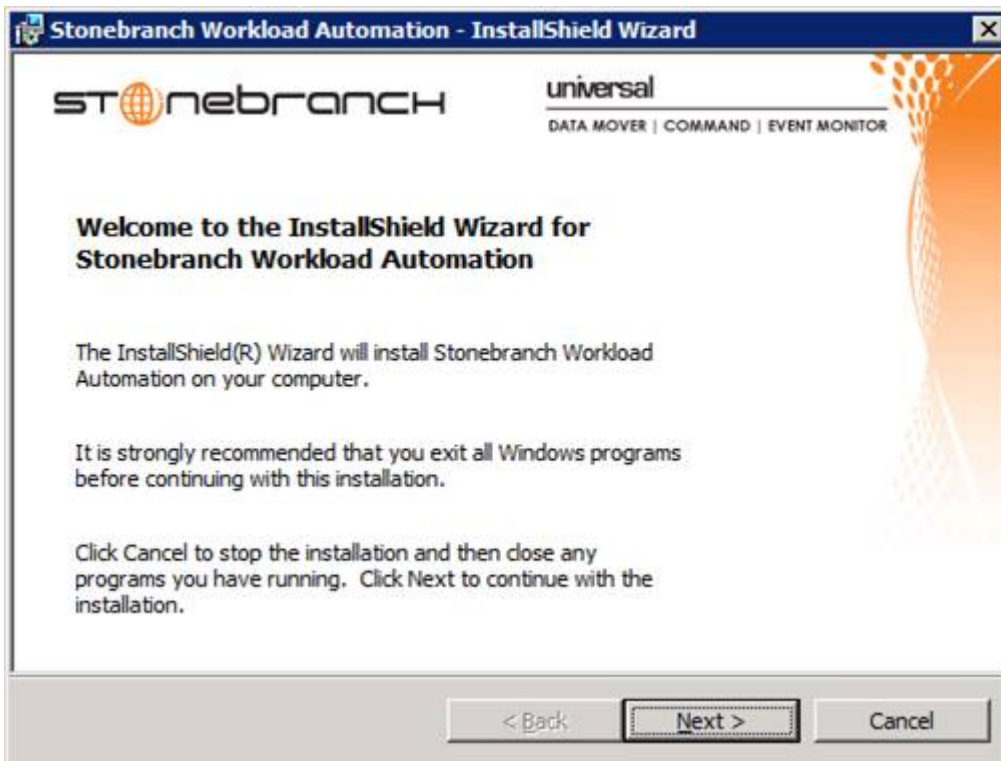
**Step 1** Execute the 32-bit Workload Automation install package as described in [Workload Automation 5 for Windows - Installation Procedures](#).

If you are performing an interactive install and the Workload Automation install detects an existing 64-bit Universal Products or Workload Automation installation, the install will display the following message:



**Step 2** Select **<Yes>** to save all configured settings and remove the existing 64-bit install.

Once the removal of the 64-bit install is complete, the 32-bit install will resume with the following dialog:



Notice the absence of the **(x64)** label, which identifies this as the 32-bit Workload Automation install.

Because no Workload Automation install exists on the system at this point, the installation will proceed as would a clean install. This means some information - the Universal Broker service account and Universal server component working directories, for example - will need to be reentered or specified via command line properties (see [Windows Installer Command Line Parameters](#)).

Upon completion of a 64- to 32-bit Workload Automation crossgrade:

- User-configured settings stored in Universal component **.conf\*** files are retained.
- Some configuration settings that are dependent upon install locations will be updated (for example, trace file directories).
- The location of the Broker spool directory is preserved. The configuration files will still point to the 32-bit install location (for example, **C:\Program Files (x86)\Universal\spool\ubroker**) so that no persistent data (for example, Universal Event Subsystem records) is lost during the crossgrade.
- Any Workload Automation components that were installed to the **%SystemRoot%\SysWow64** will now reside as native 64-bit applications in the **%SystemRoot%\System32** directory.
- The icon for the Universal Configuration Manager Control Panel applet is still compiled as a 32-bit application and will continue to appear in the 32-bit Control Panel items. The Universal Configuration Manager can continue to be used to configure 64-bit Workload Automation components.

#### Executing a 64- to 32-bit Workload Automation Crossgrade from the Command Line

A 64- to 32-bit Workload Automation crossgrade can be executed from the command line, but it must be run with at least a reduced user interface.

A reduced user interface is specified using the **/qr** Windows installer command line option. A reduced user interface still displays some installation dialogs, but allows the install to run unattended (that is, without user interaction).

If a 32-bit install detects an existing 64-bit installation, it automatically will perform a crossgrade without requiring a response to the **Remove 64-bit Install?** dialog displayed for fully interactive installs.

The command to execute a 64- to 32-bit Workload Automation crossgrade would be similar to this:

```
sb-5.1.0.0-windows-i386.exe /s /v"/qr"
```

In this command:

- **/s** option instructs the Windows installer initialization to execute silently
- **/v** option identifies the start of the Windows installer command line switches
- **/qr** instructs the installer to execute with a reduced user interface.

Attempts to perform a completely silent crossgrade from the command line using the **/qn** or **/qb** Windows installer switch will fail.

New installs and 32-bit upgrades from older 32-bit installations still can be done silently.

See [Installing Workload Automation 5 via the Command Line Interface](#) for more information regarding command line installs.

## Migrating an Opswise Agent to UAG for Workload Automation 5 for Windows

- [Overview](#)
- [UAG Crossgrade Script](#)
  - [UAG Crossgrade Script Actions](#)
  - [Executing a UAG Crossgrade After a Workload Automation Install](#)

### Overview

The Universal Automation Center Agent (UAG) provided by the Workload Automation 5 installation replaces all previous versions (1.7 and earlier) of the Opswise Automation Center Agent.

When instructed to do so, the Workload Automation 5 installation will convert an Opswise Agent's existing settings to the UAG configuration file (**uags.conf**). The install also can disable the existing Opswise Agent in favor of the UAG server.

This migration commonly is referred to throughout these pages as a UAG crossgrade, or simply, crossgrade.



#### Note

The installation procedures for Windows may contain information pertinent to this upgrade (see [Workload Automation 5 for Windows - Installation Procedures Overview](#)).

### UAG Crossgrade Script

The crossgrade is handled by a script named **opsmerge.vbs**, which is installed to **.\Universal\UagSrv\bin**.

The script is executed from the install only when the **CONVERT\_OPSAGENT=yes** command line option is specified.

For example, to install Workload Automation 5 from the 32-bit distribution file and migrate existing Opswise Agent settings, enter the following from the command prompt:

```
sb-5.1.0.0-windows-i386.exe /v"CONVERT_OPSAGENT=yes"
```

To install the 64-bit Workload Automation 5 package so that existing Opswise Agent settings are preserved, enter the following from the command prompt:

```
sb-5.1.0.0-windows-x64.exe /v"CONVERT_OPSAGENT=yes"
```

Additional command line parameters that allow additional control over the migration also are available.

The following example will not only execute a crossgrade from an existing Opswise Agent to a 64-bit Workload Automation 5 installation, the command line parameters specified after the **/v** switch will instruct the install to set the **AUTOMATION\_CENTER\_TRANSPORTS** and **AUTOMATION\_CENTER\_CORE** configuration options to **4803@opshost.com** and **HUB01**, respectively. The **/!\*** option will cause the install to generate an installation log file named **c:\temp\crossgrade.log**.

```
sb-5.1.0.0-windows-x64.exe /v"/!*v c:\temp\crossgrade.log CONVERT_OPSAGENT=yes  
AC_TRANSPORTS=4803@opshost.com AC_CORE=HUB01"
```

For a complete list of Workload Automation 5 installation command line parameters see [Windows Installer Command Line Parameters](#).

### UAG Crossgrade Script Actions

The script performs the following actions:

- Stops an active Opswise Agent Guardian service and sets its startup type from "Automatic" to "Manual."
- Copies an existing **.\cache\qname** (pre-1.6) or **.\var\qname** (1.6 and later) file from the Opswise agent install directory to

- `.\Universal\UagSrv\var`. The target directory is created if it does not already exist.
- Renames the old Opwise agent `qname` file to `qname.yymmdd-hhmm`.
- Checks the existing `.\etc\agent.props` file for the properties listed below. Each property is mapped to the `uags.conf` name listed beside it.

agent.props Property Name	uags.conf Configuration Keyword
config.loglvl	loglvl
config.txtdebug	txtdebug
config.clusters	agent_clusters
network.core	automation_center_core
network.name	netname
network.transports	automation_center_transports

The script should be able to handle any combination of whitespace in `agent.props` before/after property names and before/after values. If the value itself contains a space, it is enclosed in quotes inside the UAG configuration file.

The `opsmerge.vbs` script maps the last uncommented instance of an agent property to its corresponding UAG configuration keyword and writes the converted property to a file named `uags.conf.yymmdd-hhmm`.

- On Windows XP and Server 2003, this file resides in `%AllUsersProfile%\Application Data\Universal\conf`, where `%AllUsersProfile%` expands by default to `C:\Documents and Settings\AllUsers`.
- On Windows Vista, Server 2008, and Windows 7, the file resides in `%AllUsersProfile%\Universal\conf`, where `%AllUsersProfile%` expands by default to `C:\Program Data`.

This file becomes input to the Universal Products Install Merge utility (`upimerge`), which eventually combines the contents of the converted file with the installed UAG configuration file, `%AllUsersProfile%\Application Data\Universal\conf\uags.conf`.

If the `opsmerge.vbs` script fails, it will NOT stop the install.

Script output is written to `%temp%\opsmerge.log`.

### Executing a UAG Crossgrade After a Workload Automation Install

A UAG crossgrade can be performed after an install by executing `opsmerge.vbs` via script.

The `opsmerge.vbs` script accepts the following command line parameters:

Parameter	Description	Default
<code>/opsdir</code>	The Opwise Automation Center agent install directory.	<code>C:\opwise.</code>
<code>/arcdir</code>	The location of the <code>uags.conf.yymmdd-hhmm</code> file, used as input to the <code>upimerge</code> utility.	<b>Windows XP, Server 2003:</b> <code>%AllUsersProfile%\Application Data\Universal\conf\.archive</code> , where <code>%AllUsersProfile%</code> expands by default to <code>C:\Documents And Settings\All Users</code> .  <b>Windows Vista, Server 2008, Windows 7:</b> <code>%AllUsersProfile%\Universal\conf\.archive</code> , where <code>%AllUsersProfile%</code> expands by default to <code>C:\ProgramData</code> .
<code>/qnamedir</code>	The location of the <code>qname</code> file that UAG uses will be stored.	<code>%ProgramFiles%\Universal\UagSrv\var</code> .
<code>/help</code>	Displays command line help.	n/a

To execute the script with `cscript`, use the following syntax:

```
cscript opsmerge.vbs [/opsdir:dir] [arcdir:dir] [/qnamedir:dir] [//nologo]
```

## 32-Bit Workload Automation 5 for Windows on 64-Bit Windows Systems

- [Overview](#)
- [64-bit Windows Systems](#)
- [Modifying the Working Folder for Universal Server Components](#)
- [Applications Installed in the Windows System Folder](#)
  - [Example 1](#)
  - [Example 2](#)
- [Identifying 32- and 64-bit Workload Automation 5 Installations](#)

### Overview

The information on this page applies only to 32-bit Workload Automation 5 packages installed on 64-bit Windows systems.

A native 64-bit Workload Automation 5 for Windows package, which was made available starting with the 5.1.0 release, requires no special handling on 64-bit Windows systems.

### 64-bit Windows Systems

All Workload Automation 5 components have been tested and verified on the 64-bit editions of the following Windows systems:

- Windows Server 2003 SP1 and higher
- Windows XP SP3
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows Server 2008 R2

This page describes some modifications that may need to be made to the default installation options to ensure that the installed Workload Automation 5 function correctly.

### 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)`.

The default installation folder for 32-bit Windows applications developed by Stonebranch is `C:\Program Files (x86)\Universal`. The Workload Automation 5 installation package should have no problems installing its applications to this directory.

Supported Stonebranch releases prior to 5.1.0, installed on 64-bit editions of Windows prior to XP SP2 or Server 2003 SP1, may receive this error - '`C:\Program`' is not recognized as an internal or external command, operable program or batch file - under the following circumstances:

- Universal Command Manager is run using the `-script` option.
- Universal Event Monitor Server invokes an event handler that executes a script.



#### Note

Stonebranch releases starting with 5.1.0 are only supported on Windows versions where the above error is not expected to occur.

The error above 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.

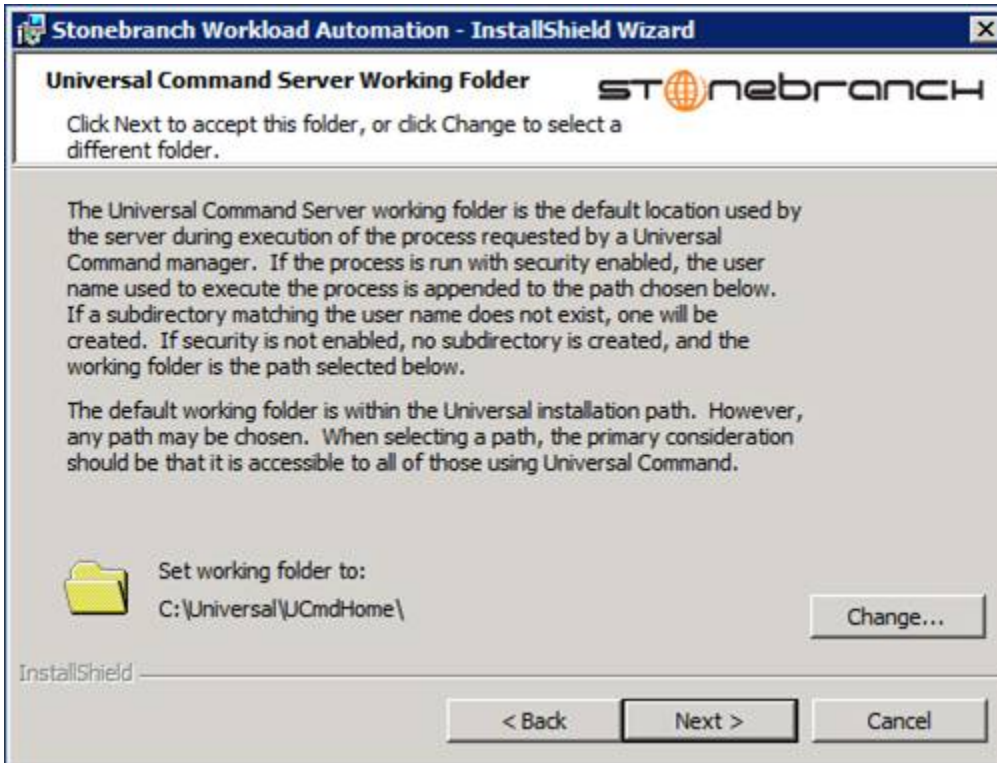
To resolve this issue, it is necessary to change the value (or location) of the Universal Command Server and/or Universal Event Monitor Server working directory.

Either of the following changes can be made:

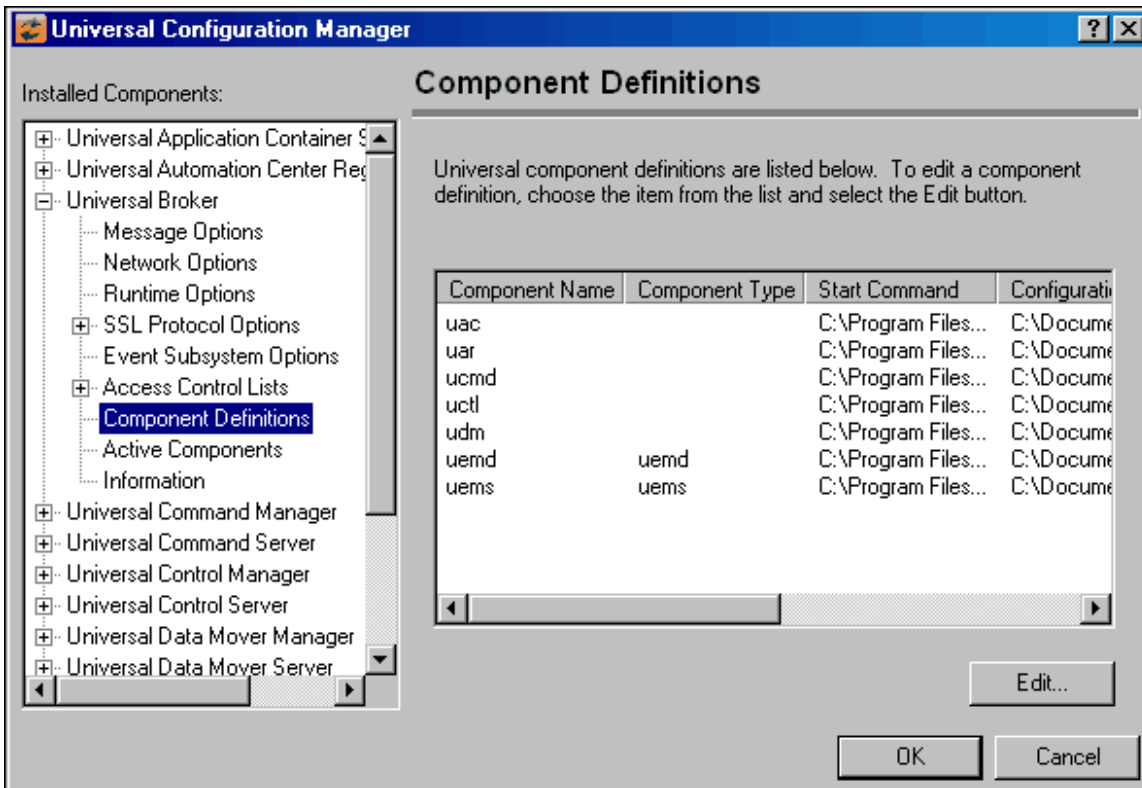
- If short path names are permitted on your system, use the `dir /x` command from the command prompt to find the short name of the `Program Files (x86)` directory (for example, `Progra~2`). Use this value as part of the working directory's path name.
- Change the working directory to a location outside of the default installation path (for example, `C:\Universal`). 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.



2. Any time after installation via the Universal Configuration Manager, on the Component Definitions page for Universal Broker.



## Applications Installed in the Windows System Folder

The 32-bit Workload Automation 5 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`).

The following table identifies the affected Workload Automation 5 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

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:

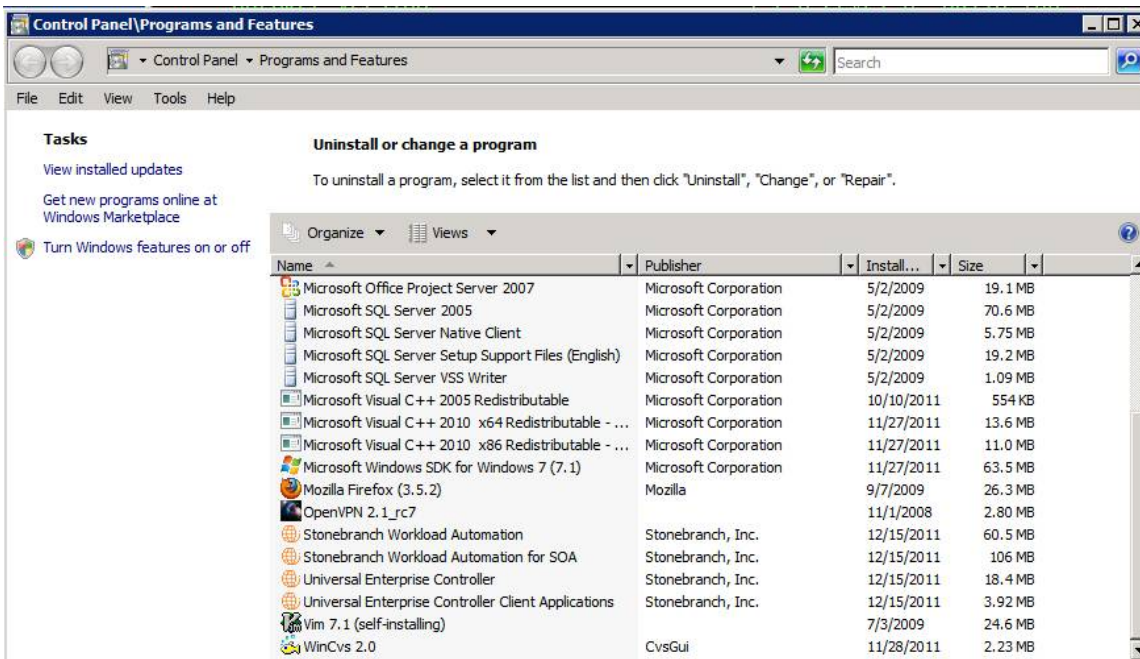
```
%SystemRoot%\SysWOW64\cmd.exe /C %SystemRoot%\SysWOW64\ucopy
```

## Identifying 32- and 64-bit Workload Automation 5 Installations

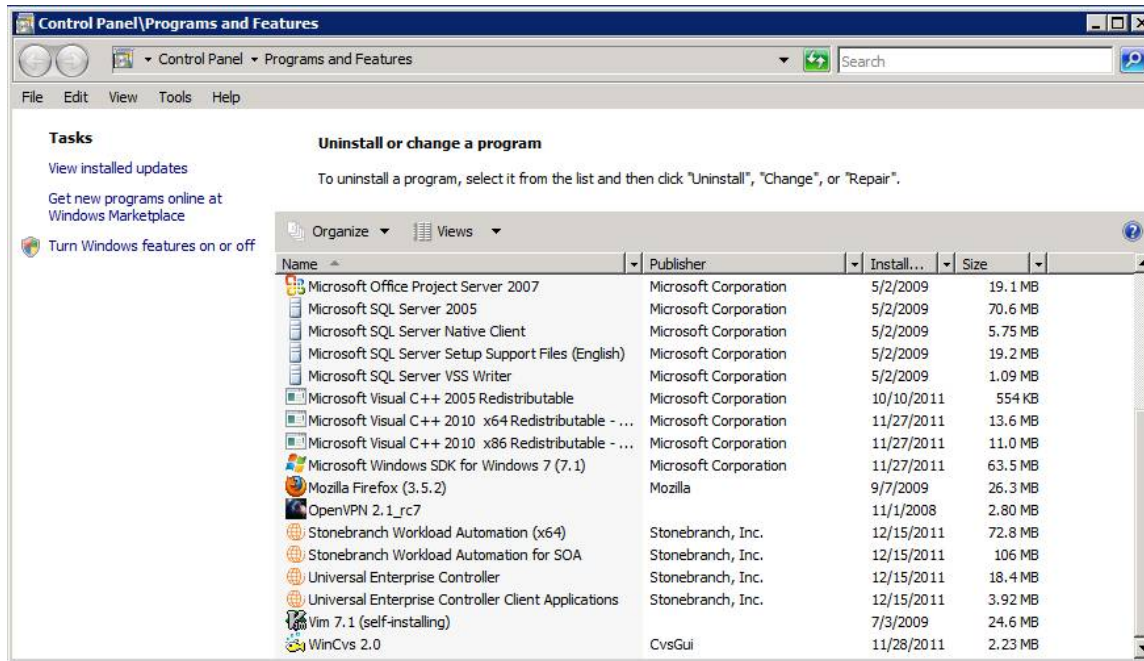
To help identify whether a 32- or 64-bit Workload Automation 5 installation exists on your system, look for the **(x64)** indicator in the package name or file version information. If it is present, a 64-bit version of Workload Automation for Windows package is installed. If it is missing, the 32-bit version of the Workload Automation for Windows package is installed.

For example, the screen shots below show 32- and 64-bit Workload Automation for Windows installs as they appear in Programs and Features on a Windows Server 2008 system.

### 32-bit Workload Automation Install



### 64-bit Workload Automation Install (with x64 label)



The **(x64)** label also appears in a component's version information, which is obtained by executing the component with the `-v` command line option.

For example, a 64-bit Universal Command Manager displays version information similar to that shown below when the `-v` option is passed to `ucmd.exe`:

```
C:\Users\Administrator>ucmd -v
ucmd 5.1.0 Level 0 Release Build 100 (x64) 11/22/11 17:24:49
xps 5.1.0 Level 0 Build 100 11/22/11 17:18:02
(c) Copyright 2000-2011 Stonebranch, Inc. All rights reserved.
```

The 32-bit version of Universal Command Manager displays the same version information, but the **(x64)** label is not shown.

## Workload Automation 5 for Windows - File Inventory Lists

- Workload Automation 5 for Windows - File Inventory Lists
- Universal Automation Center Agent
- Universal Broker
- Universal Command Manager
- Universal Command Server
- Universal Connector
- Universal Control Manager
- Universal Control Server
- Universal Data Mover Manager
- Universal Data Mover Server
- Universal Event Monitor Manager
- Universal Event Monitor Server
- Universal Configuration Manager
- Universal Copy
- Universal Encrypt
- Universal Event Log Dump
- Universal Message Translator
- Universal Products Install Merge
- Universal Query
- Universal Spool Utilities
- System Files

### Workload Automation 5 for Windows – File Inventory Lists

The Workload Automation 5.1.0 for Windows package includes the files required for the following components / utilities:

- Universal Automation Center Agent
- Universal Broker
- Universal Command Manager
- Universal Command Server
- Universal Configuration Manager
- Universal Connector
- 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 page lists the files installed with each Workload Automation 5 for Windows 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 Windows system folder. The actual name of this directory depends on the Windows version. It may also depend on whether the 32-bit or the 64-bit Workload Automation package is installed:

- For all supported 32-bit Windows editions, the path is `\Windows\System32`.
- For all supported 64-bit Windows editions when the 64-bit Workload Automation package is installed, the path is `\Windows\System32`.
- For all supported 64-bit Windows editions when the 32-bit Workload Automation package is installed, the path is `\Windows\SysWow64`.

### Universal Automation Center Agent

File	Description
UAGSrv\bin\uagsrv.exe	Universal Automation Center Agent Application program.

<b>UAGSrv\bin\uagscfg.dll</b>	Universal Automation Center Agent configuration control.
<b>UAGSrv\bin\uagscfg.hlp</b>	Universal Automation Center Agent bootstrap configuration control help file.
<b>UAGSrv\bin\ops_copyfile.exe</b>	Universal Automation Center Agent copy utility.
<b>UAGSrv\bin\ops_scan.exe</b>	Universal Automation Center Agent scan utility.
<b>UAGSrv\bin\opsmerge.vbs</b>	Visual Basic script used to migrate an existing 1.5, 1.6, or 1.7 OpsWise agent to UAG.
<b>UAGSrv\bin\ftp\bin\cygwin1.dll</b>	Cygwin library, version 1005.25.0.0 (provided to ensure compatibility with the Cygwin FTP client).
<b>UAGSrv\bin\ftp\bin\ftp.exe</b>	Cygwin FTP client, version 1.3.2.
<b>UAGSrv\bin\sftp\bin\cygcrypto-0.9.8.dll</b>	Cygwin encryption library
<b>UAGSrv\bin\sftp\bin\cygcc_s-1.dll</b>	Cygwin GCC library
<b>UAGSrv\bin\sftp\bin\cygssp-0.dll</b>	Cygwin GCC Static Stack Protection library
<b>UAGSrv\bin\sftp\bin\cygwin1.dll</b>	Cygwin library, version 1007.5.0.0
<b>UAGSrv\bin\sftp\bin\cygz.dll</b>	Cygwin Zlib library
<b>UAGSrv\bin\sftp\bin\expect.exe</b>	Interactive program automation utility
<b>UAGSrv\bin\sftp\bin\sftp.exe</b>	Secure FTP client
<b>UAGSrv\bin\sftp\bin\sftp.exp</b>	SFTP sample script for Expect utility
<b>UAGSrv\bin\sftp\bin\ssh.exe</b>	Secure shell client
<b>UAGSrv\bin\sftp\library\init.tcl</b>	TCL initialization script
<b>UBroker\tml\luagcmp</b>	Template file for the UAG component definition.
<b>UBroker\tml\luagcfg</b>	Template file for the UAG configuration.

nls\luagmceng.umf	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\confluags.conf	UAG configuration file.
%ALLUSERSPROFILE%\Application Data\Universal\compluag	UAG component definition file.

## 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 Workload Automation database files.
UBroker\tml	XML template files used by I-Management Console for remotely configuring Workload Automation.
nls\*.utt	Code page files used for text translation between different operating systems and platforms.
nls\usmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\confluacl.conf	Universal Access Control List (ACL) configuration file. <b>%ALLUSERSPROFILE%</b> is a system environment variable that expands to the All Users profile directory, typically <b>C:\Documents and Settings\All Users</b> .
%ALLUSERSPROFILE%\Application Data\Universal\conflubroker.conf	Universal Broker configuration file. <b>%ALLUSERSPROFILE%</b> is a system environment variable that expands to the All Users profile directory, typically <b>C:\Documents and Settings\All Users</b> .

## 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\lucmmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\conflucmd.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.

## 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\lucmmceng.umc	English message catalog.
%ALLUSERSPROFILE%\Application Data\Universal\complucmd	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\conflucmds.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.

## Universal Connector

File	Description
\USap\bin\usap.exe	Universal Connector application program.
\USap\bin\uspcfg.dll	Used by Universal Configuration Manager to manage Universal Connector properties.
\USap\bin\uspcfg.hlp	Universal Connector configuration control context-sensitive help file.
\USap\bin\librfc32.dll	SAP Remote Function Call (RFC) library, v7100. <ul style="list-style-type: none"> <li>For 32-bit packages, the file version is 7100.3.143.6091.</li> <li>For 64-bit packages, the file version is 7100.1.269.7182.</li> </ul>
\UBroker\tml\uspcfg	Universal Connector remote configuration template.
\nls\uspmceng.umc	Universal Connector English message catalog.
\nls*.stt	Spoolist translation table files.
%ALLUSERSPROFILE%\Application Data\Universal\conf\usap.conf	Universal Connector configuration file.

## Universal Control Manager

File	Description
System32\uctl.exe	Application program.
UCtlMgr\bin\uctccfg.dll	Used by Universal Configuration Manager to manage Universal Control Manager properties.
UCtlMgr\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	Universal Control Manager configuration file. %ALLUSERSPROFILE% is a system

<b>Data\Universal\confuctl.conf</b>	environment variable that expands to the All Users profile directory, typically <b>C:\Documents and Settings\All Users</b> .
-------------------------------------	--

## Universal Control Server

File	Description
<b>UCtlSrv\bin\luctsrv.exe</b>	Application program.
<b>UCtlSrv\bin\luctscfg.dll</b>	Used by Universal Configuration Manager to manage Universal Control Server properties.
<b>UCtlSrv\bin\luctscfg.hlp</b>	Universal Control Server configuration help file.
<b>Nls\*.utt</b>	Code page files used for text translation between different operating systems and platforms.
<b>Nls\luctmceng.umc</b>	English message catalog.
<b>%ALLUSERSPROFILE%\Application Data\Universal\compluctl</b>	Universal Control Server component definition file. <b>%ALLUSERSPROFILE%</b> is a system environment variable that expands to the All Users profile directory, typically <b>C:\Documents and Settings\All Users</b> .
<b>%ALLUSERSPROFILE%\Application Data\Universal\confuctls.conf</b>	Universal Control Server configuration file. <b>%ALLUSERSPROFILE%</b> is a system environment variable that expands to the All Users profile directory, typically <b>C:\Documents and Settings\All Users</b> .

## Universal Data Mover Manager

File	Description
<b>System32\udm.exe</b>	Application program.
<b>UdmMgr\bin\udmccfg.dll</b>	Used by Universal Configuration Manager to manage Universal Data Mover Manager properties.
<b>UdmMgr\bin\udmccfg.hlp</b>	Universal Data Mover Manager configuration help file.
<b>nls\*.utt</b>	Code page files used for text translation between different operating systems and platforms.
<b>nls\udmmceng.umc</b>	English message catalog.
<b>%ALLUSERSPROFILE%\Application</b>	Universal Data Mover Manager configuration file. <b>%ALLUSERSPROFILE%</b> is a



<b>Data\Universal\conf\udm.conf</b>	system environment variable that expands to the All Users profile directory, typically <b>C:\Documents and Settings\All Users</b> .
-------------------------------------	---

## Universal Data Mover Server

File	Description
<b>UdmSrv\bin\udmsrv.exe</b>	Application program.
<b>UdmSrv\bin\udmscfg.dll</b>	Used by Universal Configuration Manager to manage Universal Data Mover Server properties.
<b>UdmSrv\bin\udmscfg.hlp</b>	Universal Data Mover Server configuration help file.
<b>nls*.utt</b>	Code page files used for text translation between different operating systems and platforms.
<b>nls\udmmceng.umc</b>	English message catalog.
<b>%ALLUSERSPROFILE%\Application Data\Universal\compludm</b>	Universal Data Mover Server component definition file. <b>%ALLUSERSPROFILE%</b> is a system environment variable that expands to the All Users profile directory, typically <b>C:\Documents and Settings\All Users</b> .
<b>%ALLUSERSPROFILE%\Application Data\Universal\conf\udms.conf</b>	Universal Data Mover Server configuration file. <b>%ALLUSERSPROFILE%</b> is a system environment variable that expands to the All Users profile directory, typically <b>C:\Documents and Settings\All Users</b> .

## Universal Event Monitor Manager

File	Description
<b>System32\uem.exe</b>	Application program.
<b>UemMgr\bin\uemccfg.dll</b>	Used by Universal Configuration Manager to manage Universal Event Monitor Manager properties.
<b>UemMgr\bin\uemccfg.hlp</b>	Universal Event Monitor Manager configuration help file.
<b>nls*.utt</b>	Code page files used for text translation between different operating systems and platforms.
<b>nls\uemmceng.umc</b>	English message catalog.
<b>%ALLUSERSPROFILE%\Application Data\Universal\conf\uem.conf</b>	Universal Event Monitor Manager configuration file. <b>%ALLUSERSPROFILE%</b> is a system environment variable that expands to the All Users profile directory, typically

C:\Documents and Settings\All Users.

## Universal Event Monitor Server

File	Description
System32\uemload.exe	Event definition and event handler database load utility.
UemSrv\bin\uemsvr.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\compluemd	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\compluems	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\confluems.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.

## Universal Configuration Manager

File	Description
UCfgMgr\bin\ucfgmgr.cpl	Universal Configuration Manager control panel application.
UCfgMgr\bin\ucfgmgr.cpl	Universal Configuration Manager help file.

## Universal Copy

File	Description

<b>System32\ucopy.exe</b>	Utility used for binary file copies. Similar to the UNIX cat command. Installed only if Universal Command Server is installed.
---------------------------	--

## Universal Encrypt

File	Description
<b>System32\uencrypt.exe</b>	Application program. Installed only if Universal Command Manager is installed.

## Universal Event Log Dump

File	Description
<b>System32\ueld.exe</b>	Application program.
<b>UEld\bin\uelcfg.dll</b>	Used by Universal Configuration Manager to manage Universal Event Log Dump properties.
<b>UEld\bin\uelcfg.hlp</b>	Universal Event Log Dump configuration help file.
<b>Nls\uelmceng.umc</b>	English message catalog.
<b>%ALLUSERSPROFILE%\Application Data\Universal\conflueld.conf</b>	Universal Event Log Dump configuration file. <b>%ALLUSERSPROFILE%</b> is a system environment variable that expands to the All Users profile directory, typically <b>C:\Documents and Settings\All Users</b> .

## Universal Message Translator

File	Description
<b>System32\umet.exe</b>	Application program, always installed. Used to map application-specific error messages to error codes.

## Universal Products Install Merge

File	Description
<b>UPIMerge\bin\upimerge.exe</b>	Application program, always installed. Provides command line access to the same functionality used by the Workload Automation installation to merge options from a new configuration file into an existing file.

## Universal Query

File	Description
------	-------------

<b>System32\luquery.exe</b>	Application program.
<b>UQuery\bin\luqrcfg.dll</b>	Used by Universal Configuration Manager to manage Universal Query properties.
<b>UQuery\bin\luqrcfg.hlp</b>	Universal Query configuration help file.
<b>Nls\*.utt</b>	Code page files used for text translation between different operating systems and platforms.
<b>Nls\luqrmceng.umc</b>	English message catalog.
<b>%ALLUSERSPROFILE%\Application Data\Universal\confluquery.conf</b>	Universal Query configuration file.  %ALLUSERSPROFILE% is a system environment variable that expands to the All Users profile directory, typically <b>C:\Documents and Settings\All Users</b> .

## Universal Spool Utilities

File	Description
<b>USpool\bin\luslist.exe</b>	Used to list the contents of Universal Spool files.
<b>USpool\bin\luslrm.exe</b>	Used to remove records from Universal Spool files.
<ul style="list-style-type: none"> <li>• <b>USpool\bin\ludb_archive.exe</b></li> <li>• <b>USpool\bin\ludb_dump.exe</b></li> <li>• <b>USpool\bin\ludb_load.exe</b></li> <li>• <b>USpool\bin\ludb_recover.exe</b></li> <li>• <b>USpool\bin\ludb_stat.exe</b></li> <li>• <b>USpool\bin\ludb_upgrade.exe</b></li> <li>• <b>USpool\bin\ludb_verify.exe</b></li> </ul>	Miscellaneous spool file utilities. Should be used only for debugging purposes, and only at the request of Stonebranch, Inc. Customer Support.

## System Files

The following files will be installed only if they are newer than the existing file.

The directories shown in the following table are relative to the **%SYSTEMROOT%** directory, where **%SYSTEMROOT%** is an environment variable that resolves to **C:\Windows** on all Windows platforms.

File	Description
<b>System32\asycfilt.dll</b>	Version 2.40.4275.1. This DLL is one of the components of the Microsoft OLE library.
<b>System32\comcat.dll</b>	Version 4.71.1460.1 of the Microsoft Component Category Manager library.

<b>Microsoft C-Runtime v8.0.</b> <sup>1</sup>	Version 8.0 of the Microsoft C runtime side-by-side assembly. <ul style="list-style-type: none"> <li>• For 32-bit packages, the file version is 8.0.50727.762.</li> <li>• For 64-bit packages, the file version is 8.0.50727.4053.</li> </ul>
<b>Microsoft Foundation Classes v8.0.50727.762</b> <sup>2</sup>	Version 8.0.50727.762 of the Microsoft Foundation Class (MFC) side-by-side assembly.
<b>System32\msiexec.exe</b>	Version 3.1.4000.1823 of the Microsoft Windows Installer (see <a href="#">Windows Installer</a> for more information).
<b>System32\oleaut32.dll</b>	Version 2.40.4275.1. This DLL is one of the components of the Microsoft OLE library.
<b>System32\olepro32.dll</b>	Version 5.0.4275.1. This DLL is one of the components of the Microsoft OLE library.
<b>System32\psapi.dll</b>	Version 4.0.1371.1 of the Microsoft process status library
<b>System32\stdole2.tlb</b>	Version 2.40.4275.1. This file is one of the components of the Microsoft OLE library.

<sup>1</sup> 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.

<sup>2</sup> 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.

## Universal Enterprise Controller for Windows

Error formatting macro: redirect: java.lang.NullPointerException

## Universal Enterprise Controller for Windows - Overview

### Overview

The following information is provided for the installation of Universal Enterprise Controller for Windows:

- [Universal Enterprise Controller for Windows - Package](#)
- [Universal Enterprise Controller for Windows - Installation Requirements](#)
- [Universal Enterprise Controller for Windows - Installation Procedures](#)
- [Universal Enterprise Controller for Windows - 64-Bit Windows Editions](#)
- [Universal Enterprise Controller for Windows - Database Configuration](#)
- [Universal Enterprise Controller for Windows - File Inventory Lists](#)

(For licensing information, see [Windows Installation - Licensing](#).)

## Universal Enterprise Controller for Windows - Package

- [Components](#)
- [Component Compatibility](#)

### Components

The Universal Enterprise Controller (UEC) 5.1.0 for Windows package includes a single component:

- Universal Enterprise Controller 5.1.0



#### 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 [UEC Client Applications - Package](#)).

### Component Compatibility

The following table identifies the compatibility of Universal Enterprise Controller 5.1.0 for Windows with previous component / product versions.

Component	Compatibility
Universal Enterprise Controller 5.1.0	<ul style="list-style-type: none"> <li>• Universal Broker 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 2.2.0.</li> <li>• Not compatible with previous versions of Universal Enterprise Controller Client Applications.</li> </ul>

The component references pertain to all supported platforms for that version.



## Universal Enterprise Controller for Windows - Installation Requirements

### System Requirements

- One of the supported Windows operating systems. Currently, the following Windows operating systems are supported for Universal Enterprise Controller for Windows:
  - Windows Server 2003 SP1 and higher
  - Windows XP SP3
  - Windows Vista
  - Windows Server 2008
  - Windows 7
  - Windows Server 2008 R2
  - Windows Server 2012
- An account with administrative privileges.
- Possible reboot: a reboot is required if the Windows Installer service is not installed, a version of the Windows Installer prior to 3.1.4000.1823 is installed, or if required files are in use at the time of the installation.
- TCP/IP.
- About 35 megabytes of disk space.

### Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opswise Automation Center 5.1.1](#) and [Indesca-Infitran 5.1.0](#) page to make sure that your platform is supported before performing an installation.

## Universal Enterprise Controller for Windows - Installation Procedures

Error formatting macro: redirect: java.lang.NullPointerException

## Universal Enterprise Controller for Windows - Installation Procedures Overview

### Overview

The following procedures are provided for the installation and modification of Universal Enterprise Controller (UEC) for Windows:

- [Installing UEC via the Graphical Interface](#)
- [Modifying a UEC Installation via the Graphical Interface](#)
- [Installing UEC via the Command Line Interface](#)
- [Modifying a UEC Installation via the Command Line Interface](#)

## Installing UEC via the Graphical Interface

### 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:

- |               |   |
|---------------|---|
| <b>Step 1</b> | Download the Universal Enterprise Controller for Windows product distribution file, <b>sb-UECtrl-5.1.0.&lt;level&gt;-windows-i386.exe</b> , to your work station. |
|---------------|---|

**Step 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](#), you can simply double-click the extracted Universal Enterprise Controller installation file, **UECtrlr.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.

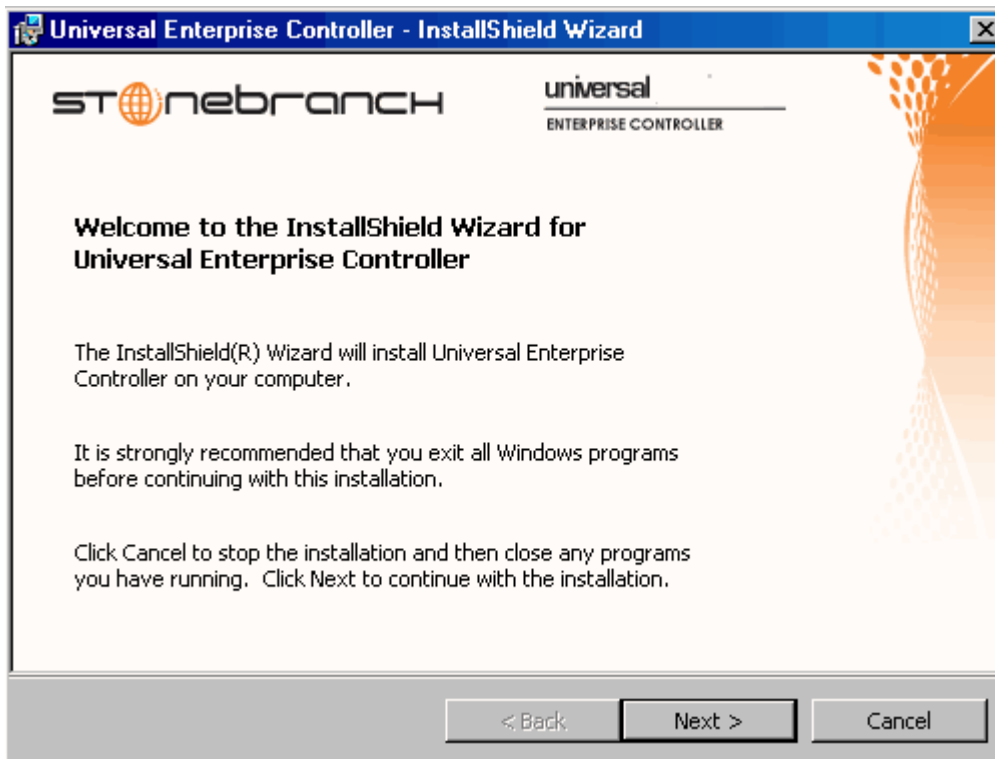
For example, the distribution file's default extraction location is based on the **TEMP** environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed.

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.

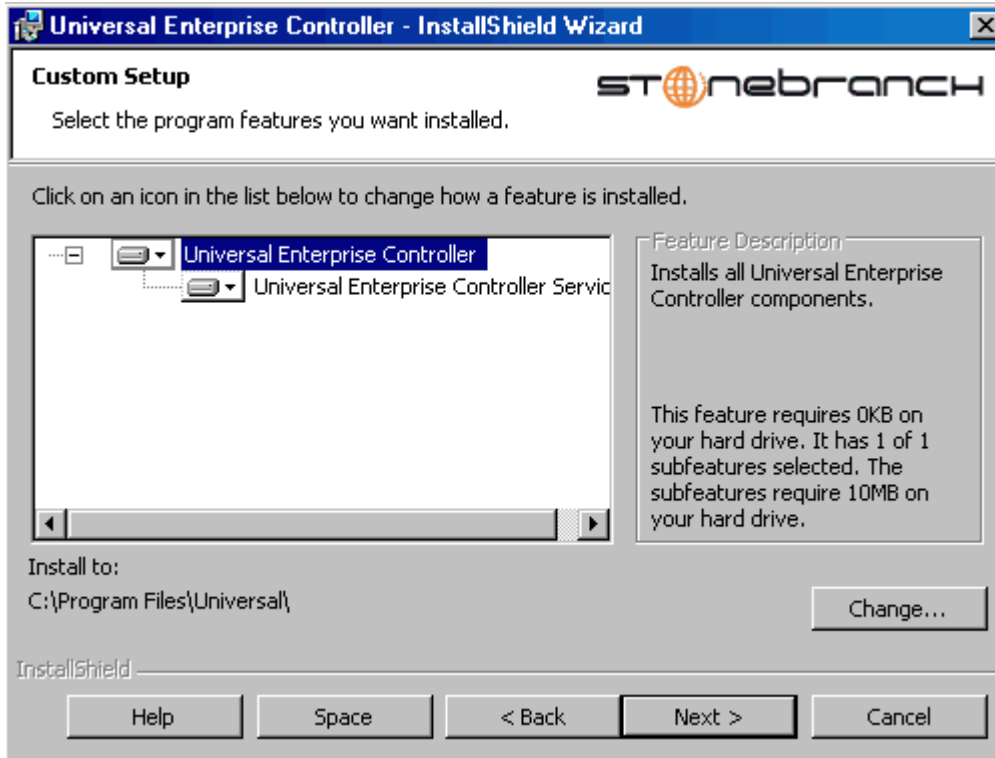
Refer to the Microsoft documentation on the Remote Desktop feature for additional information.

The installation starts after the files are extracted.

It first will verify that your machine meets the minimum system requirements (see [Universal Enterprise Controller for Windows - Installation Requirements](#)). If the requirements are met, a Welcome dialog displays.



- Step 3** Click the **Next** button. A list of Universal Enterprise Controller components included in the installation package then displays. It is from this list that you can select which components to install.



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.



#### **A Stonebranch 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.

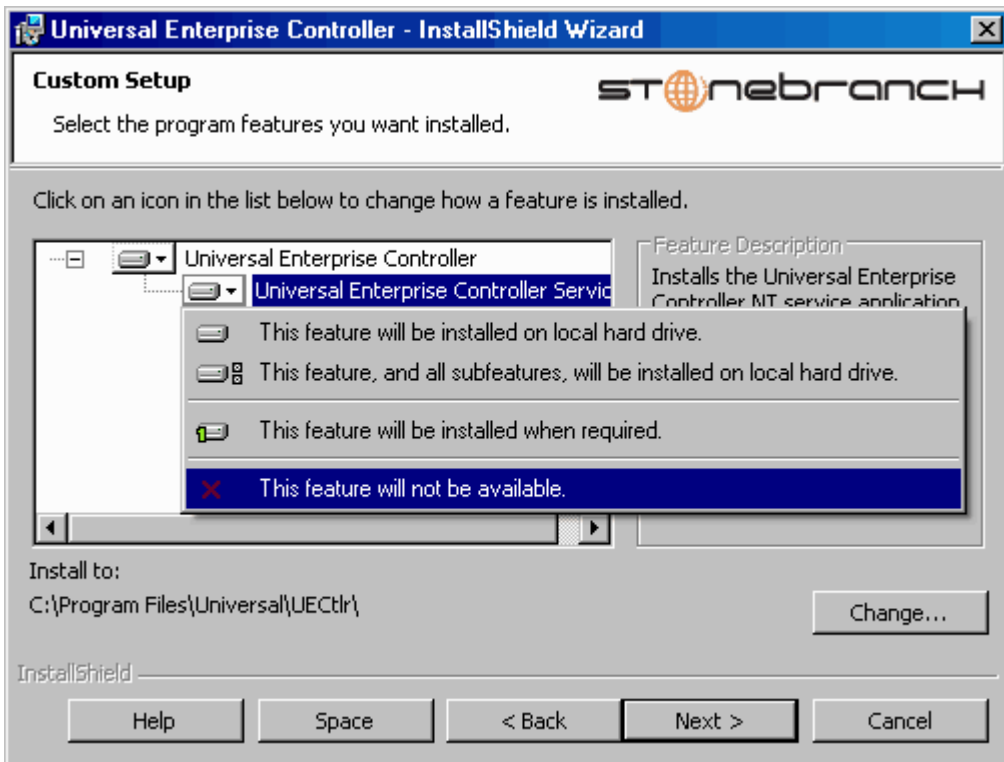
- Step 4** The previous figure shows that all Universal Enterprise Controller components will be installed in their respective directories under the **C:\Program Files\Universal** directory.

1. If you want to select a different location, click the **Change...** button.
2. 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.

**Step 5** If you do not want to install a component:

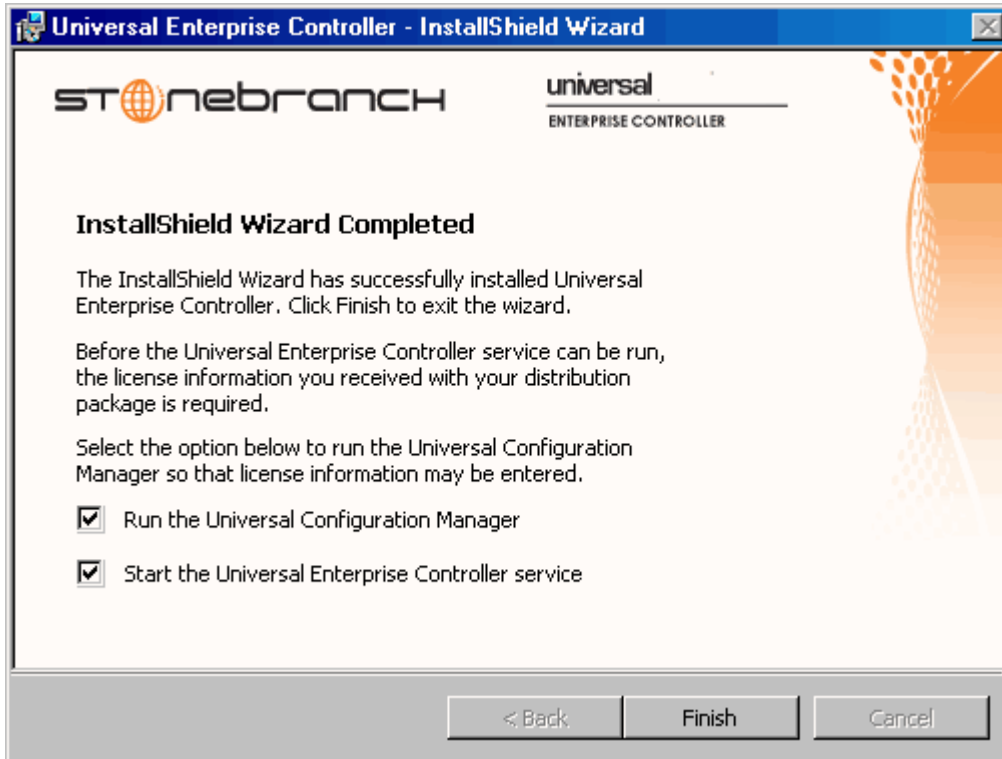
1. Click the drive icon next to that component name.
2. From the drop-down list that displays, select the X icon to mark the component as one not to be installed.

For example, the following figure indicates that the Universal Enterprise Controller Service has been selected to not be installed.



**Step 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.



**Step 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.

**Step 8** Click the **Finish** button to exit Windows Installation.



## Modifying a UEC Installation via the Graphical Interface

- Overview
- Adding or Removing Universal Enterprise Controller Components
- Repairing a Corrupted Universal Enterprise Controller Installation
- Removing a Universal Enterprise Controller Installation
  - Un-Installed Files

### Overview

After Universal Enterprise Controller is installed, Windows Installer can be run as many times as needed to modify the installation.

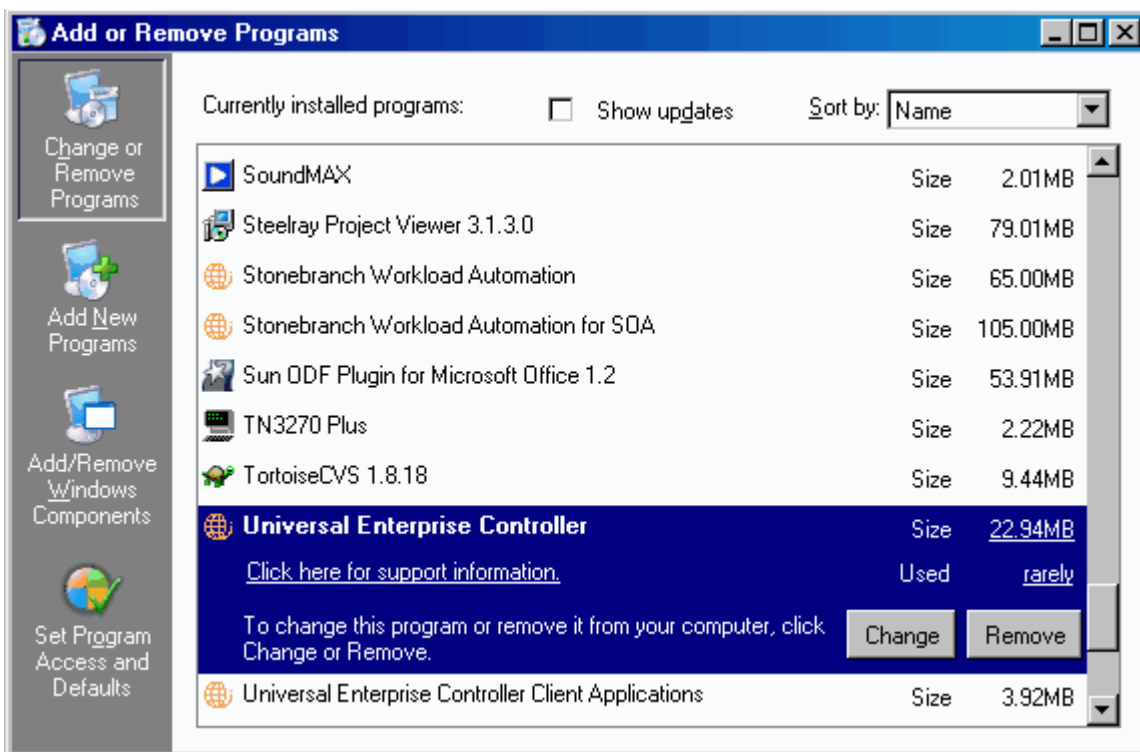
These installation modifications are:

- Adding or Removing Universal Enterprise Controller Components
- Repairing a Corrupted Universal Enterprise Controller Installation
- Removing a Universal Enterprise Controller Installation

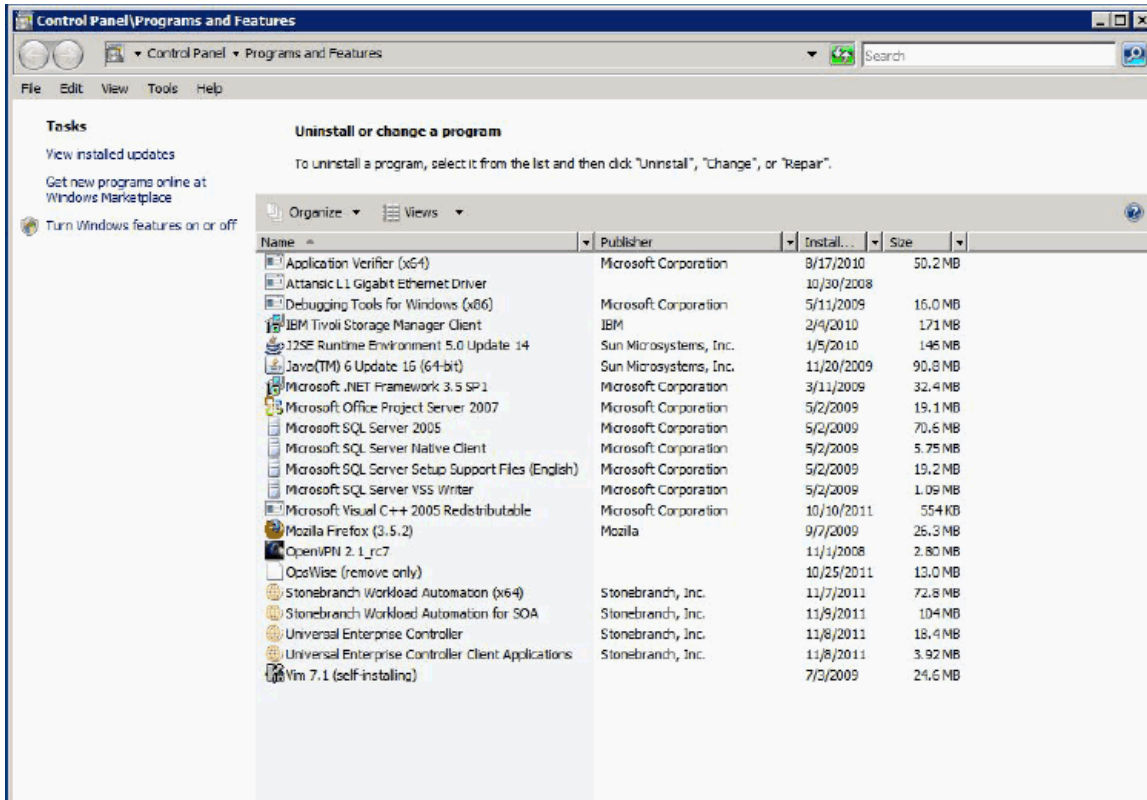
### Adding or Removing Universal Enterprise Controller Components

To add or remove components from a Universal Enterprise Controller installation, perform the following steps:

**Step 1** On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays.



**Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**  
The Programs and Features dialog, below, replaces the Add or Remove Programs dialog.



**Step 2** From the list of installed programs, select **Universal Enterprise Controller**.

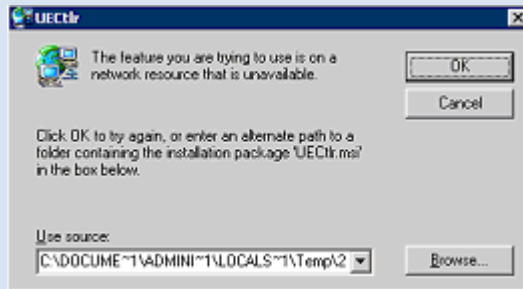
**Step 3** Click the **Change** button to start Windows Installer.

**Step 4** On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays.

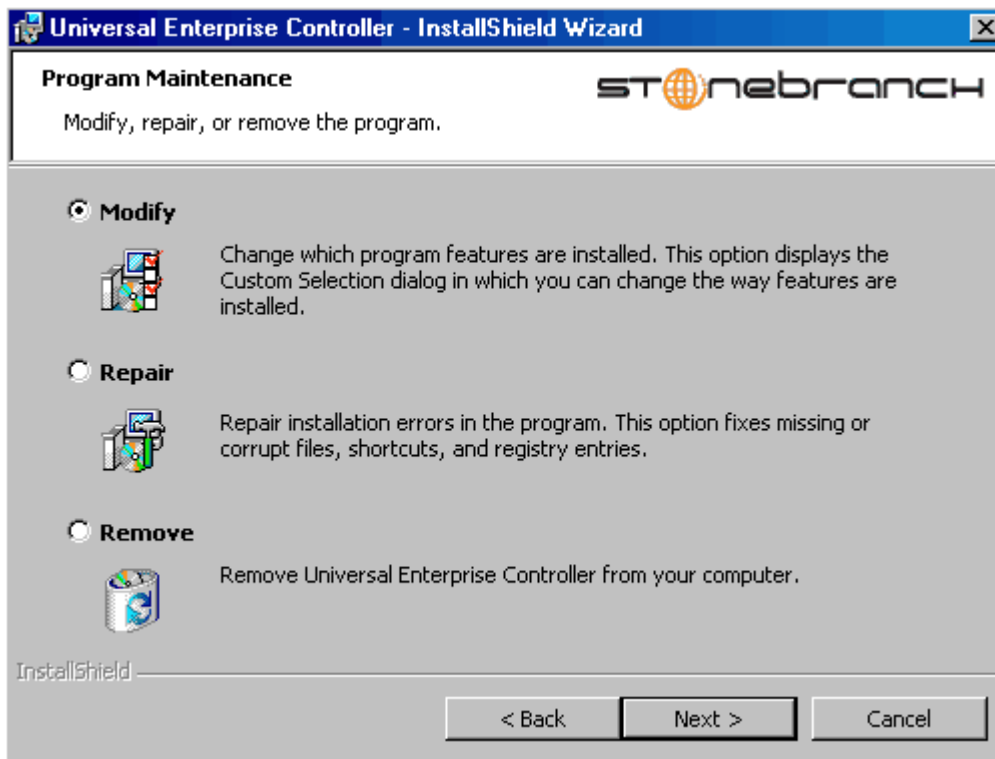
**i Installing over a Remote Desktop Session**

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.

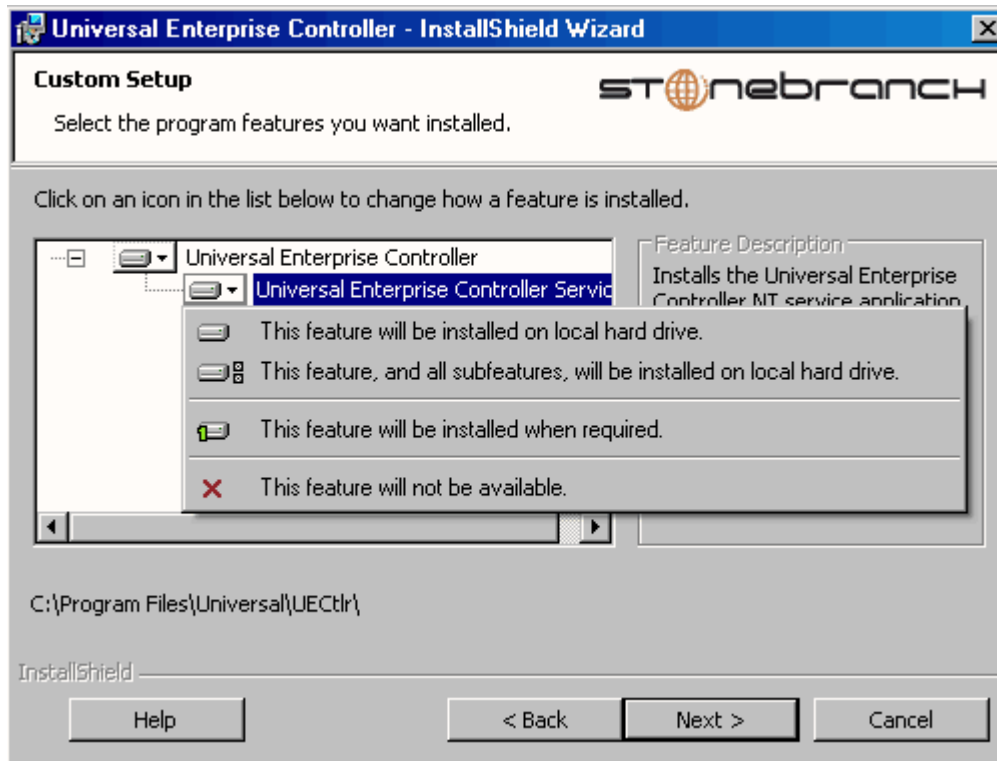
The distribution file's default extraction location is based on the **TEMP** environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed. Consequently, any attempts to modify the installation may fail because the Windows Installer can't locate the installation's source files (a dialog similar to the one shown below may be displayed).



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.



**Step 5** Click the **Modify** radio button, and then the **Next>** button, to display the Custom Setup dialog.



Currently installed components are identified by a drive icon.  
Uninstalled components are identified by an **X** icon.

**Step 6** To remove a currently installed component:

1. Click the drive icon next to that component.
2. Select the X icon from the drop-down list to mark the component for removal.

**Step 7** To add an uninstalled component:

1. Click the X icon next to that component.
2. Select the drive icon from the drop-down list to mark the component for installation.

**Step 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

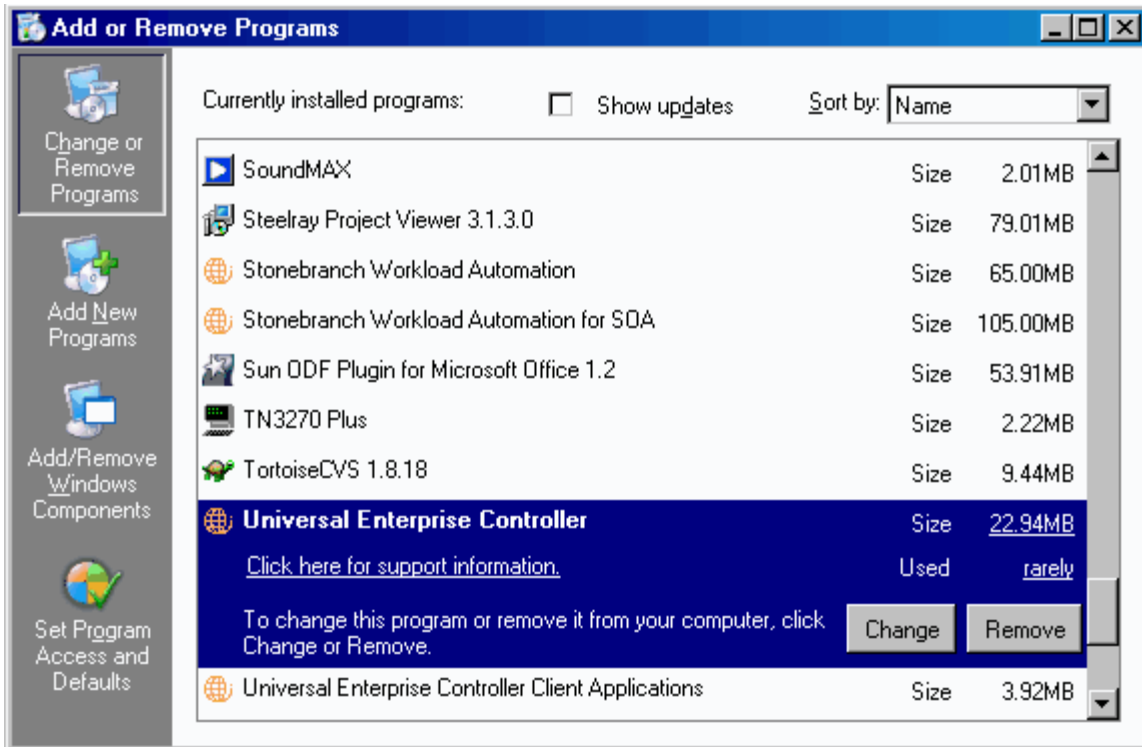
Windows Installer has the ability to recover accidentally deleted application files, configuration files, or registry entries required by Universal Enterprise Controller. This repair feature will re-install the missing items, making a complete re-install unnecessary.

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.

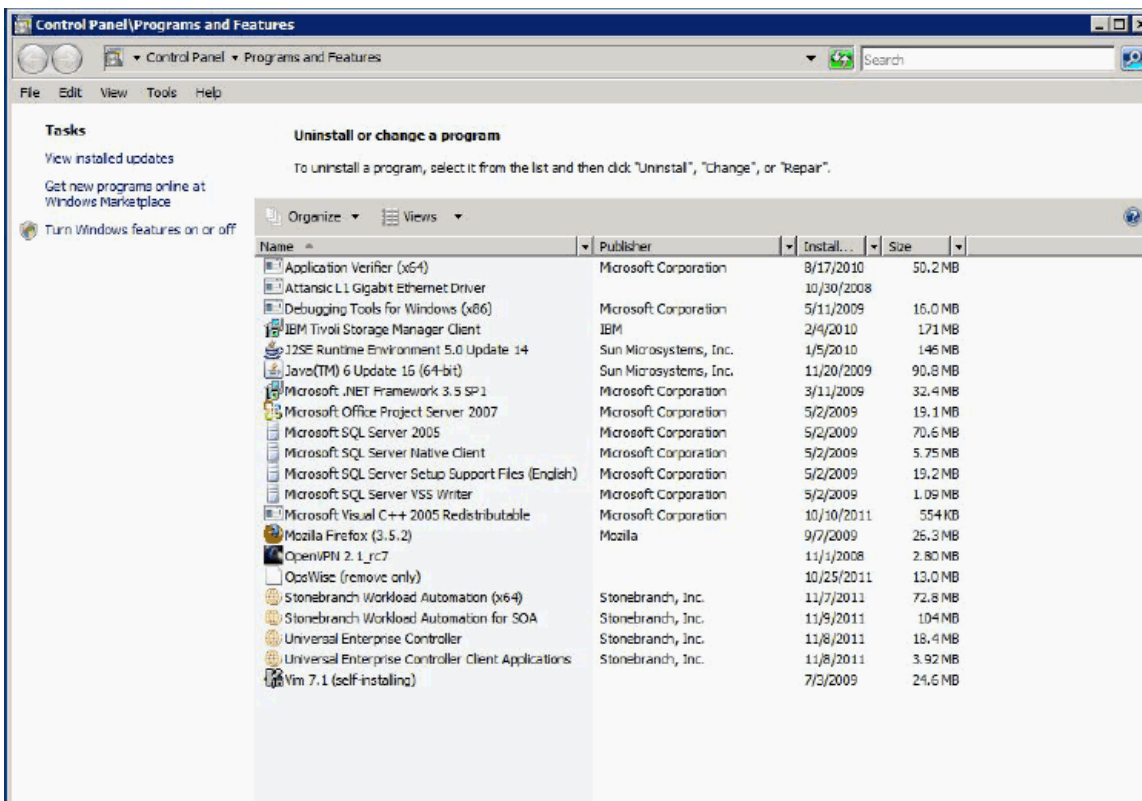
To repair an installation, perform the following steps:

**Step 1**

On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays.



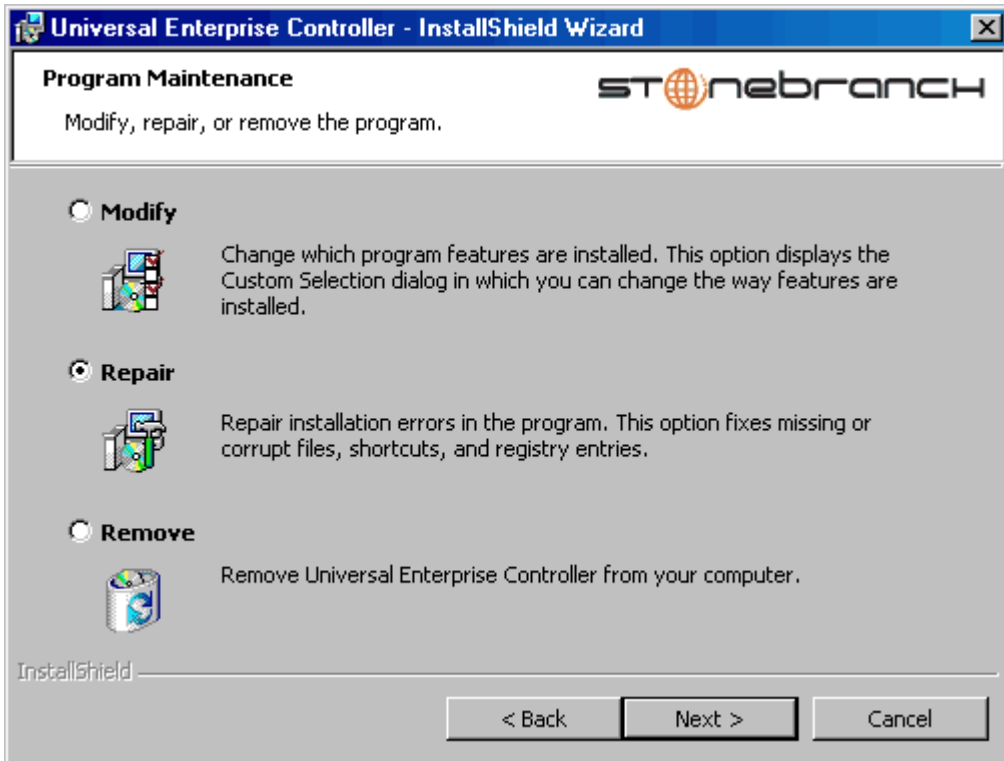
**Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**  
 The Programs and Features dialog replaces the Add or Remove Programs dialog.



**Step 2** From the list of installed programs, select **Universal Enterprise Controller**.

**Step 3** Click the **Change** button to start Windows Installer.

**Step 4** On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays.



**Step 5** Click the **Repair** radio button, then **Next>**, to display the Ready to Repair dialog.

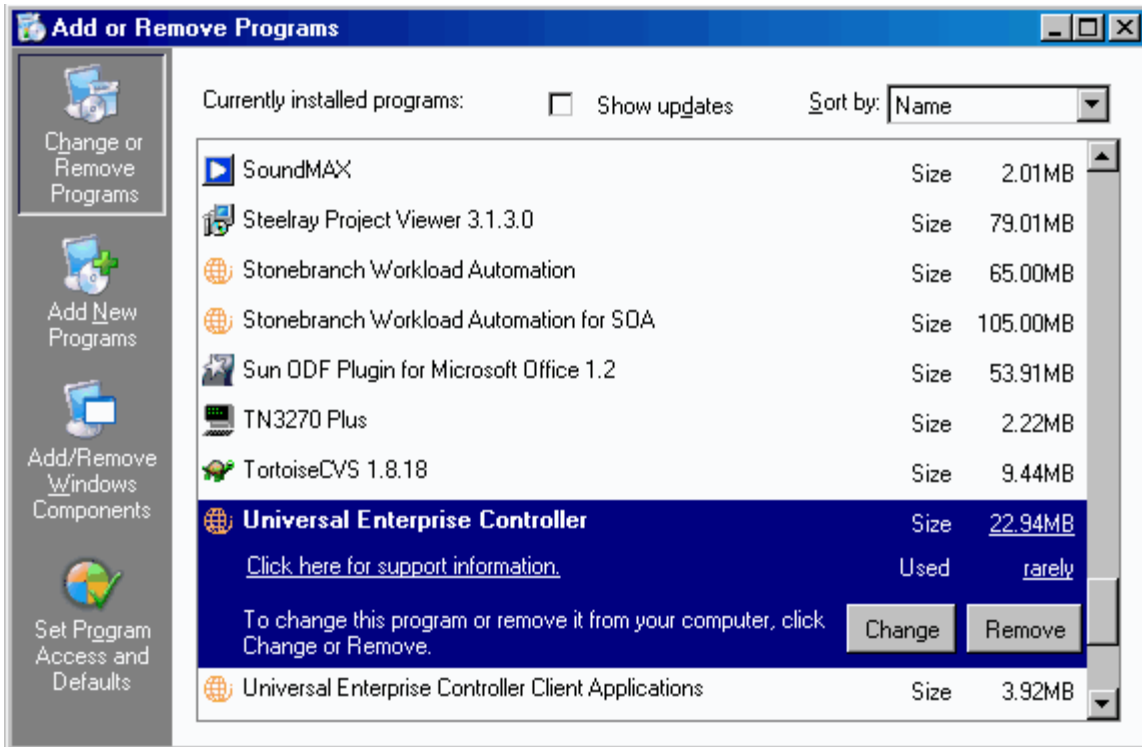
**Step 6** Follow the instructions displayed in successive dialogs to complete the repair.

### Removing a Universal Enterprise Controller Installation

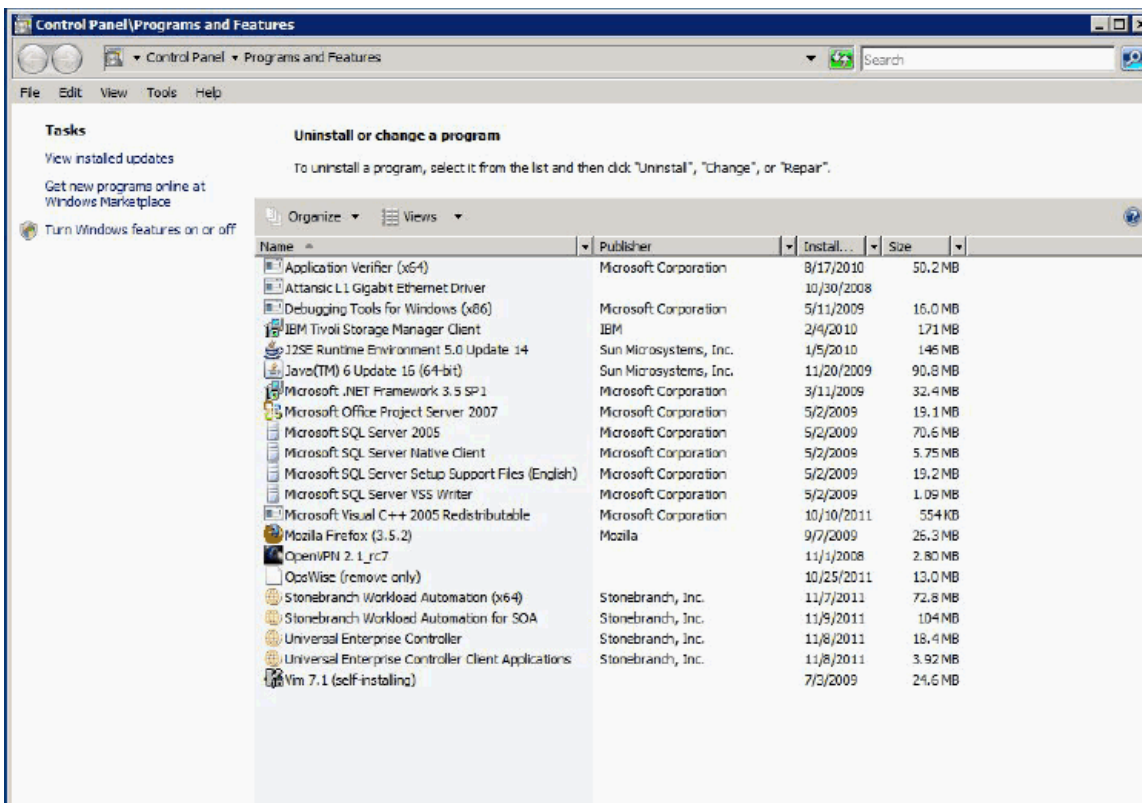
To uninstall a Universal Enterprise Controller installation, perform the following steps:

**Step 1**

On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays.



**Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**  
 The Programs and Features dialog replaces the Add or Remove Programs dialog.

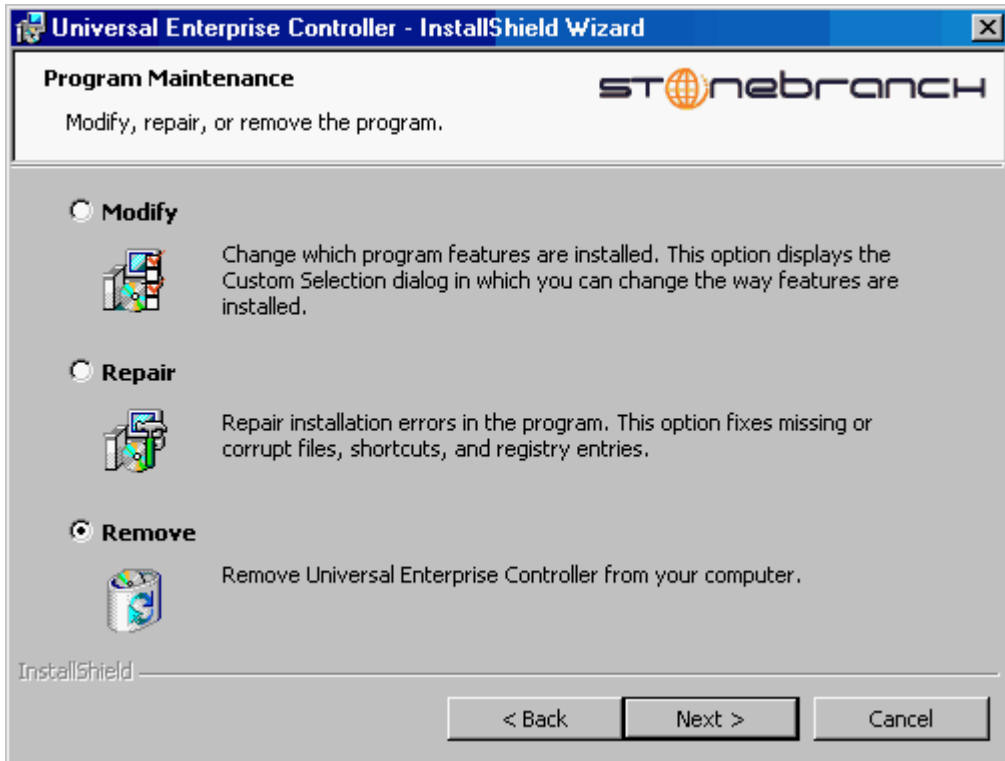


**Step 2** From the list of installed programs, select **Universal Enterprise Controller**.

**Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**  
To skip the remaining steps, click **Uninstall**.

**Step 3** Click the **Change** button to start Windows Installer.

**Step 4** On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays.



**Step 5** Select **Remove** and click the **Next>** button.

**Step 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 **.Universal** 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 **.Universal** directory, make sure that no other Stonebranch, Inc. products are installed there. (See [Universal Enterprise Controller for Windows - 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.



## Installing UEC via the Command Line Interface

- [Installing Universal Enterprise Controller via the Windows Installer Command Line Interface](#)
- [Windows Installer Command Line Syntax](#)
- [Windows Installer Parameters](#)
- [Windows Installer Installation Commands](#)
- [Detecting the Completion of Silent Installs](#)
- [Using the Distribution File for a Silent Install](#)

### Installing Universal Enterprise Controller via the Windows Installer Command Line Interface

This page describes how to install Universal Enterprise Controller using the Windows Installer command line interface.

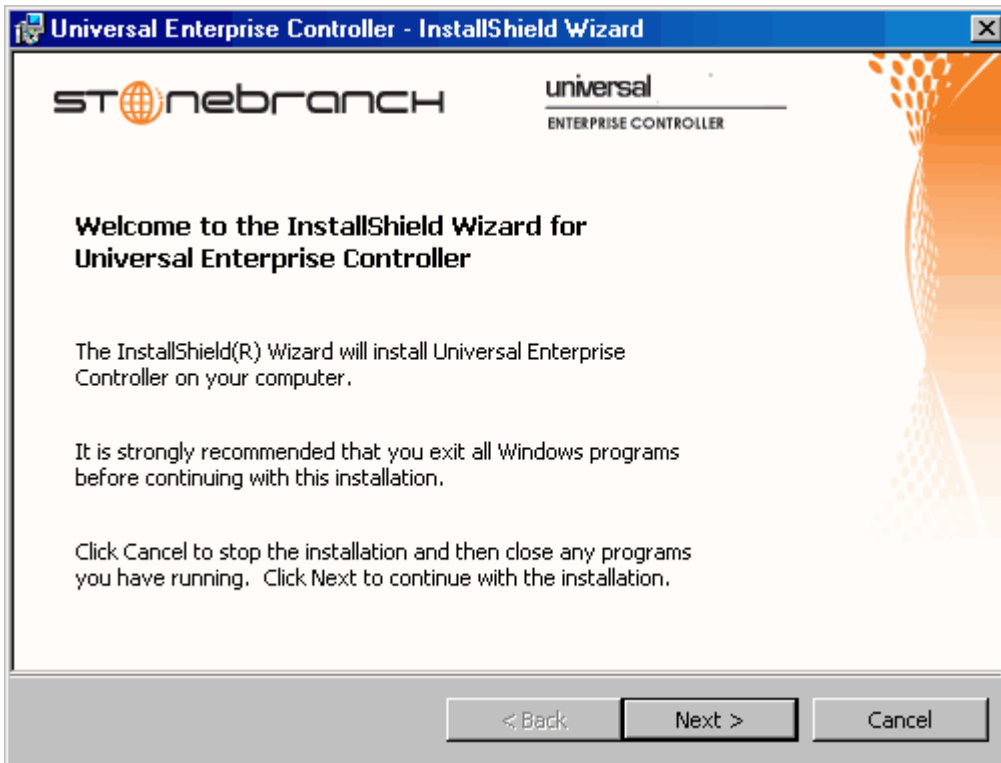
The command line interface is useful in situations where:

- Several Universal Enterprise Controller installations must be deployed.
- 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:

<b>Step 1</b>	Download the Universal Enterprise Controller for Windows product distribution file, <b>sb-UECtlr-5.1.0.&lt;level&gt;-windows-i386.exe</b> , to your work station.
<b>Step 2</b>	Execute the distribution file to extract the installation files.
<b>Step 3</b>	On the Location to Save Files dialog, select a location in which to store the installation files and click the <b>Next&gt;</b> button to extract the files.

**Step 4** On the Welcome dialog, 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](#)).

A better way to install Universal Enterprise Controller from the command line is to execute Windows Installer, **msiexec.exe** (see [Windows Installer Installation Commands](#), below).



**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](#), below).

### Windows Installer Command Line Syntax

The following figure illustrates the command line syntax when using Windows Installer to install Universal Enterprise Controller from the command line.

```
msiexec.exe [/I|/fom|/x] SetupPath\UECtrlr.msi
[INSTALLDIR=installdir]
[UECTLR={yes|no}]
[/q] [/L*v logfilepath]
```

The order in which the parameters are specified is significant. The **/I**, **/f** or **/x** command line switches must be first, followed by the name of the installation file, **UECtrlr.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** switches. These parameters are used only when the install switch (**/I**) is used.

The **/q** and **/L** switches can be specified in any order, but, if used, must come after all other parameters.

## Windows Installer Parameters

The following table describes the optional parameters that are available when using Windows Installer to install Universal Enterprise Controller from the command line.

Parameter	Description	Default
<code>/I</code>	Installs Universal Enterprise Controller. <b>/I</b> cannot be used with either the <b>/fom</b> (repair) or <b>/x</b> (remove) parameter.	n/a
<code>/fom</code>	Repairs a Universal Enterprise Controller installation. <b>om</b> (after the <b>f</b> ) are options used by the repair. There are other options available, but for behavior that matches the repair done from the graphical install, the <b>om</b> options must be used. <b>/fom</b> cannot be used with either the <b>/I</b> (install) or <b>/x</b> (remove) parameter.	n/a
<code>/x</code>	Removes Universal Enterprise Controller. <b>/x</b> cannot be used with either the <b>/I</b> (install) or <b>/fom</b> (repair) parameter.	n/a
<code>SetupPath \\UECtrlr.msi</code>	Path to the <b>UECtrlr.msi</b> file. If the product distribution file is used, <b>SetupPath</b> 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, <b>SetupPath</b> must point to the new location.	(none)
<code>INSTALLDIR</code>	Sets the root installation directory to <code>&lt;installdir&gt;</code> . Each component will be installed under this directory. <b>INSTALLDIR</b> is required if you want to install UEC under a directory different from the one specified by the <b>PROGRAMFILES</b> environment variable (typically <code>C:\Program Files\Universal</code> ). <b>INSTALLDIR</b> is valid only when the <b>/I</b> switch is used. Otherwise, it is ignored. If the directory contains spaces, you must use double ( " ) quotation marks around the path name.	(none)
<code>UECTLR</code>	Prevents the UEC service from being installed (if value is set to <b>no</b> ). This has the same effect as choosing not to install the Universal Enterprise Controller from the Custom Setup dialog (see <a href="#">Universal Enterprise Controller for Windows - Installation Procedures</a> ). It is valid only when the <b>/I</b> switch is used. Otherwise, it is ignored.	<b>yes</b>
<code>/q</code>	Instructs Windows Installer to run without a graphical interface (silent install). If <b>/q</b> is omitted from the command line, UEC installation is started from the command line, but run with a graphical interface. This is useful when an installation log file is desired. See <a href="#">Windows Installer Command Line Syntax</a> , <a href="#">Windows Installer Installation Commands</a> , and <a href="#">Detecting the Completion of Silent Installs</a> , below, for additional information regarding silent installs.	n/a
<code>/L*v</code>	Instructs Windows Installer to create an installation log file named <code>&lt;logfilepath&gt;</code> (full path name). If <code>&lt;logfilepath&gt;</code> contains spaces, you must enclose it with double ( " ) quotation marks. <b>*v</b> are flags used to specify the level of detail ( <b>verbose</b> ) contained in the log file. To reduce the amount of output generated, <b>*v</b> can be omitted. However, using these options is good practice; they can assist Stonebranch Customer Support with problem determination should any errors occur during installation.	n/a

## 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\UECtrlr.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\UECtrlr.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\UECtrlr.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\UECtrlr.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\UECtrlr.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:

```
sb-UECtrlr-5.1.0.0-windows-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.

## Modifying a UEC Installation via the Command Line Interface

- [Modifying a Universal Enterprise Controller Installation via the Windows Installer Command Line Interface](#)
- [Adding or Removing Universal Enterprise Controller Components](#)
- [Repairing a Corrupted Universal Enterprise Controller Installation](#)
- [Removing a Universal Enterprise Controller Installation](#)

### Modifying a Universal Enterprise Controller Installation via the Windows Installer Command Line Interface

This page 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 [Modifying a UEC Installation via the 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\UECtrlr.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\UECtrlr.msi /l 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\UECtrlr.msi /q
```

## Universal Enterprise Controller for Windows - 64-Bit Windows Editions

- Universal Enterprise Controller - Installing on 64-bit Windows Editions
- Applications Installed in the Windows System Folder
  - Example 1
  - Example 2

### Universal Enterprise Controller – Installing on 64-bit Windows Editions

All Workload Automation 5 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

This page 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.

### 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`).

The following table identifies the affected Workload Automation 5 applications.

File Name	Description
ucert.exe	Universal Certificate
ueclod.exe	UEC Load
uencrypt.exe	Universal Encrypt

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\ueclod
```

#### Example 2

To execute UECLoad within the 32-bit command shell, issue the following command:

```
%SystemRoot%\SysWOW64\cmd.exe /C %SystemRoot%\SysWOW64\ueclod
```



## Universal Enterprise Controller for Windows - Database Configuration

### 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:

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

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

<b>Step 1</b>	Inside the UEC database directory, create a text file named <b>DB_CONFIG</b> .
<b>Step 2</b>	Inside the <b>DB_CONFIG</b> file, add the following string: <b>set_tmp_dir *dbpath*</b> ( <b>dbpath</b> is the path to the location in which the database files reside.)
<b>Step 3</b>	Start / restart UEC.

## Universal Enterprise Controller for Windows - File Inventory Lists

- Universal Enterprise Controller - File Inventory Lists
- Universal Enterprise Controller
- Universal Products Install Merge Utility
- System Files

### 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 Controller 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 all 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>UECtrl\bin\acl.xml</code>	Used to store Access Control List entries for Universal Enterprise Controller.
<code>UECtrl\bin\brokers.xml</code>	Used to store a list of Universal Brokers that will be monitored by the Universal Enterprise Controller.
<code>UECtrl\bin\groups.xml</code>	Used to store defined groups of Universal Brokers.
<code>UECtrl\bin\librfc32.dll</code>	Support file for Universal Enterprise Controller.

<b>UECtlr\bin\ueccfg.dll</b>	Used by Universal Configuration Manager to manage Universal Enterprise Controller configuration options.
<b>UECtlr\bin\ueccfg.hlp</b>	Universal Enterprise Controller configuration help file.
<b>UECtlr\bin\uecdbrec.bat</b>	Recovers a Universal Enterprise Controller database, specified from the command line. The default database is <b>uec_evm.db</b> .
<b>UECtlr\bin\uecmngnt.dll</b>	Used to write error messages to the Windows Application event log.
<b>UECtlr\bin\uectls.exe</b>	Universal Enterprise Controller installation file.
<b>UECtlr\bin\users.xml</b>	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.
<b>UPIMerge\bin\upimerge.exe</b>	Command line interface to configuration file merge routines).
<b>USpool\bin\ludb_dump.exe</b>	UEC database utility; to be used only upon request of Stonebranch, Inc. Customer Support.
<b>USpool\bin\ludb_load.exe</b>	UEC database utility; to be used only upon request of Stonebranch, Inc. Customer Support.

The following file is installed under %ALLUSERSPROFILE%\ **Application Data**, which, by default, resolves to:

- C:\Documents and Settings\All Users\Application Data on 2003 and XP.
- C:\ProgramData on Vista and Windows 2008 Server.

<b>Universal\conf\ uec.conf</b>	Universal Enterprise Controller configuration file.
---------------------------------	---

The following files are installed under environment variable %systemRoot%, which, by default, resolves to:

- C:\Windows on all platforms.
  - On x86 platforms, these files are installed under %systemRoot%\system32.
  - On x64 platforms, these files are installed under %systemRoot%\SysWow64.

<b>ucert.exe</b>	Universal Certificate utility, used to generate X.509 Certificates.
<b>uecload.exe</b>	Universal Enterprise Controller load utility.
<b>uencrypt.exe</b>	Universal Encrypt utility, used to encrypt sensitive Universal application command line options.

## Universal Products Install Merge Utility

File	Description

**UPIMergebin\upimerge.exe**

Application program, always installed. Provides command line access to the same functionality used by the Workload Automation 5 installation to merge options from a new configuration file into an existing file.

## System Files

The following files will be installed only if they are newer than the existing files.

The directories shown in this table are relative to the %SYSTEMROOT% directory, where %SYSTEMROOT% is an environment variable that resolves to C:\Windows on all Windows platforms.

File	Description
<b>System32\asycfilt.dll</b>	Version 2.40.4275.1. This DLL is one of the components of the Microsoft OLE library.
<b>System32\comcat.dll</b>	Version 4.71.1460.1 of the Microsoft Component Category Manager library.
<b>Microsoft C-Runtime v8.0.50727.762 <sup>1</sup></b>	Version 8.0.50727.762 of the Microsoft C runtime side-by-side assembly.
<b>System32\msiexec.exe</b>	Version 3.1.4000.1823 of the Microsoft Windows Installer (see <a href="#">Windows Installer</a> for more information).
<b>System32\oleaut32.dll</b>	Version 2.40.4275.1. This DLL is one of the components of the Microsoft OLE library.
<b>System32\olepro32.dll</b>	Version 5.0.4275.1. This DLL is one of the components of the Microsoft OLE library.
<b>System32\psapi.dll</b>	Version 4.0.1371.1 of the Microsoft process status library
<b>System32\stdole2.tlb</b>	Version 2.40.4275.1. This file is one of the components of the Microsoft OLE library.

<sup>1</sup> 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.

## UEC Client Applications

Error formatting macro: redirect: java.lang.NullPointerException

## UEC Client Applications - Overview

### UEC Client Applications

The following information is provided for the installation of UEC Client Applications:

- [UEC Client Applications - Package](#)
- [UEC Client Applications - Installation Requirements](#)
- [UEC Client Applications - Installation Procedures](#)
- [UEC Client Applications - 64-Bit Windows Editions](#)
- [UEC Client Applications - File Inventory Lists](#)

## UEC Client Applications - Package

### Components

The Universal Enterprise Controller (UEC) 5.1.0 Client Applications for Windows package includes the desktop application versions of the following components:

- I-Administrator 5.1.0
- I-Activity Monitor 5.1.0
- I-Management Console 5.1.0

### Component Compatibility

The following table identifies the compatibility of Universal Enterprise Controller Client Applications 5.1.0 with previous component / product versions.

Component	Compatibility
Universal Enterprise Controller Client Applications 5.1.0	Not compatible with previous versions of Universal Enterprise Controller for Windows.

The component references pertain to all supported platforms for that version.

## UEC Client Applications - Installation Requirements

- [System Requirements](#)
- [Platform Requirements](#)
- [Java Runtime Environment](#)

### System Requirements

- One of the supported Windows operating systems. Currently, the following Windows operating systems are supported for UEC Client Applications:
  - Windows Server 2003 SP1 and higher (Itanium not supported)
  - Windows XP SP3
  - Windows Vista
  - Windows Server 2008 (Itanium not supported)
  - Windows 7
  - Windows Server 2008 R2 (Itanium not supported)
  - Windows Server 2012
- 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.
- Possible reboot: a reboot is required if the Windows Installer service is not installed, a version of the Windows Installer prior to 3.1.4000.1823 is installed, or if required files are in use at the time of the installation.
- TCP/IP.
- 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.
- Sun Java Runtime Environment (JRE 1.5 or above).

### Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opswise Automation Center 5.1.1](#) and [Indesca-Infitran 5.1.0](#) page to make sure that your platform is supported before performing an installation.

### Java Runtime Environment

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](http://www.java.com).



#### **A Stonebranch 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.



## **UEC Client Applications - Installation Procedures**

Error formatting macro: redirect: java.lang.NullPointerException

## **UEC Client Applications - Installation Procedures Overview**

### **UEC Client Applications - Installation Procedures**

The following procedures are provided for the installation and modification of UEC Client Applications:

- [Installing UEC Client Applications via the Graphical Interface](#)
- [Modifying a UEC Client Applications Installation via the Graphical Interface](#)
- [Installing UEC Client Applications via the Command Line Interface](#)
- [Modifying a UEC Client Applications Installation via the Command Line Interface](#)

## Installing UEC Client Applications via the Graphical Interface

### 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:

<b>Step 1</b>	Download the UEC Client Applications for Windows product distribution file, <code>sb-UEClient-5.1.0.&lt;level&gt;-windows-i386.exe</code> , to your work station.
---------------	---

**Step 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 [Windows Installer](#)), you can simply double-click the extracted Client Applications installation file, `UEClient.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.

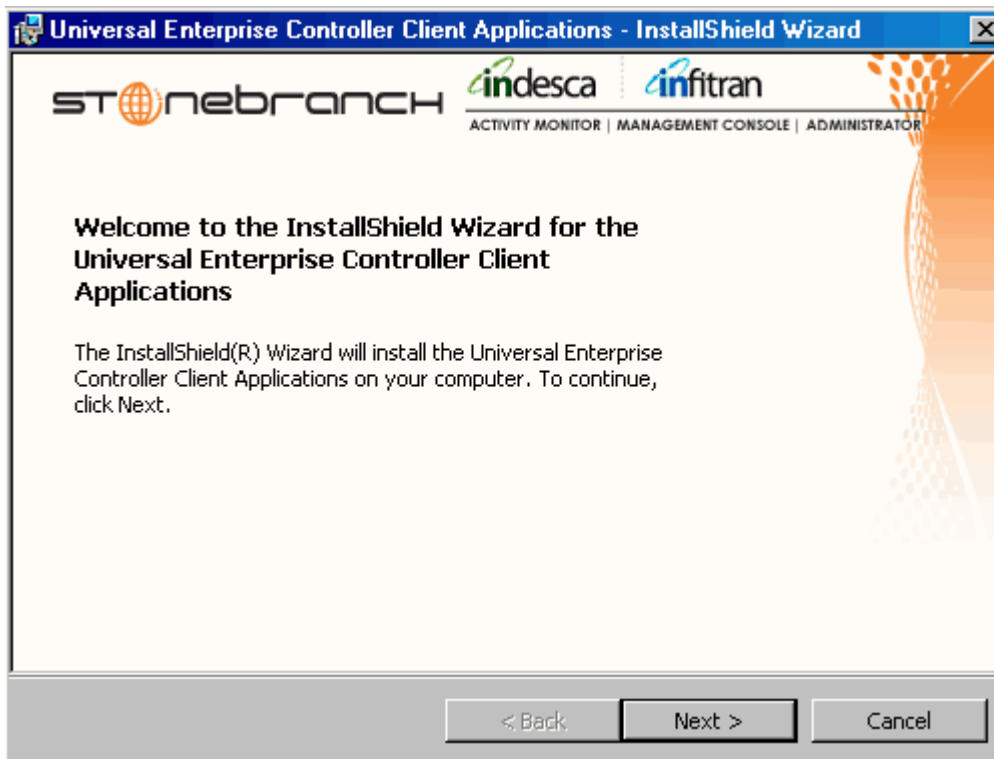
For example, the distribution file's default extraction location is based on the TEMP environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed.

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.

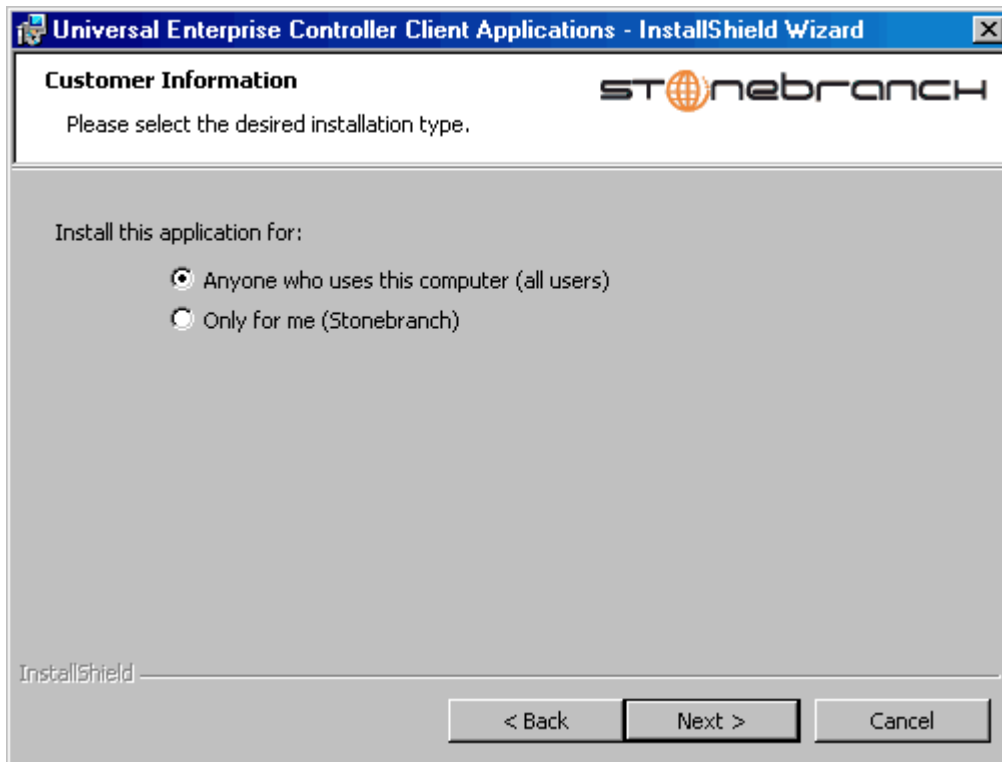
The installation automatically will begin after the files are extracted.

It first will verify that your machine meets the minimum system requirements (see [UEC Client Applications - Installation Requirements](#)). If the requirements are met, a Welcome dialog displays.



**Step 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, below, displays).
- 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 Step 5).



**Step 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 [Modifying a UEC Client Applications Installation via the 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 [Modifying a UEC Client Applications Installation via the 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.

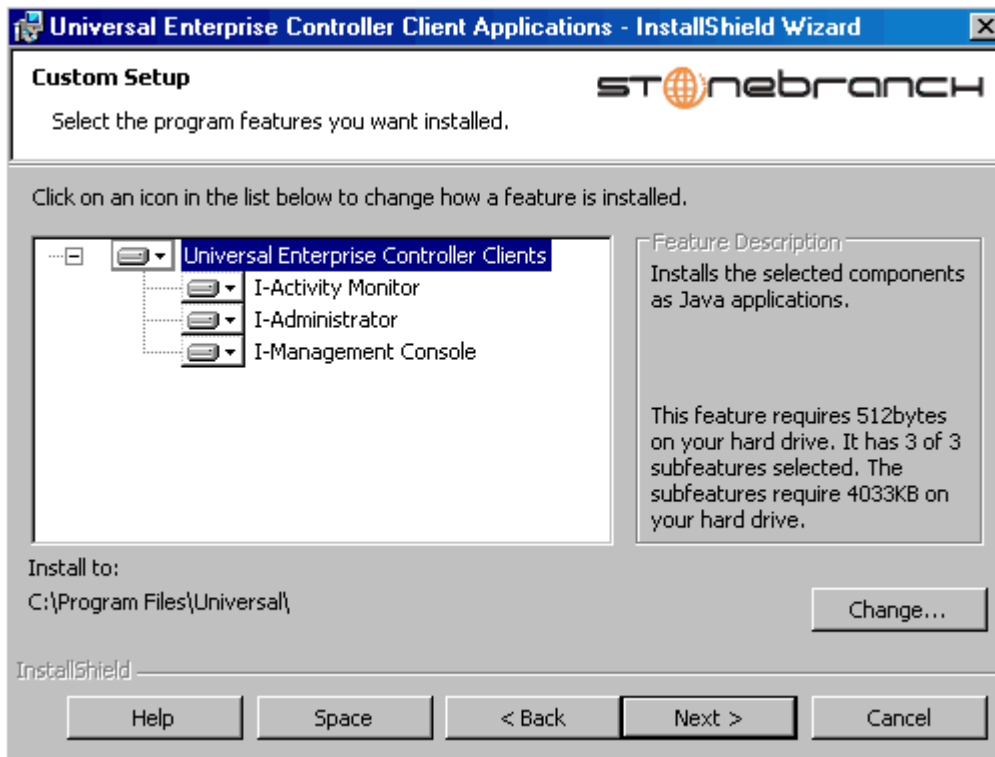
**Note**

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.

**A Stonebranch Tip****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.

- Step 5** Click the **Next>** button. A list of UEC Client Applications components included in the installation package then displays. It is from this list that you can select which components to install.



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.

 **A Stonebranch Tip**  
The directory identified in the figure above 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.

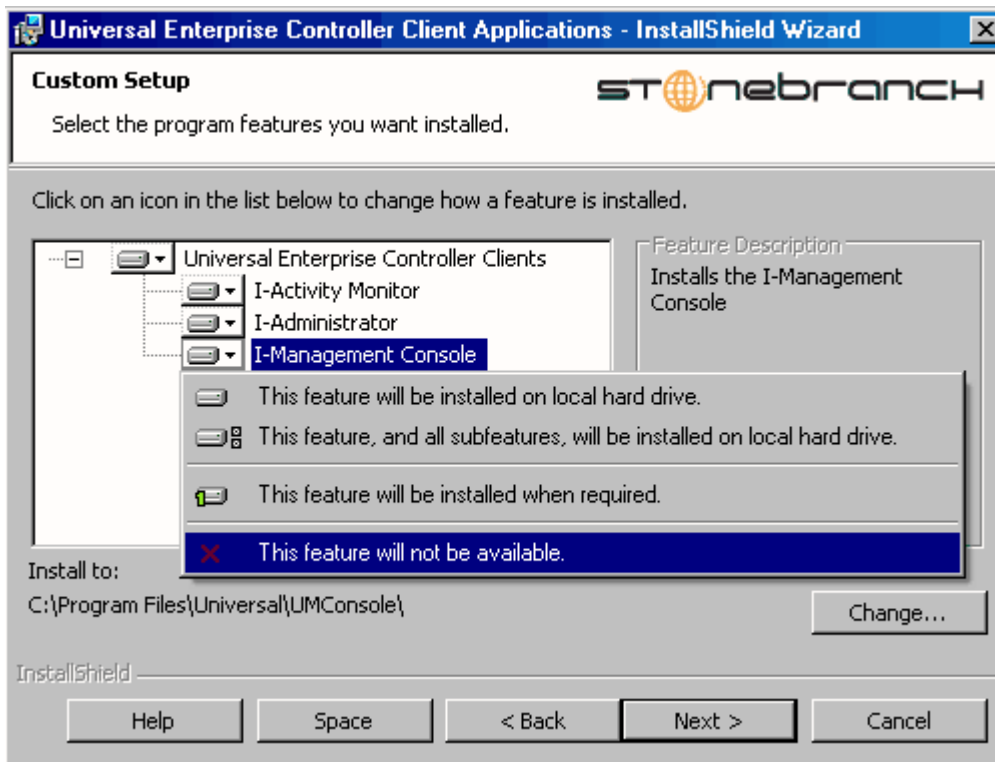
- Step 6** The figure above shows that all UEC Client Applications will be installed in their respective directories under the `C:\Program Files\Universal` directory.

1. If you want to select a different location, click the **Change...** button.
2. 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.

**Step 7** If you do not want to install a component:

1. Click the drive icon next to that component name.
2. From the drop-down list that displays, select the X icon to mark the component as one not to be installed.

For example, the figure below indicates that I-Management Console has been selected to not be installed.





- Step 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.

**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 [UEC Client Applications - Installation Requirements](#)).

- Step 9** Click the **Finish** button to exit Windows Installation.

## Modifying a UEC Client Applications Installation via the Graphical Interface

- [Modifying a UEC Client Applications Installation via the Windows Installer Graphical Interface](#)
- [Adding or Removing UEC Client Applications Components](#)
- [Repairing a Corrupted UEC Client Applications Installation](#)
- [Removing a UEC Client Applications Installation](#)
  - [Un-Installed Files](#)

### 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.

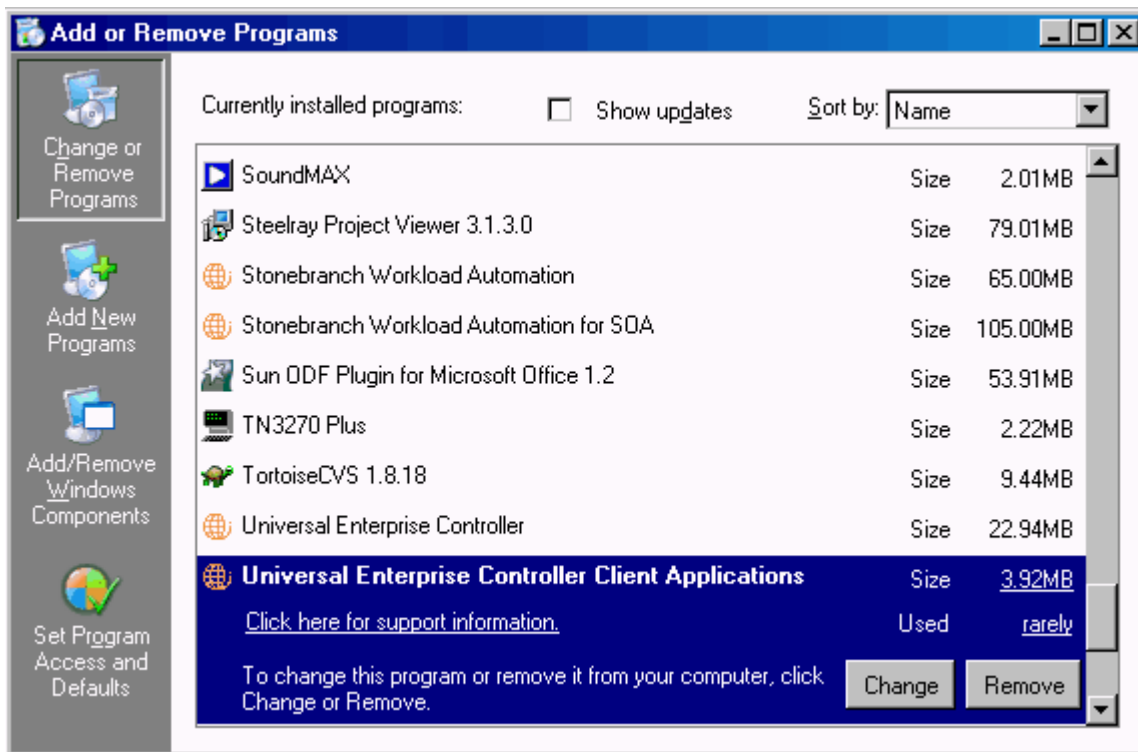
These installation modifications are:

- 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

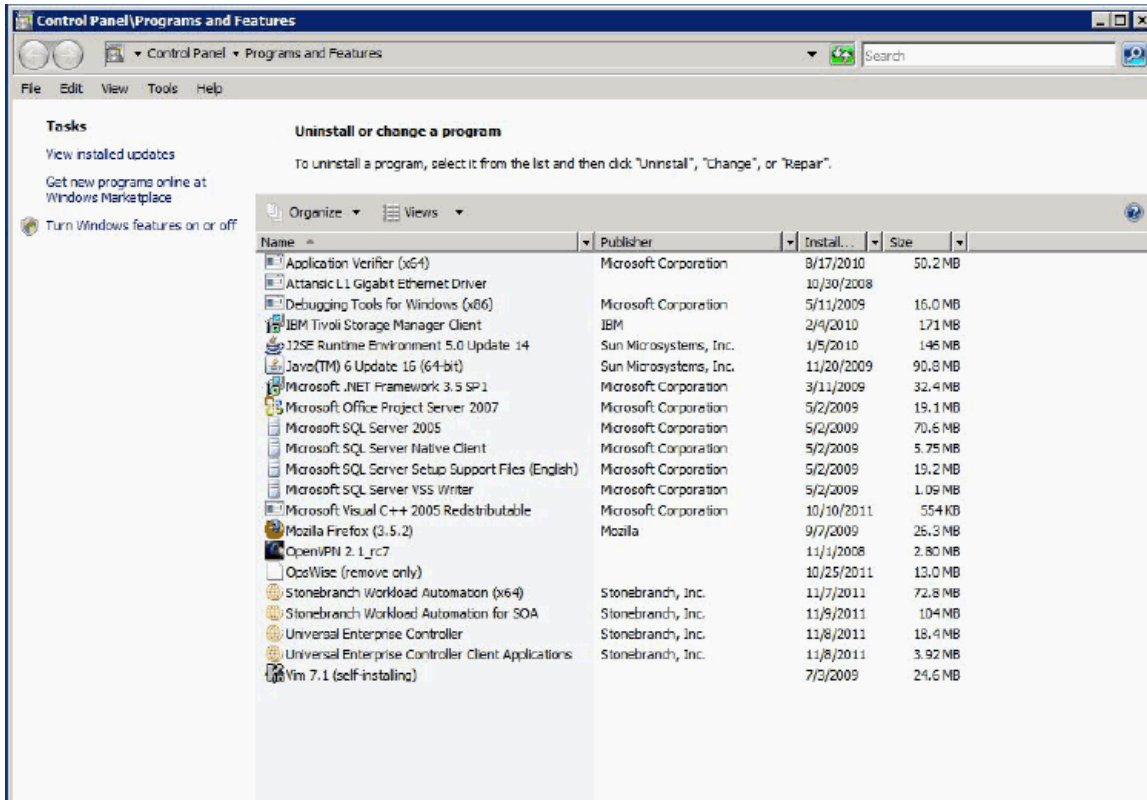
To add or remove components from a UEC Client Applications installation, perform the following steps:

**Step 1** On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays.



#### Windows Vista, Windows 7, Windows Server 2008 / 2008 R2

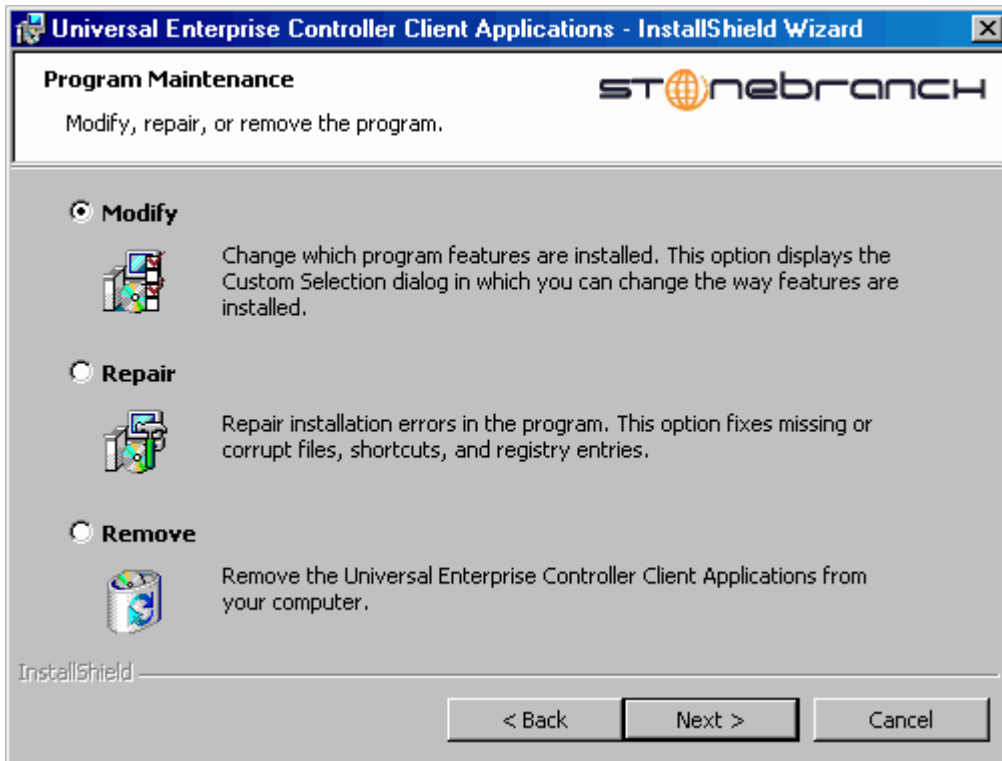
The Programs and Features dialog replaces the Add or Remove Programs dialog.



**Step 2** From the list of installed programs, select **Universal Enterprise Controller Client Applications**.

**Step 3** Click the **Change** button to start Windows Installer.

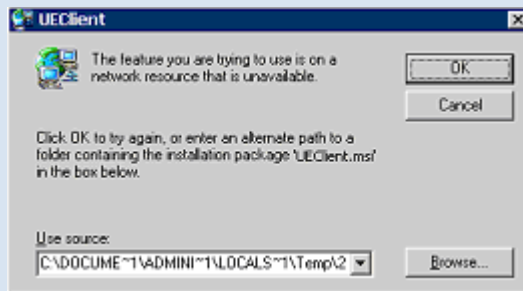
**Step 4** On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays.



#### **Windows Server 2003**

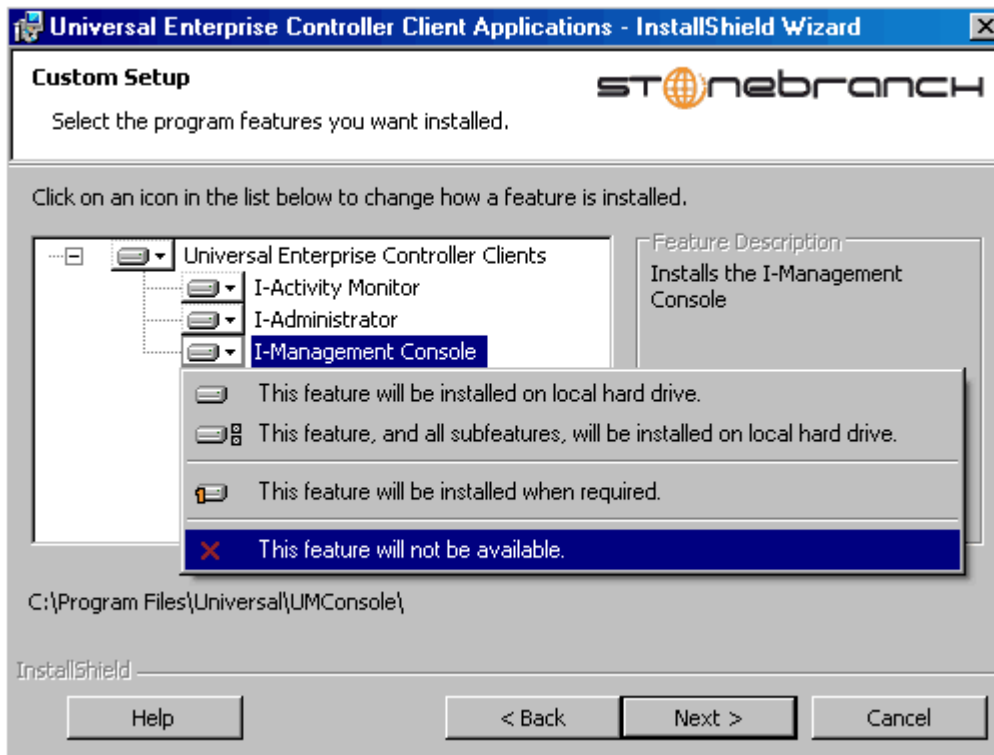
If the UEC Client Applications were installed via a Remote Desktop connection, the environment settings used during that session may no longer be available. Windows Server 2003 allows multiple Remote Desktop sessions for multiple users, and each session has its own environment. Depending on the way the 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 below may be displayed).



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.

**Step 5** Click the **Modify** radio button, and then the **Next>** button, to display the Custom Setup dialog.



Currently installed components are identified by a drive icon.

Uninstalled components are identified by an **X** icon.

**Step 6** To remove a currently installed component:

1. Click the drive icon next to that component.
2. Select the X icon from the drop-down list to mark the component for removal.

**Step 7** To add an uninstalled component:

1. Click the X icon next to that component.
2. Select the drive icon from the drop-down list to mark the component for installation.

**Step 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.

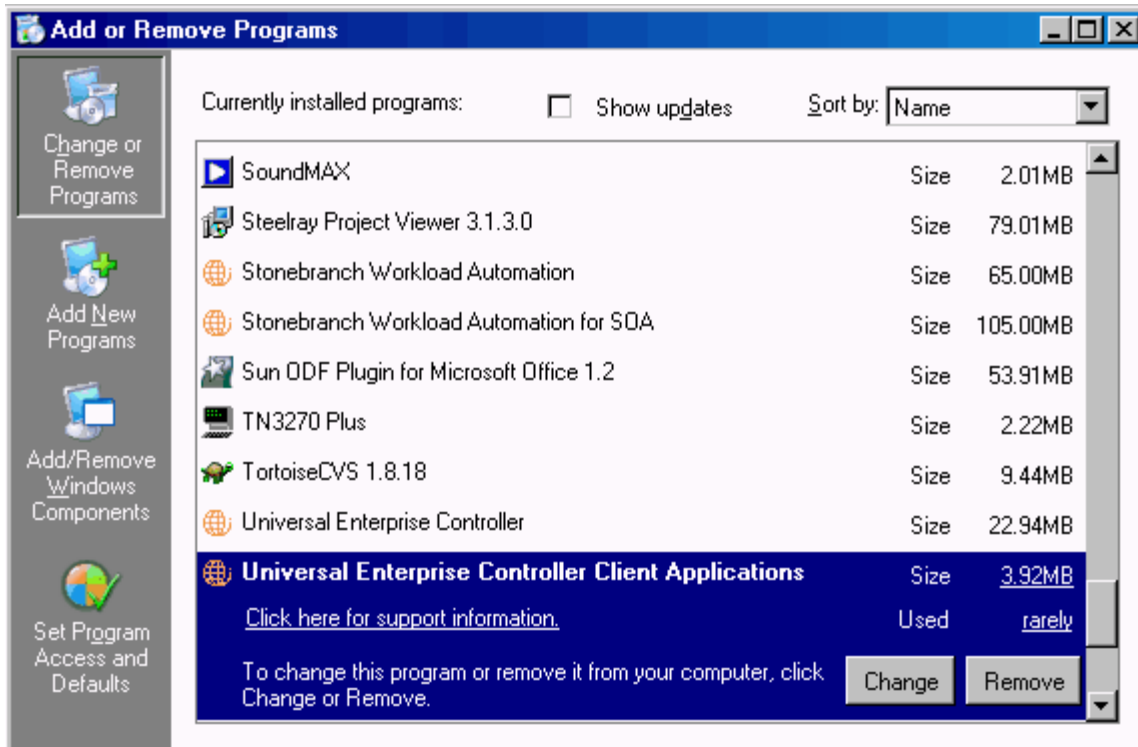
## Repairing a Corrupted UEC Client Applications Installation

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:

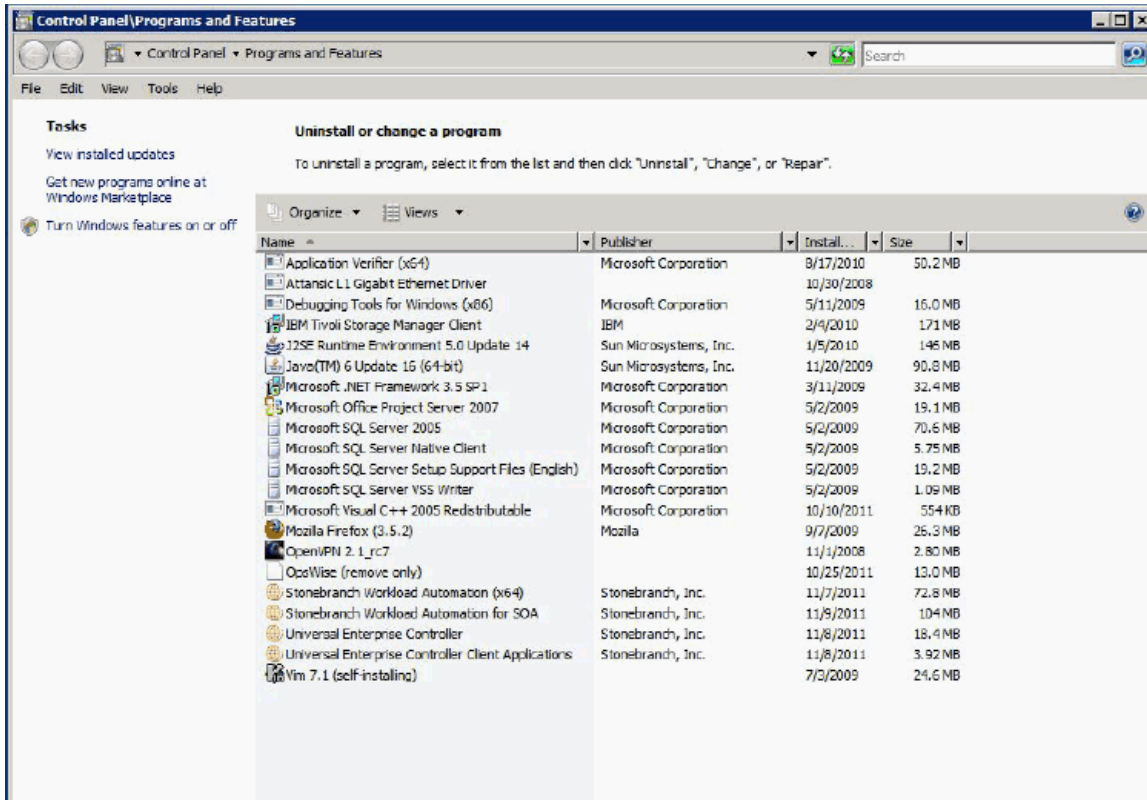
**Step 1**

On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays.



**Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**

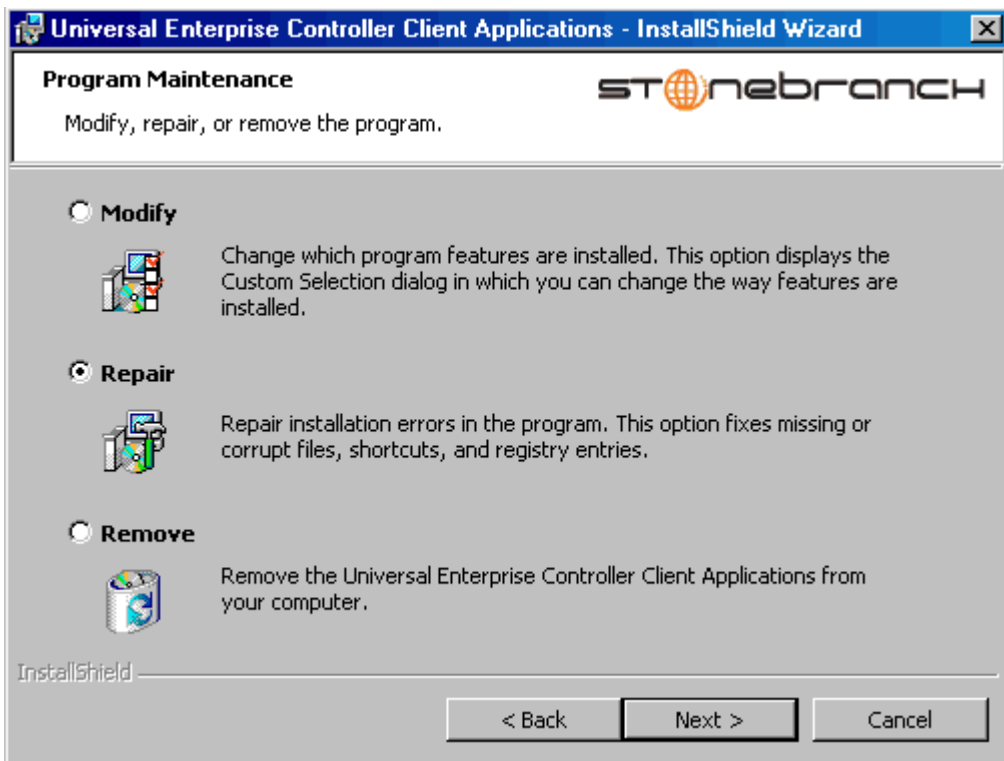
The Programs and Features dialog replaces the Add or Remove Programs dialog.



**Step 2** From the list of installed programs, select **Universal Enterprise Controller Client Applications**.

**Step 3** Click the **Change** button to start Windows Installer.

**Step 4** On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays.



**Step 5** Click the **Repair** radio button, and then the **Next>** button, to display the Ready to Repair dialog.

**Step 6** Follow the instructions displayed in successive dialogs to complete the repair.

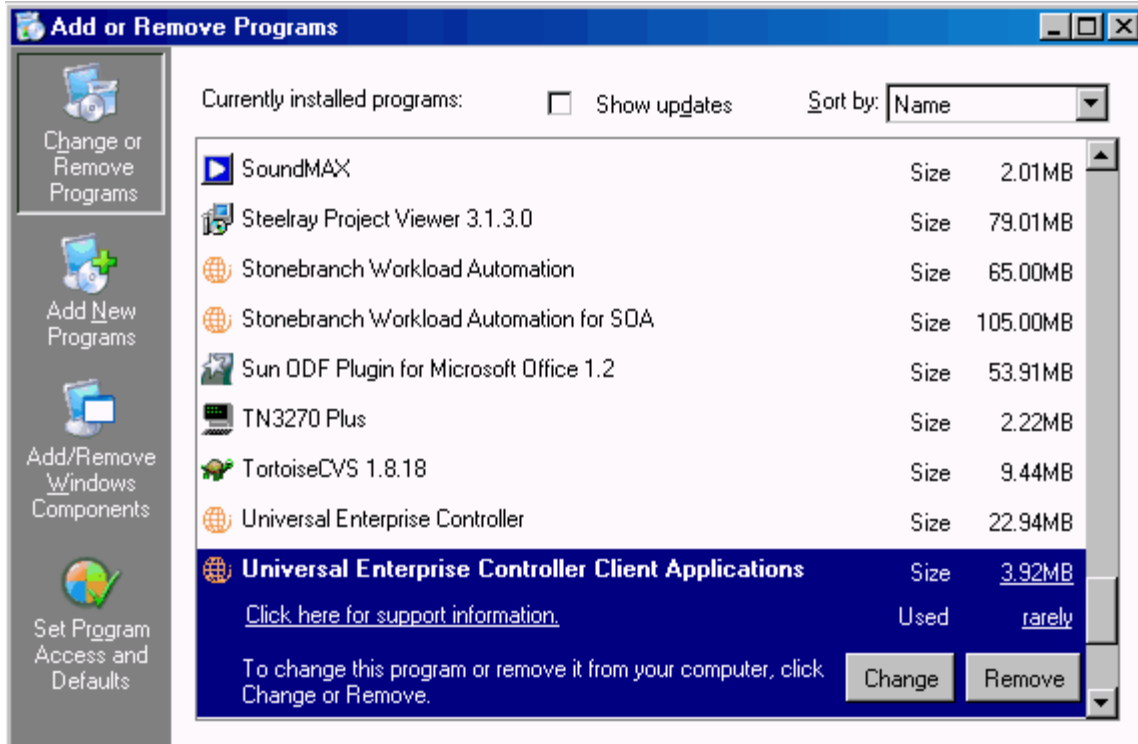
### Removing a UEC Client Applications Installation

To uninstall a UEC Client Applications installation, perform the following steps:

**Step 1**

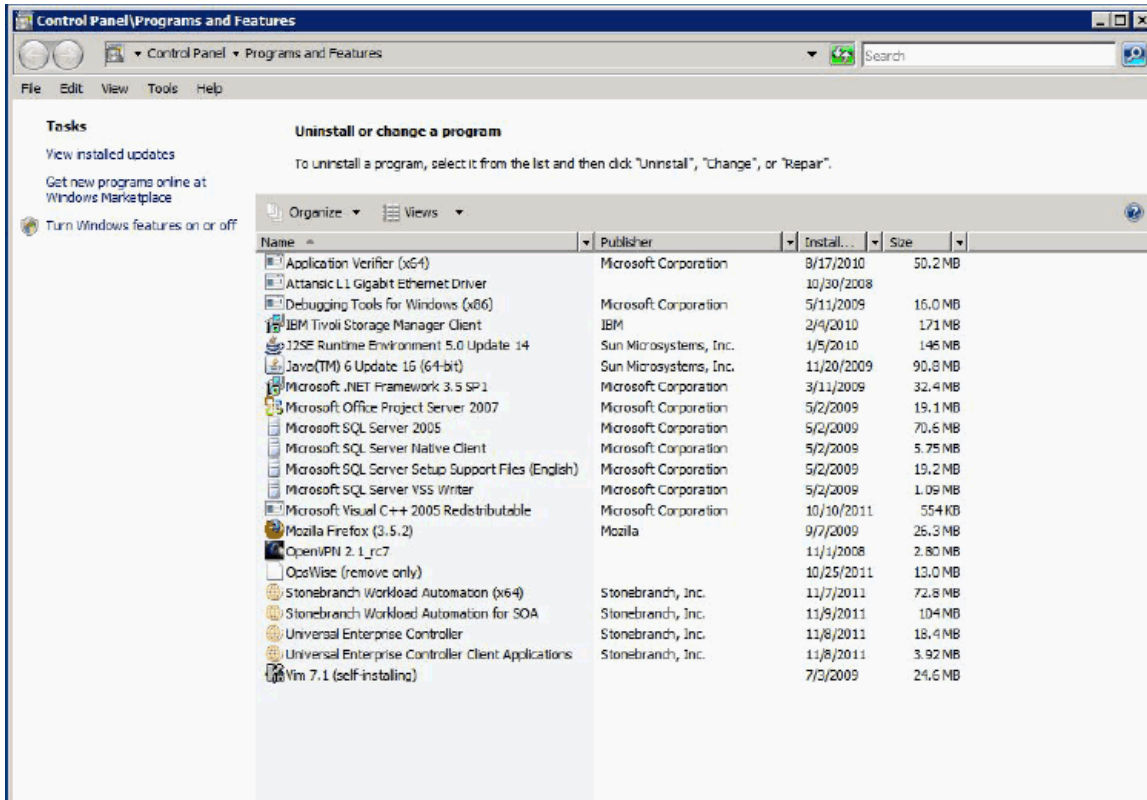


On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays.




**Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**

The Programs and Features dialog replaces the Add or Remove Programs dialog.

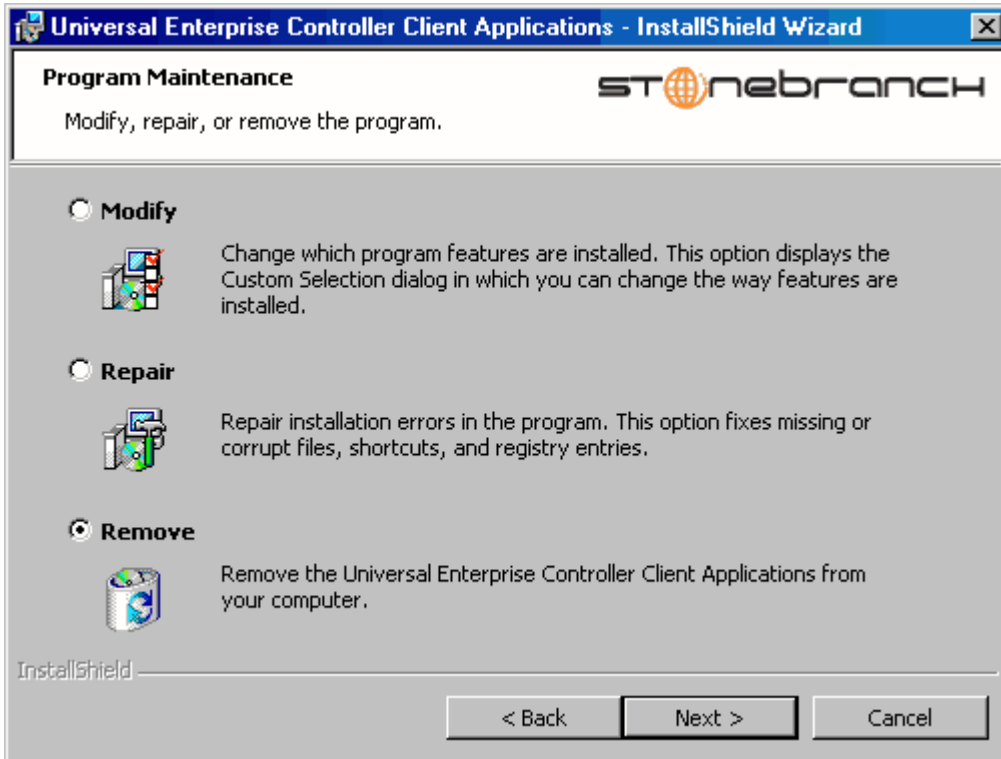


**Step 2** From the list of installed programs, select **Universal Enterprise Controller Client Applications**.

 **Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**  
To skip the remaining steps, click **Uninstall**.

**Step 3** Click the **Change** button to start the installation program.

**Step 4** On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays.



**Step 5** Select **Remove** and click the **Next>** button.

**Step 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 **.Universal** 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 **.Universal** directory, make sure that no other Stonebranch, Inc. products are installed there. (See [UEC Client Applications - File Inventory Lists](#) for a list of files and directories installed with UEC Client Applications.)

## Installing UEC Client Applications via the Command Line Interface

- Installing UEC Client Applications via the Windows Installer Command Line Interface
- Windows Installer Command Line Syntax
- Windows Installer Command Line Parameters
- Windows Installer Installation Commands
- Detecting the Completion of Silent Installs
- Using the Distribution File for a Silent Install

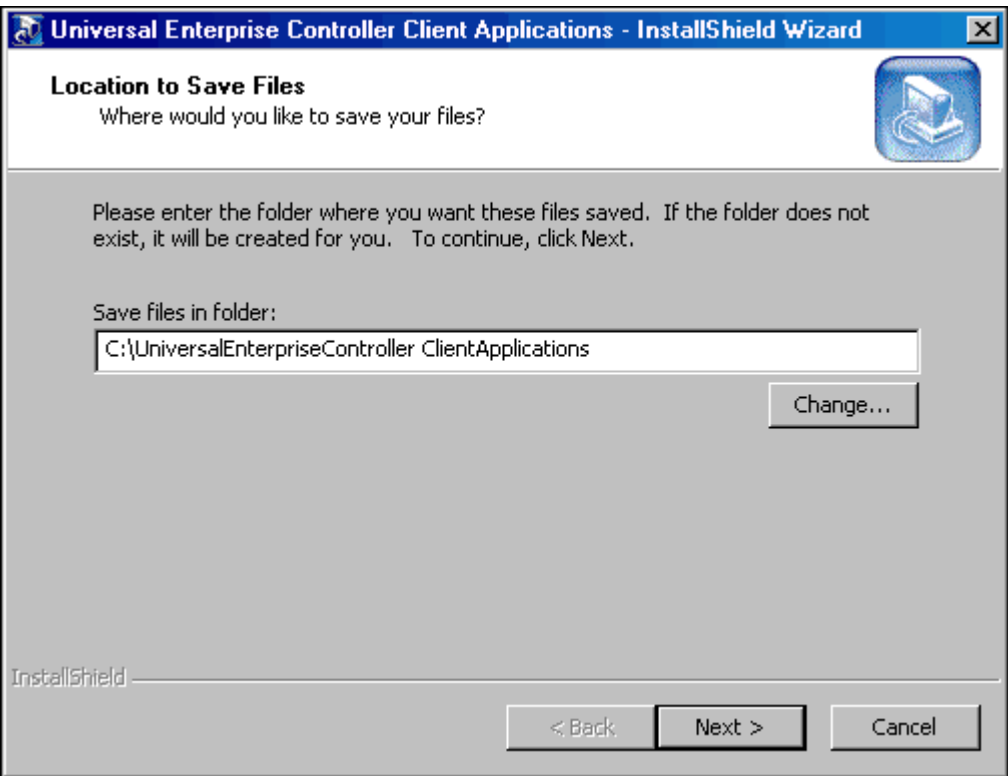
### Installing UEC Client Applications via the Windows Installer Command Line Interface

This page describes how to install UEC Client Applications using the Windows Installer command line interface.

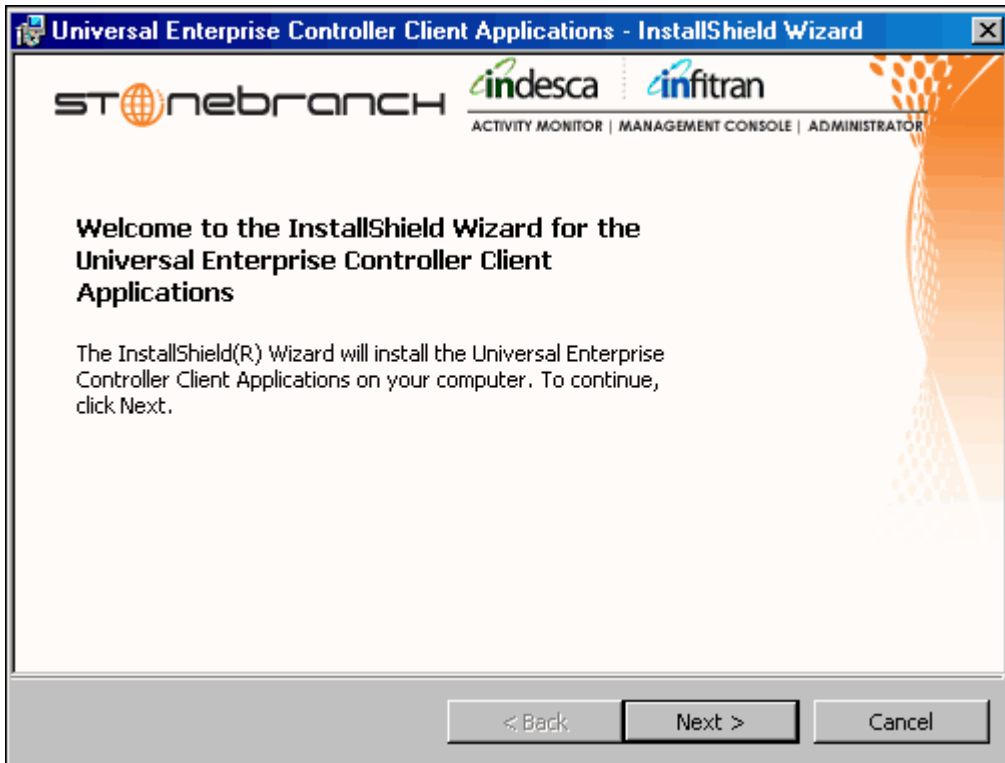
The command line interface is useful in situations where:

- Several UEC Client Applications installations must be deployed across many different systems.
- 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:

<b>Step 1</b>	Download the UEC Client Applications for Windows product distribution file, <code>sb-UEClient-5.1.0.&lt;level&gt;-windows-i386.exe</code> , to your work station.
<b>Step 2</b>	Execute the distribution file to extract the installation files.
<b>Step 3</b>	<p>On the Location to Save Files dialog, select a location in which to store the installation files and click the <b>Next&gt;</b> button to extract the files.</p> 

**Step 4** On the Welcome dialog, 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 the UEC Client Applications can continue (see [Determining if Windows Installer will be Installed or Upgraded](#)).

A better way to install UEC Client Applications from the command line is to execute Windows Installer, **msiexec.exe** (see [Windows Installer Installation Commands](#), below).



**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](#), below).

## Windows Installer Command Line Syntax

The following figure 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]
```

The order in which the parameters are specified is significant. The **/I**, **/f** or **/x** command line switches 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** switches. These parameters are used only when the install switch (**/I**) is used.

The **/q** and **/L** switches can be specified in any order, but, if used, must come after all other parameters.

## Windows Installer Command Line Parameters

The following table describes the optional parameters that are available when using Windows Installer to install UEC Client Applications from the command line.

Parameter	Description	Default
<code>/I</code>	Installs the UEC Client Applications.  <b>/I</b> cannot be used with either the <b>/fom</b> (repair) or <b>/x</b> (remove) parameter.	n/a
<code>/fom</code>	Repairs a UEC Client Applications installation.  <b>om</b> (after the <b>/f</b> ) are options used by the repair. There are other options available, but for behavior that matches the repair done from the graphical install, the <b>om</b> options must be used.  <b>/fom</b> cannot be used with either the <b>/I</b> (install) or <b>/x</b> (remove) parameter.	n/a
<code>/x</code>	Removes the UEC Client Applications.  <b>/x</b> cannot be used with either the <b>/I</b> (install) or <b>/fom</b> (repair) parameter.	n/a
<code>SetupPath \UEClient.msi</code>	Path to the <b>UEClient.msi</b> file.  If the product distribution file is used, <b>SetupPath</b> 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, <b>SetupPath</b> must point to the new location.	(none)
<code>INSTALLTO</code>	Sets the root installation directory to <code>&lt;installdir&gt;</code> . Each component will be installed under this directory.  <b>INSTALLTO</b> is required only if you want to install the UEC Client Applications in a directory that is different from the default, which varies depending on the type of installation being done (see <b>PERUSER</b> parameter, below).  <b>INSTALLTO</b> is valid only when the <b>/I</b> switch is used. Otherwise, it is ignored. If the directory contains spaces, you must use double ( " ) quotation marks around the path name.	<b>per-machine installation:</b> Directory specified by the <b>PROGRAMFILES</b> environment variable (typically <b>C:\Program Files\Universal</b> ).  <b>per-user installation:</b> Location under the directory specified by the <b>USERPROFILE</b> environment variable. For example, if the <b>USERPROFILE</b> directory is set to <b>C:\Documents and Settings\username</b> , the default target directory will be <b>C:\Documents and Settings\username\Application Data\Universal</b> . If the UEC Client Applications already are installed, the installation directory will default to its existing location.

<p style="text-align: center; border: 1px solid black; padding: 5px;">PERUSER</p>	<p>Performs an installation for all users or a specific user account:</p> <p>Values are <b>0</b> and <b>1</b>:</p> <ul style="list-style-type: none"> <li>• <b>0</b> specifies a per-machine installation; 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.</li> <li>• <b>1</b> specifies a per-user installation; 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.</li> </ul> <p><b>PERUSER</b> is required only under the following situations:</p> <ul style="list-style-type: none"> <li>• For per-user installs, when the installation is being done with a Windows account that is a member of the Administrators group.</li> <li>• For uninstalls, where a Windows account that is a member of the Administrators group is removing a per-user installation.</li> </ul>	<p><b>0</b>, if UEC Client Applications is installed using an Administrative account.</p> <p><b>1</b>, if a regular user account (that is, a non-Administrative user) is executing the installation.</p>
<p style="text-align: center; border: 1px solid black; padding: 5px;">UECADMIN</p>	<p>Prevents I-Administrator from being installed (if value is <b>no</b>). This has the same effect as choosing not to install the application version of I-Administrator from the Custom Setup dialog (see <a href="#">UEC Client Applications - Installation Procedures</a>).</p> <p><b>UECADMIN</b> is valid only when the <b>/I</b> switch is used. Otherwise, it is ignored.</p>	<p><b>yes</b></p>
<p style="text-align: center; border: 1px solid black; padding: 5px;">UAMONITOR</p>	<p>Prevents I-Activity Monitor from being installed (if value is <b>no</b>). This has the same effect as choosing not to install the application version of I-Activity Monitor utility from the Custom Setup dialog (see <a href="#">UEC Client Applications - Installation Procedures</a>).</p> <p><b>UAMONITOR</b> is valid only when the <b>/I</b> switch is used. Otherwise, it is ignored.</p>	<p><b>yes</b></p>
<p style="text-align: center; border: 1px solid black; padding: 5px;">UMGMTCON</p>	<p>Prevents I-Management Console from being installed (if value is <b>no</b>). This has the same effect as choosing not to install the application version of I-Management Console utility from the Custom Setup dialog (see <a href="#">UEC Client Applications - Installation Procedures</a>).</p> <p><b>UMGMTCON</b> is valid only when the <b>/I</b> switch is used. Otherwise, it is ignored.</p>	<p><b>yes</b></p>

/q	<p>Instructs Windows Installer to run without a graphical interface (silent install).</p> <p>If <b>/q</b> is omitted from the command line, the UEC Client Applications installation is started from the command line, but run with a graphical interface. This is useful when an installation log file is desired.</p> <p>See <a href="#">Windows Installer Command Line Syntax</a>, <a href="#">Windows Installer Installation Commands</a>, and <a href="#">Detecting the Completion of Silent Installs</a>, below, for additional information regarding silent installs.</p>	n/a
/L*v	<p>Instructs Windows Installer to create an installation log file named <b>&lt;logfilepath&gt;</b> (full path name). If <b>&lt;logfilepath&gt;</b> contains spaces, you must enclose it with double ( " ) quotation marks around the path name.</p> <p><b>*v</b> are flags used to specify the level of detail (<b>verbose</b>) contained in the log file. To reduce the amount of output generated, <b>*v</b> can be omitted. However, using these options is good practice; they can assist Stonebranch Customer Support with problem determination should any errors occur during installation.</p>	n/a

## 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 msixec.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, **msixec.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:

```
sb-UEClient-5.1.0.0-windows-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**  
**msixec.exe** switch for quiet install with no graphic display of progress.

## Modifying a UEC Client Applications Installation via the Command Line Interface

- [Modifying a UEC Client Applications Installation via the Windows Installer Command Line Interface](#)
- [Adding or Removing UEC Client Applications Components](#)
- [Repairing a Corrupted UEC Client Applications Installation](#)
- [Removing UEC Client Applications from the Command Line](#)

### Modifying a UEC Client Applications Installation via the Windows Installer Command Line Interface

This page describes how to modify a UEC Client Applications installation via the Windows Installer command line interface.

After UEC Clients are installed, Windows Installer can be run as many times as needed to modify the installation by:

- 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

Currently, it is only possible to add or remove UEC Client Applications components using the Windows Installer graphical interface. (see [Adding or Removing Components](#) in [Modifying a UEC Client Applications Installation via the 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\UEClient.msi /q
```

To remove a *per-user* installation that was installed using an Administrator account, issue the following command:

```
msiexec.exe /x SetupPath\UEClient.msi PERUSER=1 /q
```



**A Stonebranch 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.

## **UEC Client Applications - 64-Bit Windows Editions**

### **UEC Client Applications – Installing on 64-bit Windows Editions**

All Workload Automation 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 the UEC Client Applications should not require any modification when installing on 64-bit Windows editions.

## UEC Client Applications - File Inventory Lists

- UEC Client Applications - File Inventory List
  - System 32 Path
- I-Administrator
- I-Activity Monitor
- I-Management Console
- System Files

### 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 page 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.

#### System 32 Path

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 all 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
UECAdmin\uecadmin.jar	I-Administrator Java archive (JAR) file.
UECAdmin\lib\ueccommon.jar	Common routines shared between I-Administrator and I-Management Console.
UECAdmin\lib\uecumc.jar	I-Management Console resources used by I-Administrator.

#### I-Activity Monitor

File	Description
UAMonitor\uam.jar	I-Activity Monitor Java archive (JAR) file.

#### I-Management Console

File	Description
UMConsole\uecumc.jar	I-Management Console Java archive (JAR) file.

**UMConsole\lib\ueccommon.jar**

Common routines shared between I-Management Console and I-Administrator.

## System Files

The following files will be installed only if they are newer than the existing file.

File	Description
<b>System32\msiexec.exe</b>	Version 3.1.4000.1823 of the Microsoft Windows Installer (see <a href="#">Windows Installer</a> for more information).

## **Workload Automation 5 for SOA for Windows**

Error formatting macro: redirect: java.lang.NullPointerException

## **Workload Automation 5 for SOA for Windows - Overview**

### **Workload Automation 5 for SOA for Windows**

The following information is provided for the installation of Workload Automation 5 for SOA for Windows:

- [Workload Automation 5 for SOA for Windows - Package](#)
- [Workload Automation 5 for SOA for Windows - Installation Requirements](#)
- [Workload Automation 5 for SOA for Windows - Pre-Installation - Upgrade Backups](#)
- [Workload Automation 5 for SOA for Windows - Installation Procedures](#)
- [Workload Automation 5 for SOA for Windows - Configuring and Starting UAC Server](#)
- [Workload Automation 5 for SOA for Windows - 64-Bit Windows Editions](#)
- [Workload Automation 5 for SOA for Windows - File Inventory Lists](#)

(For licensing information, see [Windows Installation - Licensing](#).)



## Workload Automation 5 for SOA for Windows - Package

### Components

The Workload Automation 5 for SOA 5.1.0 for Windows package includes the following components:

- Workload Automation 5 for SOA
  - Universal Application Container Server
  - Universal Application Container
  - Universal Application Interface

### Component Compatibility

The following table identifies the compatibility of Workload Automation 5 for SOA for Windows 5.1.0 with previous component / product versions.

Component	Compatibility
Workload Automation 5 for SOA 5.1.0	Universal Command Manager 4.3.0, 4.2.0, 4.1.0, and 3.2.0.

The component references pertain to all supported platforms for that version.

## Workload Automation 5 for SOA for Windows - Installation Requirements

### System Requirements

- One of the supported Windows operating systems. Currently, the following Windows operating systems are supported for Workload Automation 5 for SOA for Windows:
  - Windows Server 2003 SP1 and higher
  - Windows XP SP3
  - Windows Vista
  - Windows Server 2008
  - Windows 7
  - Windows Server 2008 R2
  - Windows Server 2012
- 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.
- Workload Automation 5.1.0.0 or later (32-bit packages only).

### Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opswise Automation Center 5.1.1](#) and [Indesca-Infitran 5.1.0](#) page to make sure that your platform is supported before performing an installation.




## Workload Automation 5 for SOA for Windows - Pre-Installation - Upgrade Backups

### Workload Automation 5 for SOA for Windows – Pre-Installation / Upgrade Backups

Before upgrading to the latest release of Workload Automation 5 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 table identifies the files – and their locations – that should be backed up or copied before you install a new release or upgrade a current release.

File	Location
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  <div style="background-color: #ffffcc; padding: 5px;">  <b>Note</b> The names of the jar files differ depending on which JMS Provider you are using. </div>
JMS Provider Client Properties Files	Program Files\Universal\uai\xml  <div style="background-color: #ffffcc; padding: 5px;">  <b>Note</b> These are suggested locations; you can place these files anywhere on the files system. If you have located these files under the <b>uai</b> directory, you should back them up. </div>
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  <div style="background-color: #ffffcc; padding: 5px;">  <b>Note</b> You can place these files anywhere on the files system. If you have located these files under the <b>uai</b> directory, you should back them up. </div>

## Workload Automation 5 for SOA for Windows - Installation Procedures

Error formatting macro: redirect: java.lang.NullPointerException

## **Workload Automation 5 for SOA for Windows - Installation Procedures Overview**

### **Workload Automation 5 for SOA for Windows - Installation Procedures**

The following procedures are provided for the installation and modification of Workload Automation 5 for SOA for Windows:

- [Installing SOA for Windows via the Graphical Interface](#)
- [Modifying a SOA for Windows Installation via the Graphical Interface](#)
- [Installing SOA for Windows via the Command Line Interface](#)
- [Modifying a SOA for Windows Installation via the Command Line Interface](#)

## Installing SOA for Windows via the Graphical Interface

### Installing Workload Automation 5 for SOA via the Windows Installer Graphical Interface

To install Workload Automation 5 for SOA for Windows using the Windows Installer graphical interface, perform the following steps:

- |               |  |
|---------------|--|
| <b>Step 1</b> | Download the Workload Automation 5 for SOA for Windows product distribution file, <b>sb-soa-5.1.0.&lt;level&gt;-windows-i386.exe</b> , to your work station. |
|---------------|--|

**Step 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 [Windows Installer](#)), you can simply double-click the extracted Workload Automation 5 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.

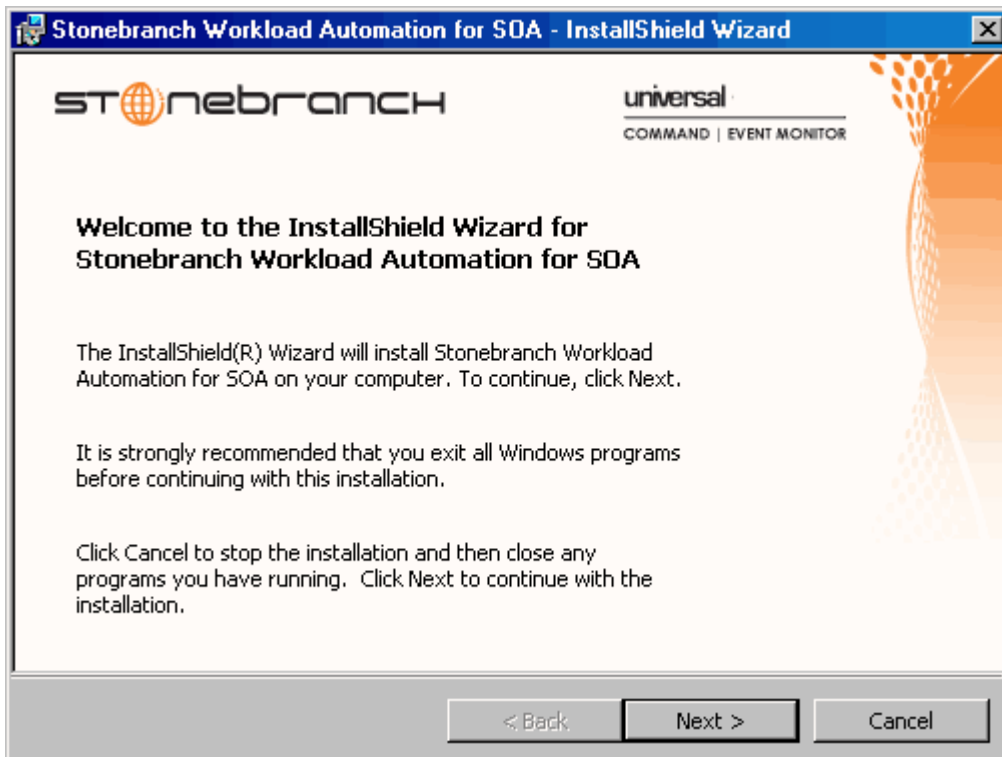
For example, the distribution file's default extraction location is based on the **TEMP** environment variable. The location referenced by this variable can change between Remote Desktop sessions, and any files extracted there may not be accessible after the session is closed.

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.

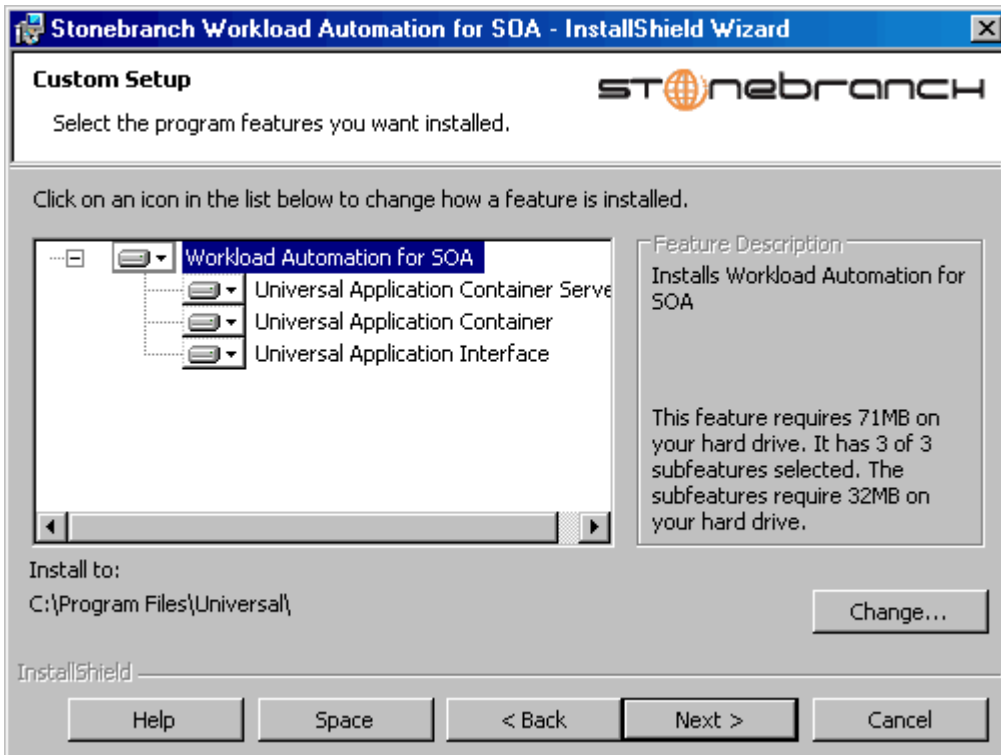
(Refer to the Microsoft documentation on the Remote Desktop feature for additional information.)

The installation automatically begins after the file extraction completes.

It first will verify that your machine meets the minimum system requirements (see [Installation Requirements](#)). If the requirements are met, a Welcome dialog displays.



- Step 3** Click the **Next** button. A list of Workload Automation 5 for SOA components included in the installation package then displays. It is from this list that you can select which components to install.



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.



**A Stonebranch Tip**

If the installation detects an existing Workload Automation 5 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.

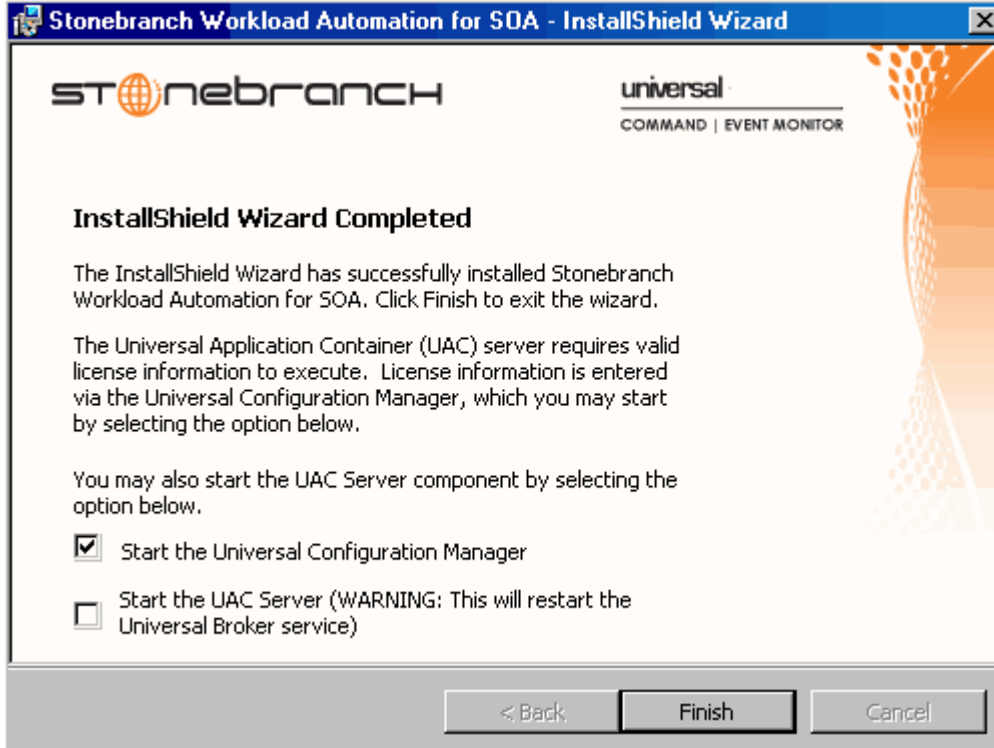
- Step 4** The previous figure shows that all Workload Automation 5 for SOA components will be installed in their respective directories under the **C:\Program Files\Universal** directory.

1. If you want to select a different location, click the **Change...** button.
2. 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.



**Step 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 displays.



**Step 6** If the Universal Application Container (UAC) Server was installed, the following options display on this dialog:

- **Start the Universal Configuration Manager**
- **Start the UAC Server**

License information must be entered into the Universal Application Container Server's configuration before the server will run.

Select both of these options to enter the license information and then start the UAC server.

**Step 7** Click the **Finish** button to exit the Windows installation.

## Modifying a SOA for Windows Installation via the Graphical Interface

- Modifying a Workload Automation 5 for SOA Installation via the Windows Installer Graphical Interface
- Repairing a Corrupted Workload Automation 5 for SOA Installation
- Removing a Workload Automation 5 for SOA Installation
  - Un-Installed Files

### Modifying a Workload Automation 5 for SOA Installation via the Windows Installer Graphical Interface

This section describes how to modify a Workload Automation 5 for SOA installation via the Windows Installer graphical interface.

After installing the Workload Automation 5 for SOA, run the installation programs as needed to modify the installation by:

- Repairing a Corrupted Workload Automation 5 for SOA Installation.
- Removing a Workload Automation 5 Installation.



#### Note

Although Windows Installer provides a Modify selection for Workload Automation 5 for SOA, it cannot be modified; that is, individual components cannot be added or removed.

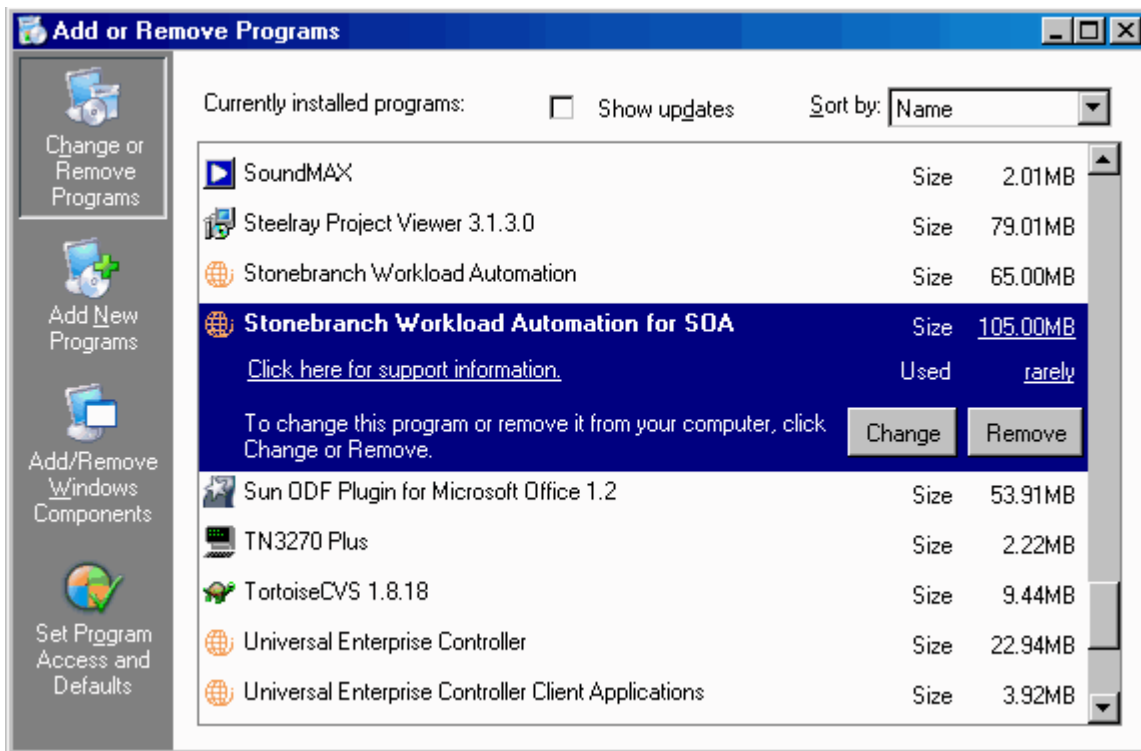
### Repairing a Corrupted Workload Automation 5 for SOA Installation

The installation program has the ability to recover accidentally deleted application files, configuration and component definition files, or registry entries required by Workload Automation 5 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.

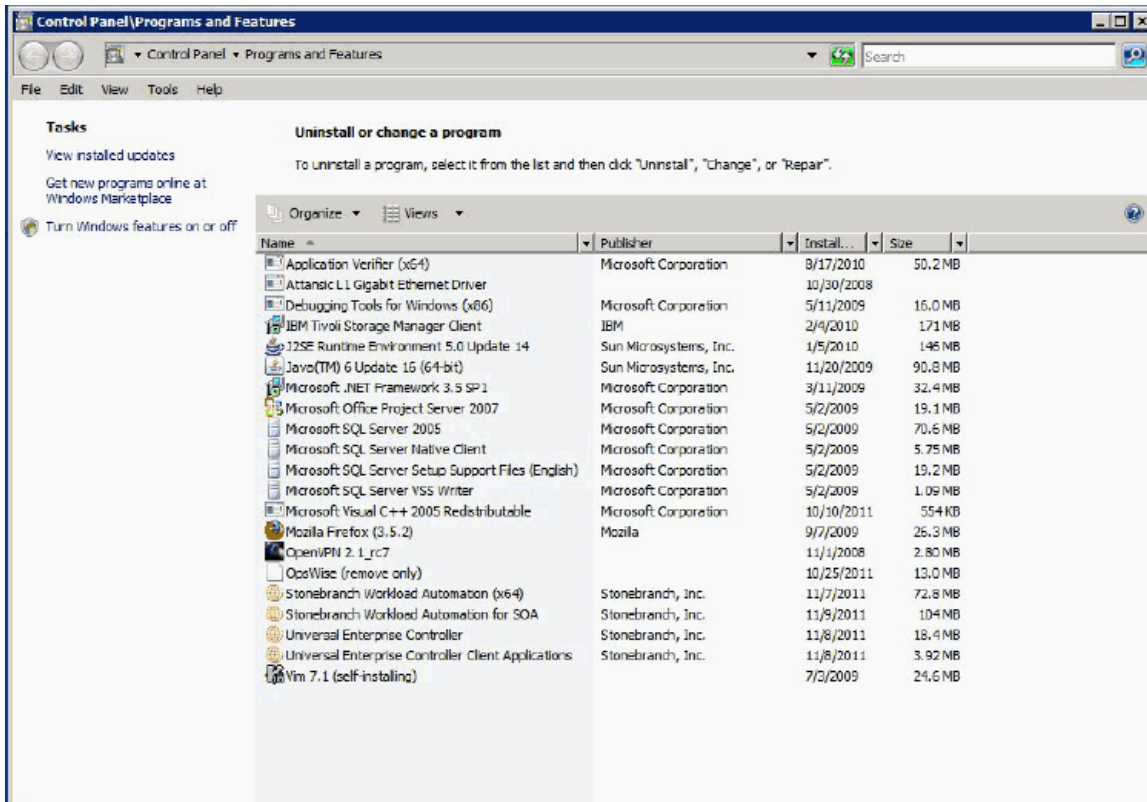
To repair an installation, perform the following steps:

**Step 1** On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays.



**Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**

The Programs and Features dialog replaces the Add or Remove Programs dialog.



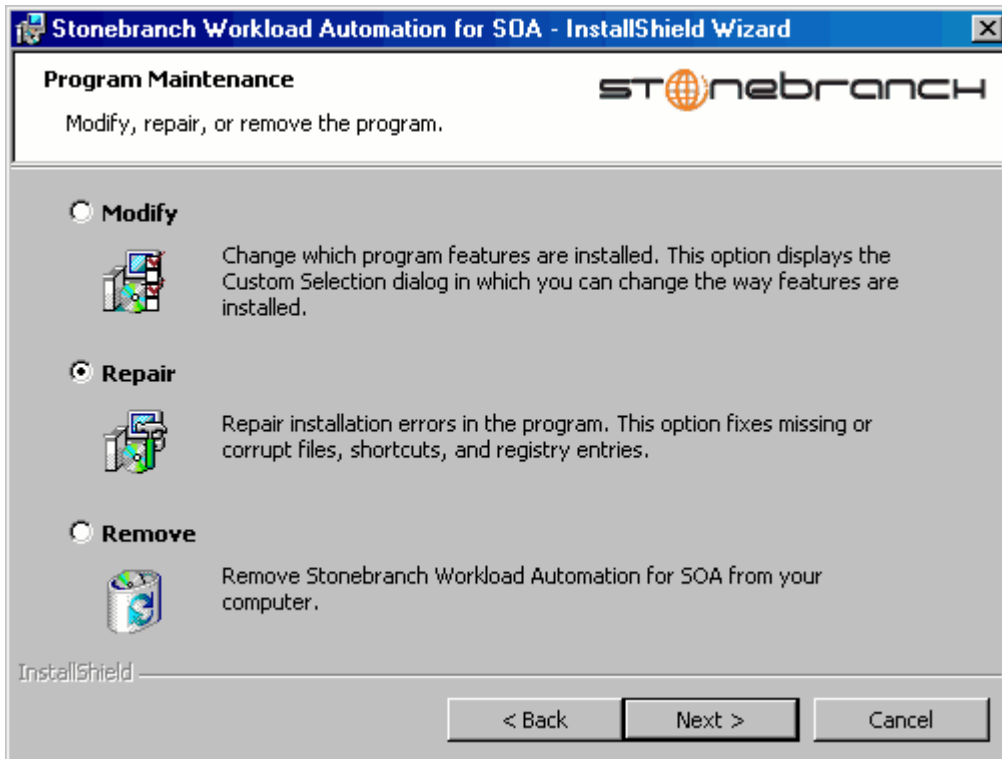
**Step 2** From the list of installed programs, select **Workload Automation 5 for SOA**.

**Step 3** Click the **Change** button to start the installation program.

**Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**

To skip the remaining steps, click "**Repair**".

**Step 4** On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays.



**Step 5** Select **Repair** and click the **Next>** button to display the Ready to Repair dialog.

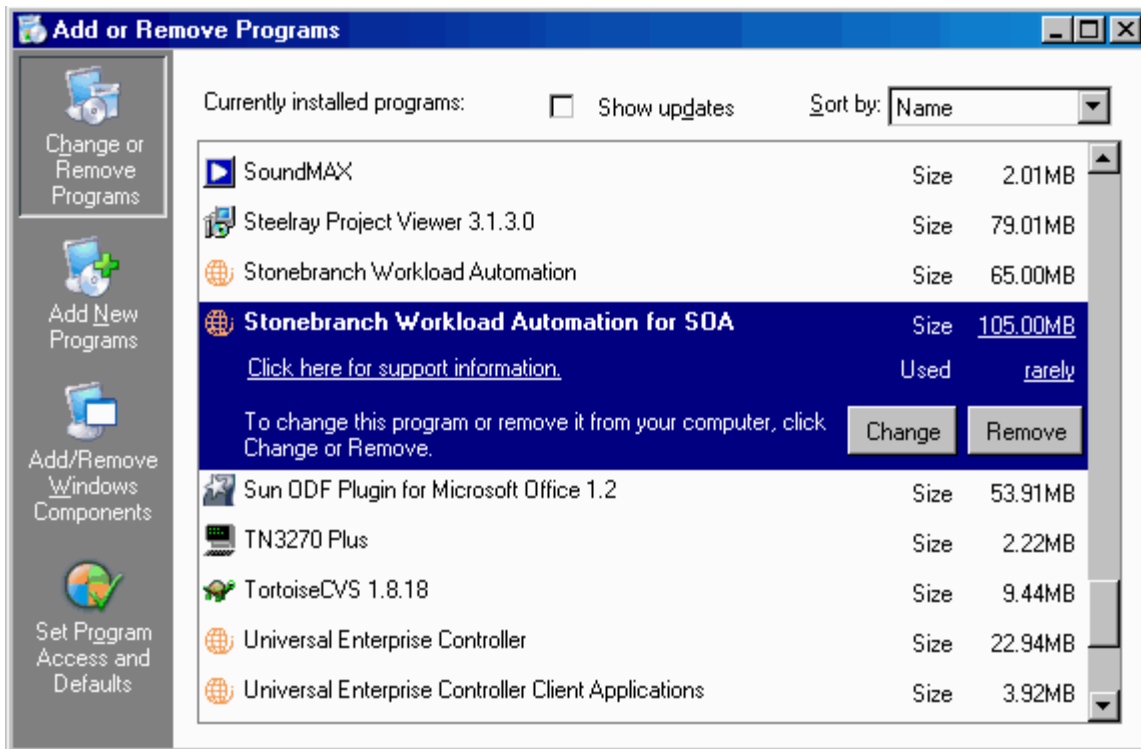
**Step 6** Follow the instructions in successive dialogs to complete the repair.

### Removing a Workload Automation 5 for SOA Installation

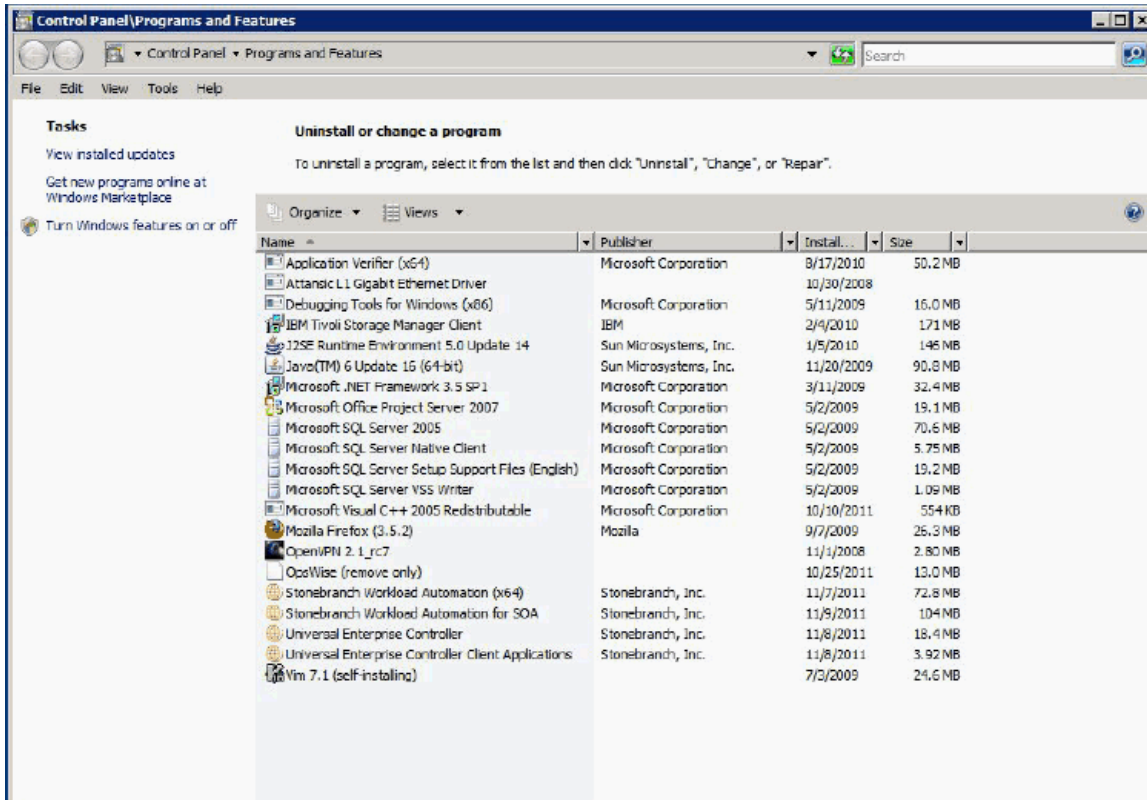
To uninstall a Workload Automation 5 for SOA installation, perform the following steps:

**Step 1**

On the Windows Control Panel, select **Add or Remove Programs**. The Add or Remove Programs dialog displays.



**Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**  
The Programs and Features dialog replaces the Add or Remove Programs dialog.

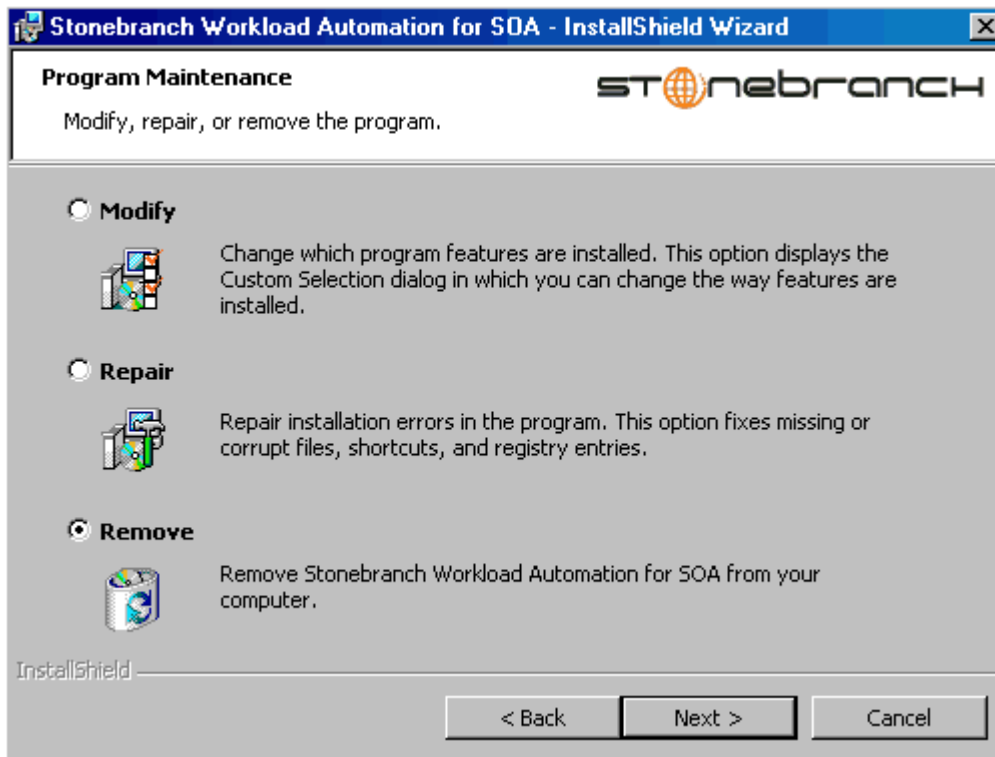


**Step 2** From the list of installed programs, select Workload Automation 5 for SOA.

**Step 3** Click the **Change** button to start the installation program.

**i** **Windows Vista, Windows 7, Windows Server 2008 / 2008 R2**  
To skip the remaining steps, click "Uninstall".

**Step 4** On the Welcome dialog, click the **Next>** button. The Program Maintenance dialog displays.



**Step 5** Select **Remove** and click the **Next>** button.

**Step 6** On the subsequent dialog, click the **Remove** button to remove the Workload Automation 5 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 Workload Automation 5 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 [Workload Automation 5 for SOA for Windows - File Inventory Lists](#) for a list of files and directories installed with Workload Automation 5 for SOA.)

## Installing SOA for Windows via the Command Line Interface

- Installing Workload Automation 5 for SOA via the Windows Installer Command Line Interface
- Windows Installer Command Line Syntax
- Windows Installer Command Line Parameters
- Windows Installer Installation Commands
- Detecting the Completion of a Silent Install
- Using the Distribution File for a Silent Install

### Installing Workload Automation 5 for SOA via the Windows Installer Command Line Interface

This page describes how to install Workload Automation 5 for SOA using the Windows Installer command line interface.

The command line interface is useful in situations where:

- Several Workload Automation 5 for SOA installations must be deployed across many different systems.
- It is not practical or convenient to perform a graphical interface installation.

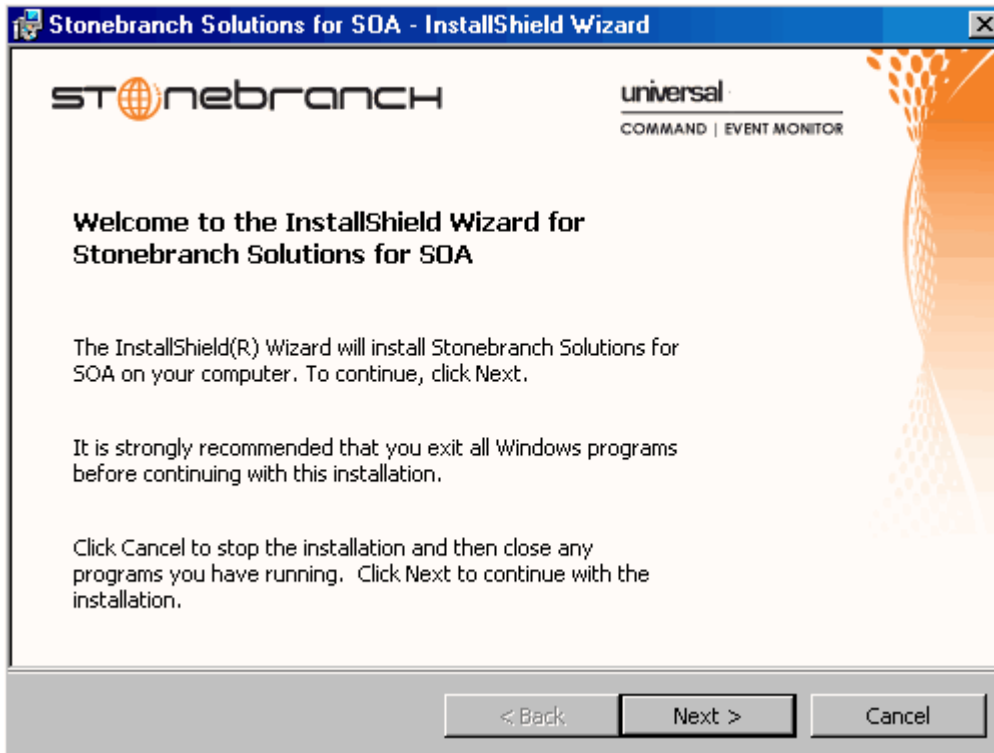
To use the Windows Installer command line interface, the Workload Automation 5 for SOA installation files first must be extracted from the product distribution file. Since there is no command line option available to unpack the distribution file, this must be done via the graphical user interface:

<b>Step 1</b>	Download the Workload Automation 5 for SOA for Windows product distribution file, <code>sb-soa-5.1.0.&lt;level&gt;-windows-i386.exe</code> , to your work station.
<b>Step 2</b>	Execute the distribution file to extract the installation files.
<b>Step 3</b>	On the Location to Save Files dialog, select a location in which to store the installation files and click the <b>Next&gt;</b> button to extract the files.

The screenshot shows a Windows-style dialog box titled "Stonebranch Solutions For SOA - InstallShield Wizard". The main heading is "Location to Save Files" with the question "Where would you like to save your files?". Below this, there is a text box containing the path "C:\Stonebranch Solutions for SOA" and a "Change..." button to its right. At the bottom of the dialog, there are three buttons: "< Back", "Next >", and "Cancel".



**Step 4** On the Welcome dialog, 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 Workload Automation 5 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 Workload Automation 5 for SOA can continue (see [Determining if Windows Installer will be Installed or Upgraded](#)).

A better way to install Workload Automation 5 for SOA 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 Workload Automation 5 for SOA for Windows distribution file (see [Using the Distribution File for a Silent Install](#)).

### Windows Installer Command Line Syntax

The following figure illustrates the command line syntax when using Windows Installer to install Workload Automation 5 for SOA from the command line.

```
msiexec.exe [/i|/fom|/x] SetupPath\UPforSOA.msi
[INSTALLDIR=installdir]
[/q] [/L*v logfilepath]
```

The order in which the parameters are specified is significant.

The **/i**, **/f**, or **/x** command line switches 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 Command Line Parameters

The following table describes the optional parameters that are available when using Windows Installer to install Workload Automation 5 for SOA from the command line.

Parameter	Description	Default
<code>/i</code>	Installs Workload Automation 5 for SOA for Windows.  <i>/i</i> cannot be used with either the <i>/fom</i> (repair) or <i>/x</i> (remove) parameter.	n/a
<code>/fom</code>	Repairs a Workload Automation 5 for SOA for Windows installation.  <b>om</b> (after the <b>f</b> ) are options used by the repair. There are other options available, but for behavior that matches the repair done from the graphical install, the <b>om</b> options must be used.  <i>/fom</i> cannot be used with either the <i>/i</i> (install) or <i>/x</i> (remove) parameter.	n/a
<code>/x</code>	Removes a Workload Automation 5 for SOA for Windows installation.  <i>/x</i> cannot be used with either the <i>/i</i> (install) or <i>/fom</i> (repair) parameter.	n/a
<code>SetupPath \UPforSOA.msi</code>	Specifies the path to the <b>UPforSOA.msi</b> file. <b>SetupPath</b> is the directory in which files extracted from the downloaded distribution package reside.	(none)
<code>INSTALLDIR</code>	Sets the root installation directory to <b>&lt;installdir&gt;</b> . All components are installed in this directory.  <b>INSTALLDIR</b> is required only if you want to install Workload Automation 5 for SOA for Windows under a directory different from the one specified by the <b>PROGRAMFILES</b> environment variable (typically <b>C:\Program Files\Universal</b> ).  <b>INSTALLDIR</b> is valid only when the <i>/i</i> switch is used. Otherwise, it is ignored. If the directory contains spaces, you must use double ( " ) quotation marks around the path name.	(none)
<code>/q</code>	Instructs Windows Installer to run without a graphical interface (silent install).  If <i>/q</i> is omitted from the command line, the Workload Automation 5 for SOA installation is started from the command line, but run with a graphical interface. This is useful when an installation log file is desired.  See <a href="#">Windows Installer Command Line Syntax</a> , <a href="#">Windows Installer Installation Commands</a> , and <a href="#">Detecting the Completion of a Silent Install</a> for additional information regarding silent installs.	n/a
<code>/L*v</code>	Instructs Windows Installer to create an installation log file named <b>&lt;logfilepath&gt;</b> (full path name). If <b>&lt;logfilepath&gt;</b> contains spaces, you must enclose it with double ( " ) quotation marks.  <b>*v</b> are flags used to specify the level of detail ( <b>verbose</b> ) contained in the log file. To reduce the amount of output generated, <b>*v</b> can be omitted. However, using these options is good practice; they can assist Stonebranch Customer Support with problem determination should any errors occur during installation.	n/a

## Windows Installer Installation Commands

The following commands allow installation of Workload Automation 5 for SOA from the command line using Windows Installer (**msiexec.exe**).

To execute an interactive Workload Automation 5 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 Workload Automation 5 for SOA installation, issue the following command:

```
msiexec.exe /i SetupPath\UPforSOA.msi /L C:\Temp\install.log
```

To install Workload Automation 5 for SOA silently to a non-default location, issue the following command:

```
msiexec.exe /i SetupPath\UPforSOA.msi INSTALLDIR=D:\Universal /q
```

To install Workload Automation 5 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 Workload Automation 5 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 Workload Automation 5 for SOA for Windows distribution file, issue the following command:

```
sb-soa-5.1.0.-windows-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.

## Modifying a SOA for Windows Installation via the Command Line Interface

- [Modifying a Workload Automation 5 for SOA Installation via the Windows Installer Command Line Interface](#)
- [Repairing a Corrupted Workload Automation 5 for SOA Installation](#)
- [Removing a Workload Automation 5 for SOA Installation](#)

### Modifying a Workload Automation 5 for SOA Installation via the Windows Installer Command Line Interface

This page describes how to modify a Workload Automation 5 for SOA installation via the Windows Installer command line interface.

After installing Workload Automation 5 for SOA, you may rerun the installation program as needed in order to:

- Repair a Corrupted Workload Automation 5 for SOA Installation.
- Remove a Workload Automation 5 for SOA Installation.

(For a description of the parameters used in these procedures, see [Windows Installer](#).)

### Repairing a Corrupted Workload Automation 5 for SOA Installation

To recover accidentally deleted application files, configuration and component definition files, or registry entries required by Workload Automation 5 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 Workload Automation 5 for SOA distribution file.

To repair a Workload Automation 5 for SOA installation from the command line, issue the following command:

```
msiexec.exe /fom SetupPath\UPforSOA.msi /q
```

To repair a Workload Automation 5 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 Workload Automation 5 for SOA Installation

To uninstall a Workload Automation 5 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
```

## Workload Automation 5 for SOA for Windows - Configuring and Starting UAC Server

### Workload Automation 5 for SOA for Windows – Configuring and Starting the UAC Server

After installing Workload Automation 5 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 Command Agent for SOA 5.1.0 Reference Guide](#).)

**Step 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 Workload Automation 5 applications. Before UAC Server will run, its configuration and component definition information must reside in the local Broker's cache.

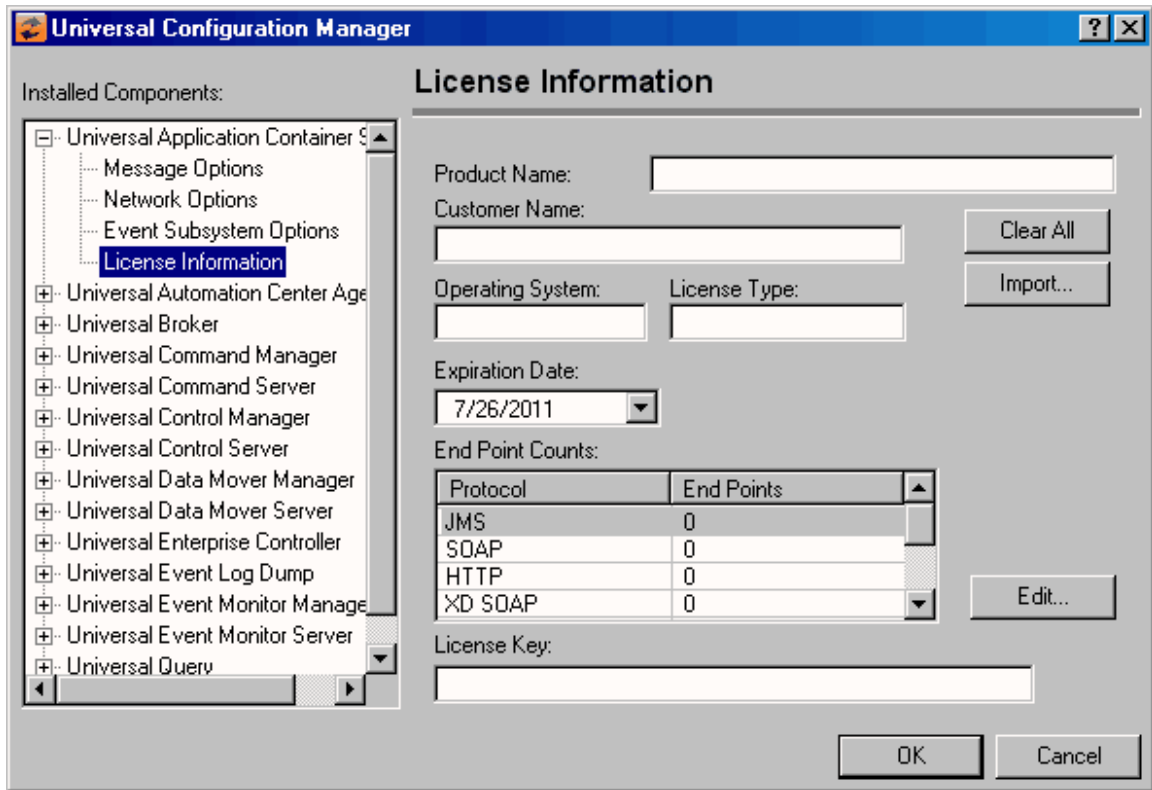
To refresh the local Broker's cache without stopping and restarting the Broker, issue a REFRESH command via Universal Control (For more information, see the [Indesca 5.1.0 User Guide](#).)

**Step 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.

1. Open the Windows Control Panel to start the Universal Configuration Manager.
2. In the Installed Components tree, click on the + symbol next to Universal Application Container Server to display its property pages.
3. Select License Information to display the License Information property page.



4. 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.
5. After entering or importing the license information, click the **OK** button to save the information in the UAC Server configuration file.

**Step 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.

## **Workload Automation 5 for SOA for Windows - 64-Bit Windows Editions**

### **Workload Automation 5 for SOA for Windows – Installing on 64-bit Windows Editions**

All Workload Automation 5 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 Workload Automation 5 for SOA should not require any modification when installing on 64-bit Windows editions.

## Workload Automation 5 for SOA for Windows - File Inventory Lists

- [Workload Automation 5 for SOA for Windows - File Inventory Lists](#)
- Universal Application Container Server
- Universal Application Container
- Universal Application Interface
- Sun Java Runtime Environment, 1.5.0\_14
- System Files

### Workload Automation 5 for SOA for Windows – File Inventory Lists

The Workload Automation 5 for SOA 5.1.0 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 page lists the files installed with each Workload Automation 5 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
<b>UAC\startUAC.bat</b>	Starts the Universal Application Container.
<b>UAC\shutdownUAC.bat</b>	Stops the Universal Application Container.
<b>UAC\UAC.xsd</b>	File used by Universal Application Container to validate content of the <b>UAC.xml</b> inbound configuration file. It requires no user interaction.
<b>UAC\derby.log</b>	Universal Application Container database log file.
<b>%ALLUSERSPROFILE%\Application Data\Universal\confluac_log4jConfiguration.xml</b>	Universal Application Container logging configuration file that you may need to modify if you want to change the logging level of the UAC component.
<b>UAC\INTEventLogAppender.dll</b>	Universal Application Container Windows logging DLL.
<b>UAC\uacValidateInbound.bat</b>	Script that validates the contents of the <b>UAC.xml</b> file using the schema definition contained in <b>UAC.xsd</b> .
<b>UAC\bin\uacsrv.exe</b>	Universal Application Container Server program file.
<b>UAC\bin\uacscfg.dll</b>	Used by Universal Configuration Manager to manage Universal Application Container Server configuration options.
<b>UAC\bin\uacscfg.hlp</b>	Universal Application Container Server configuration help file.



<b>UAC\lib\uacsv.jar</b>	Universal Application Container Server class file.
<b>UBroker\tml\uaccfg</b>	Universal Application Container Server configuration template file.
<b>UBroker\tml\uaccmp</b>	Universal Application Container Server component definition template file.
<b>nls\uacmceng.umc</b>	Universal Application Container Server message catalog.
<b>%ALLUSERSPROFILE%\Application Data\Universal\conf\UAC.xml</b>	Inbound operation configuration file that you may need to modify to support your inbound workload operations.
<b>%ALLUSERSPROFILE%\Application Data\Universal\conf\uacs.conf</b>	Universal Application Container Server configuration file.
<b>%ALLUSERSPROFILE%\Application Data\Universal\compluac</b>	Universal Application Container Server component definition file.
<p><b>%ALLUSERSPROFILE%\ Application Data</b>, by default, resolves to:</p> <ul style="list-style-type: none"> <li>• <b>C:\Documents and Settings\All Users\Application Data</b> on Windows 2003 and XP.</li> <li>• <b>C:\ProgramData</b> on Windows Vista / Windows 2008 Server and later.</li> </ul>	

## Universal Application Container

File	Description
<b>UAC\container</b>	Subdirectory containing the libraries and other deployable objects needed for UAC operation.

## Universal Application Interface

File	Description
<b>UAI\lib</b>	Subdirectory containing the <b>uacsv.jar</b> file needed for communication between the <b>uacsv</b> and <b>uac</b> components. It requires no user interaction.
<b>UAI.keystore</b>	File that contains the certificates for secure communication between UAC and UAI.
<b>%ALLUSERSPROFILE%\Application Data\Universal\conf\uai_log4jConfiguration.xml</b>	Universal Application Interface logging configuration file.
<b>UAI\uai.bat</b>	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 this table are relative to the %SYSTEMROOT% directory, where %SYSTEMROOT% is an environment variable that resolves to C:\Windows on all 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 <sup>1</sup>	Version 8.0.50727.762 of the Microsoft C runtime side-by-side assembly.
Microsoft Foundation Classes v8.0.50727.762 <sup>2</sup>	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 <a href="#">Windows Installer</a> 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.

<sup>1</sup> 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.

<sup>2</sup> 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.

## UNIX Installation

Error formatting macro: redirect: java.lang.NullPointerException

## UNIX Installation - Overview

- [Workload Automation 5 for UNIX Installation](#)
  - [UNIX Packages](#)
- [Installation Methods](#)
- [Installation Summary](#)

## Workload Automation 5 for UNIX Installation

These pages provide information on the installation of Stonebranch, Inc.'s Workload Automation 5 on UNIX operating systems. Unless otherwise specified, all references to Workload Automation 5 for UNIX refer to version 5.1.0.



### Note

This information does not include installation on z/OS UNIX System Services (USS). See [zOS USS Installation](#) for z/OS USS installation instructions.

## UNIX Packages

Workload Automation 5.1.0 for UNIX is provided in two separate easily installed, one-time installation packages:

- [Workload Automation 5 for UNIX](#)
- [Workload Automation 5 for SOA for UNIX](#)



### Note

Starting with the 3.2.0 release of Universal Products, a Universal Broker must run on all systems on which a Workload Automation 5 component is running, including manager components. The Broker maintains product configuration data for all components that have a configuration file.

## 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 5.3 and above
- HP-UX 11.11 and above
- Solaris 8 and above
- Linux
  - 2.4 kernel for x86-based systems
  - 2.6 kernel for x86\_64-based, zSeries 64-bit (S/390), and Itanium (ia64) systems
  - Redhat Package Manager (RPM)

Workload Automation 5 for UNIX for all other types of UNIX operating systems are installed with the generic Workload Automation 5 for UNIX installation script.

## Installation Summary

<b>Step 1</b>	Download the distribution file from the Stonebranch <a href="#">Customer Portal</a> .
<b>Step 2</b>	Copy the distribution file to the UNIX system.
<b>Step 3</b>	Logon to the UNIX system as root or execute <b>su</b> (substitute user) command to root.
<b>Step 4</b>	Uncompress the distribution file.
<b>Step 5</b>	Extract the files from the uncompressed <b>tar</b> file.
<b>Step 6</b>	Run the Stonebranch, Inc.-supplied script.

## UNIX Installation - Licensing

- Licensing Workload Automation 5 for UNIX Components
- Product License File
  - Format
  - Sample
- Entering License Information
- Restart Universal Broker

### Licensing Workload Automation 5 for UNIX Components

After Workload Automation 5 for UNIX has been installed, you must configure the following components with product licenses before they can be used:

- Universal Command Manager
- Universal Data Mover Manager
- Universal Connector
- Universal Event Monitor Server
- Universal Application Container Server



#### Note

Universal Application Container Server (as a component of Universal Command Agent for SOA 5.1.0 for UNIX) is packaged, and licensed, separately.

### Product License File

For each component, product license information (license parameter keywords and their values) is contained in a separate text file provided by your Stonebranch, Inc. account representative.

#### Format

The format of the product license file name is: *<component name>\_<customer name>\_<operating system>\_<schedule or solution>.txt*. For example: **Indesca\_Stonebranch\_UNIX\_A1.txt**.

(For Universal Command Manager, **Indesca** is used as the *<component name>* in the product license file name and as the name of the product in the product license file itself; for Universal Data Mover Manager, **Inftran** is used as the *<component name>* in the product license file name and as the name of the product in the product license file itself - see below.)

#### Sample

The following is a sample Universal Command Manager for UNIX product license file:

```
License_Product "INDESCA"
License_Customer "STONEBRANCH"
License_OS_Type "UNIX"
License_Type "PERPETUAL"
License_Expiration_Date 2029.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

### Entering License Information

Enter each component's product license file information into its configuration file:

- Universal Command Manager: **ucmd.conf**
- Universal Data Mover Manager: **udm.conf**

- Universal Connector: **usap.conf**
- Universal Event Monitor Server: **uems.conf**
- Universal Application Container Server: **uacs.conf**

It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)

## Restart Universal Broker

For Universal Broker to read the license information, you must stop and restart it:

<b>Stop Universal Broker</b>	<pre>ubrokerd stop</pre>
<b>Start Universal Broker</b>	<pre>ubrokerd start</pre>

## **Workload Automation 5 for UNIX**

Error formatting macro: redirect: java.lang.NullPointerException

## Workload Automation 5 for UNIX - Installation Package

- [Components](#)
- [Component Compatibility](#)
- [Licensing](#)

### Components

The Workload Automation 5.1.0 for UNIX package includes the following product components:

- Universal Broker 5.1.0
- Universal Automation Center Agent 5.1.0 (for AIX, HP-UX, Solaris, and Linux)
- Universal Certificate 5.1.0
- Universal Command Manager and Server 5.1.0
- Universal Connector 5.1.0 (for AIX, HP-UX, Solaris, and UNIX)
- Universal Control Manager and Server 5.1.0
- Universal Data Mover Manager and Server 5.1.0
- Universal Encrypt 5.1.0
- Universal Event Monitor Manager and Server 5.1.0
- Universal Message to Exit Code Translator 5.1.0
- Universal Query 5.1.0

### Component Compatibility

The following table identifies the compatibility of Workload Automation 5.1.0 for UNIX components with previous component / product versions.

Component	Compatibility
Universal Broker 5.1.0	Workload Automation 5 / Universal Products releases 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Automation Center Agent 5.1.0	Universal Automation Center Agent is compatible with Automation Center 5.1.0 and above.
Universal Command 5.1.0	Universal Command 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Control 5.1.0	Universal Control 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Data Mover 5.1.0	Universal Data Mover 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1 and 3.1.0.
Universal Encrypt 5.1.0	Universal Encrypt 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Query 5.1.0	Universal Broker 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, 2.2.0, and 2.1.0.
Universal Event Monitor 5.1.0	Universal Event Monitor 4.3.0, 4.2.0, 4.1.0, 3.2.0, and 3.1.0.

The component references pertain to all supported UNIX platforms for that version.

### Licensing

For licensing information, see [UNIX Installation - Licensing](#).



## Workload Automation 5 for UNIX - Installation Requirements

- System Requirements
- Platform Requirements
- Directories and Files
  - Command Reference Directory
  - Log Directory
  - Trace Directory
  - Spool Directory
  - UAG Cache
- UAG Crossgrade

### System Requirements

Workload Automation 5 for UNIX system requirements are:

1. One of the supported UNIX operating systems. Currently, the following UNIX operating systems are supported by Workload Automation:
  - AIX 5.3 TL9 and above.
  - HP-UX 11.11 with Minimum GOLDQPK11i\_B.11.11.0612.459 Update and above.
  - HP-UX 11.23 and above (HP-UX IA64 package)
  - Solaris 8 and above (SPARC-based).
  - Solaris 10 and above (Intel-based).
  - Linux Environments with the following qualifications:
    - 2.4 kernel: x86-based systems.
    - 2.6 kernel: x86\_64-based, zSeries 64-bit (S/390), and Itanium (ia64) systems.
    - RedHat Package Manager (RPM).



#### Note

Workload Automation's native 64-bit package for x86\_x64-based systems requires glibc 2.5-42 or later.

- MP-RAS.
- Tru64 version 5.1 and above.

The list of supported systems grows rapidly. Contact Stonebranch, Inc. for a current list of supported UNIX operating systems if you require support for a system that is not listed.

2. Superuser (root) access.
3. TCP/IP Socket implementation.
4. Approximately 60 megabytes of disk space for the installation. More disk space is required for variable files, such as log files, spool files, and trace files.
5. Bourne shell or compatible.



#### Note

In order to install Workload Automation 5 components, you must be able to write to the directory from which the installation is launched.

### Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opswise Automation Center 5.1.1](#) and [Indesca-Infitrans 5.1.0](#) page to make sure that your platform is supported before performing an installation.

### Directories and Files

All product files that are written to during product execution are stored in the `/var/opt/universal` 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/universal/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 Workload Automation 5 components 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 Workload Automation 5 components 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 Workload Automation 5 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/universal/spool`** directory.

### **Universal Command Server**

#### **Space**

Spool files are located, by default, in directory **`/var/opt/universal/spool`**. 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.



#### A Stonebranch 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:

**MAX-PROCESSES x AVERAGE-STDIO-SIZE x 2 = 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.

For more information on the Manager Fault Tolerant feature and the spooling of redirected standard I/O files, see the [Indesca 5.1.0 User Guide](#) and [Infitran 5.1.0 User Guide](#).

#### Security

Universal Broker, Universal Command Server and Universal Event Monitor require read/write access to the spool directory. No other Workload Automation 5 components 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.

#### UAG Cache

UAG cache is used by Universal Automation Center Agent (UAG) for storing standard I/O files.

#### Space

Cache files are located, by default, in directory `/var/opt/universal/uag/cache`.

Cache files are created for each job that is run by the Automation Center agent. They remain in the cache until they are purged by an automated purge process scheduled nightly by the Automation Center Core process. You can configure the number of days that files remain in the cache using the user interface module Automation Center Administration->Properties->Agent Cache Retention Period In Days (for more information, see [Opswise Properties](#)).

The amount of disk space required for the cache directory depends on:

1. Number of jobs you estimate will run during the cache retention period you specified.
2. These files remain until they are purged by the automated cache purge process scheduled by the Automation Center Core daily at midnight.
3. Average size of the user processes standard output and error files.

When these numbers have been determined, the average amount of disk space is calculated with the following formula:

**(RETENTION PERIOD x MAX-JOBS) x (AVERAGE-STDOUT-SIZE + AVERAGE-STDERR-SIZE) = required disk space.**

For example:

If the files are purged every 7 days, and you run 1200 jobs on that agent server daily, and the average size of the STDOUT + STDERR files is 3,000 bytes, the average amount of required disk space is 25MB (7 x 1,200 x 3000).

The Universal Automation Center Agent Server automatically redirects the standard output and standard error files to the cache directory with no

required user input.

### **Security**

Universal Automation Center Agent require read/write access to the UAG cache directory. No other Workload Automation 5 components access the cache directory. No general user access is required.

### **UAG Crossgrade**

The Universal Automation Center Agent (UAG) provided by the Workload Automation 5 for UNIX installation replaces all previous versions (1.7 and earlier) of the Opswise Automation Center Agent.

If you are upgrading from OpsWise 1.7 and earlier to OpsWise 5.1, you must perform a UAG crossgrade (see [Migrating an Opswise Agent to UAG for Workload Automation 5 for UNIX](#)).

## Workload Automation 5 for UNIX - Installation Upgrades

- [Workload Automation for UNIX - Installation Upgrades](#)
- [Universal Command 2.2.0](#)
  - [Installation Method](#)
  - [Product Directories](#)
  - [Configuration Files](#)
- [All Workload Automation 5 Releases](#)
- [Platform Requirements](#)

### Workload Automation for UNIX – Installation Upgrades

This page 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.

### Universal Command 2.2.0

#### Installation Method

Starting with Universal Command 2.2.0, the generic Workload Automation 5 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 Workload Automation 5 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/universal`, the variable files directory is `/var/opt/universal`, and the configuration files directory is `/etc/universal`.

Executable files from versions prior to Universal Command 2.2.0 (**ucmd**, **uctl**, **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 Workload Automation 5 are stored. Configuration files will be moved from their current location to the `/etc/universal` directory.

### All Workload Automation 5 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 Workload Automation 5 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.

### Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opwise Automation Center 5.1.1](#) and [Indesca-Infitran 5.1.0](#) page to make sure that your platform is supported before upgrading.

## Workload Automation 5 for UNIX - Distribution File

- [UNIX Distribution File](#)
- [Obtaining the Distribution File](#)
- [Distribution File Format](#)

### UNIX Distribution File

Stonebranch, Inc. provides different Workload Automation for UNIX packages for different types of UNIX operating systems.

### Obtaining the Distribution File

To obtain the Workload Automation for UNIX package for your type of UNIX operating system, you must download the corresponding product distribution file from the Stonebranch [Customer Portal](#).



#### Note

A customer user name and password — provided by Stonebranch, Inc. — are required to access the Customer Portal.

After a distribution file has been downloaded, the installation files contained in that file must be extracted before the product can be installed (see [Workload Automation 5 for UNIX - Installation Procedures](#)).

### Distribution File Format

The name of each Workload Automation for UNIX distribution file has the following format:

**sb-Version.Release.Modification Level.Maintenance Level-operating system-version(.release)(-platform).tar.Z**

For example: **sb-5.1.0.0-linux-2.6-ia64.tar.Z**

In this format:

- **Version** is the current version of Workload Automation 5.
- **Release** is the current release of Workload Automation 5.
- **Modification Level** is the current Workload Automation 5 feature set.
- **Maintenance Level** is the Workload Automation 5 build level.
- **operating system** is the name of the operating system (for example, AIX or Linux).
- **version(.release)** is the supported version and, optionally, the release of the operating system.
- **platform** 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.

## Workload Automation 5 for UNIX - Installation Procedures

Error formatting macro: redirect: java.lang.NullPointerException

## **Workload Automation 5 for UNIX - Installation Procedures Overview**

### **Workload Automation 5 for UNIX Installation Procedures**

The following procedures are provided for the installation and modification of Workload Automation 5 for UNIX:

- [Workload Automation 5 for AIX Installation](#)
- [Workload Automation 5 for HP-UX Installation](#)
- [Workload Automation 5 for Solaris Installation](#)
- [Workload Automation 5 for Linux Installation](#)
- [Workload Automation 5 for UNIX Installation \(Generic\)](#)
- [Migrating an Opwise Agent to UAG for Workload Automation 5 for UNIX](#)



## Workload Automation 5 for AIX Installation

- [Installation Process](#)
- [Extracting the Workload Automation 5 for AIX Installation Files](#)
  - [Distribution File](#)
- [AIX Packages](#)
- [Installing Workload Automation 5 for AIX](#)
  - [Starting the Installation Script](#)
  - [AIX Installation Script Parameters](#)
  - [Installation Script Example](#)
  - [AIX PAM Customization](#)
- [Listing Workload Automation 5 for AIX Information](#)
- [Removing Workload Automation 5 for AIX](#)

### Installation Process

Installation of Workload Automation for AIX is a three-step process:

<b>Step 1</b>	Download the product <a href="#">distribution file</a> .
<b>Step 2</b>	Extract the installation files from the distribution file.
<b>Step 3</b>	Install the extracted files.

### Extracting the Workload Automation 5 for AIX Installation Files

The Workload Automation 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 sb-5.1.0.0-aix-5.3.tar.Z | tar xvf -
```

This command assumes the following:

- Name of the distribution file is **sb-5.1.0.0-aix-5.3.tar.Z**.
- File is located in the current working directory.

### Distribution File

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 [Distribution File Format](#) for distribution file naming conventions.)

The following table identifies the files contained in the distribution file.

File	Description
<b>Readme.unv</b>	Summary of the installation procedure.
<b>uag-5.1.0.0-aix-5.3.bff</b>	AIX backup file format package for the base UAG product.
<b>ubr-5.1.0.0-aix-5.3.bff</b>	AIX backup file format package for the base UBROKER product.
<b>ucmd-5.1.0.0-aix-5.3.bff</b>	AIX backup file format package for the base UCMD product.
<b>ucom-5.1.0.0-aix-5.3.bff</b>	AIX backup file format package for the base COMMON package.
<b>ucr-5.1.0.0-aix-5.3.bff</b>	AIX backup file format package for the base UCERT package.
<b>udm-5.1.0.0-aix-5.3.bff</b>	AIX backup file format package for the base UDM product.
<b>uem-5.1.0.0-aix-5.3.bff</b>	AIX backup file format package for the base UEM product.
<b>unvinst</b>	Installation script.
<b>upimerge.sh</b>	Script that uses the Universal Installation Merge (UIM) module.

<b>usap-5.1.0.0-aix-5.3.bff</b>	AIX backup file format package for the base USAP product.
<b>util-5.1.0.0-aix-5.3.bff</b>	AIX backup file format package for the base UTILITIES package.

**Note**

If your Workload Automation 5 for AIX distribution file does not contain all of these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

**AIX Packages**

The following table identifies the filesets contained in the AIX packages.

Package	Fileset Name	Description
UNVcom	<b>UNVcom.UNVcom.msg.en_US</b>	Workload Automation 5 common files, such as message catalogs and translation tables.
UNVuag	<b>UNVuag.UNVuag.rte</b>	Universal Automation Center Agent
UNVubr	<b>UNVubr.UNVctl.rte</b> <b>UNVubr.UNVcts.rte</b> <b>UNVubr.UNVqry.rte</b> <b>UNVubr.UNVspl.rte</b> <b>UNVubr.UNVubr.rte</b>	Universal Control Manager Universal Control Server Universal Query Manager Universal Spool Maintenance Utilities Universal Broker
UNVuclm	<b>UNVuclm.UNVuclm.rte</b> <b>UNVuclm.UNVucls.rte</b>	Universal Command Manager Universal Command Server
UNVucl	<b>UNVucl.UNVucl.rte</b>	Universal Certificate
UNVudm	<b>UNVudm.UNVdmgr.rte</b> <b>UNVudm.UNVdsrv.rte</b>	Universal Data Mover Manager Universal Data Mover Server
UNVuem	<b>UNVuem.UNVuem.rte</b> <b>UNVuem.UNVues.rte</b>	Universal Event Monitor Manager Universal Event Monitor Server
UNVusp	<b>UNVusp.UNVusp.rte</b>	Universal Connector
UNVutil	<b>UNVutil.UNVutil.rte</b>	Universal Utilities

**Installing Workload Automation 5 for AIX**

Workload Automation 5 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.

**Starting the Installation Script**

To start the installation script, **unvinst**, and install all of the AIX packages, issue the following command:

```
sh ./unvinst [--user username [--userdir dir]] [--group groupname] [--convert_opsagent option
[--opsdir dir]] [--ac_transports list]
[--ac_core core] [--info]
```

See [AIX Installation Script Parameters](#) and [Installation Script Example](#), below, for a description of the optional parameters that you can issue with **unvinst** and an example of **unvinst** with these parameters.

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-5.1.0/ubrokerd start
```

The Universal Broker daemon will be installed and run as the **username** and **groupname** specified with the [installation script parameters](#), below.

**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.

**AIX Installation Script Parameters**

The following table describes the optional parameters that are available in the installation script, **unvinst**, when installing Workload Automation 5.

Parameter	Description	Default
<pre>-u -user --user</pre>	<p>Normal UNIX <b>username</b> that is used to execute the Universal Broker daemon. The install grants this user account ownership of all installed files, with the exception of the Workload Automation server components (for example: <b>ucmsrv</b>, <b>udmsrv</b>, and <b>uemsrv</b>) which, due to security requirements, are owned by root and will have their "set user ID on execution" bit set.</p> <p>If the user account that you want to use already exists, specify that user account.</p> <p>If this user account does not exist, the install creates it. If <b>--user</b> is omitted from <b>unvinst</b>, the default user account (<b>ubroker</b>) is used.</p> <p>If you want to change the user account for an installed Workload Automation 5 for AIX system, you must perform a re-installation and use this parameter to change the user account.</p>	<b>ubroker</b>
<pre>-userdir --userdir</pre>	<p>Home directory for the created user account specified by <b>--user</b>.</p> <p>If this directory does not exist, it is created when the specified user is created. If <b>--userdir</b> is omitted from <b>unvinst</b>, the default is used.</p>	<b>/home/&lt;username&gt;</b>
<pre>-g -group --group</pre>	<p>Normal UNIX <b>groupname</b>; the Universal Broker daemon will run as this specified group. All installed files will be assigned to this group.</p> <p>If the group that you want to use already exists, specify that group.</p> <p>If this group does not exist, the install creates it. If <b>--group</b> is omitted from <b>unvinst</b>, the default group (<b>ubroker</b>) is used.</p>	<b>ubroker</b>
<pre>-c -convert --convert_opsagent</pre>	<p>Specification (<b>yes</b> or <b>no</b>) that causes <b>unvinst</b> to execute <b>opsmerge.sh</b>, which resides in <b>/opt/universal/uagsrv/bin</b>.</p> <p><b>opsmerge.sh</b> performs the following tasks:</p> <ol style="list-style-type: none"> <li>1. Searches for an existing Opswise Agent install (1.5 or later) and converts configuration options stored in the <b>agent.props</b> file to corresponding options in <b>uags.conf</b>.</li> <li>2. Searches for an active Opswise Agent daemon process and attempts to stop it.</li> <li>3. Assigns the ID used by the Opswise Agent to UAG, by moving the <b>qname</b> file from the Opswise install directory to the <b>/var/opt/universal/uag/var</b> directory.</li> </ol>	<b>yes</b>
<pre>-d -opsdir --opsdir</pre>	<p>Identifies the primary install directory for the Opswise Agent when <b>--convert_opsagent</b> is specified.</p> <p>If <b>--opsdir</b> is omitted from <b>unvinst</b>, the default is used.</p>	<b>/home/opswise</b>

<pre>-t --transports --ac_transports</pre>	<p>Specifies a value, in the format <code>port@ipaddr[,port2@ipaddr2,...,portn@ipaddrn]</code>, for the port and network address of the Opwise Automation Center Transporter(s) used for network communication. The install uses this value to set the UAG <code>AUTOMATION_CENTER_TRANSPORTS</code> configuration option. The value specified in <code>--ac_transports</code> will override any currently configured options, including those imported from <code>agent.props</code> when <code>--convert_opsagent</code> is set to <b>yes</b>.</p> <p>If <code>--ac_transports</code> is omitted from <code>unvinst</code>, and no other value specifies the port and network address of the Transporter(s), the default is used.</p>	4803@127.0.0.1
<pre>-r --core --ac_core</pre>	<p>Specifies a value for the queue name of the Opwise Automation Center Message Hub. The install uses this value to set the UAG <code>AUTOMATION_CENTER_CORE</code> configuration option. The value specified in <code>--ac_core</code> will override any currently configured options, including those imported from <code>agent.props</code> when <code>--convert_opsagent</code> is set to <b>yes</b>.</p> <p>If <code>--ac_core</code> is omitted from <code>unvinst</code>, and no other value specifies the queue name of the Opwise Automation Center Message Hub, the default is used.</p>	HUB01
<pre>-? -h --help --info</pre>	Displays command line help.	n/a

### Installation Script Example

The following example illustrates Workload Automation 5 for AIX installed with the installation script, `unvinst`, and optional parameters.

```
sh ./unvinst --user user1 --userdir /homedir/user --group usergroup --convert_opsagent no --opsdir
/homedir/ops --ac_transports 4803@127.0.0.1
--ac_core HUB01
```

### AIX PAM Customization

When security is set to use PAM, the `pam.conf` file under `/etc` must be adjusted.

Below is the minimum required PAM service definition to make Workload Automation 5 function:

ucmd	auth	required	/usr/lib/security/pam_aix
ucmd	account	required	/usr/lib/security/pam_aix

### Listing Workload Automation 5 for AIX Information

On AIX, information on an installed product is listed with the `lspp` command. The command must be executed with the superuser ID.

Issue the following command to list information for Workload Automation 5 for AIX:

```
lspp -La 'UNV*'
```

### Removing Workload Automation 5 for AIX

**Step 1** Stop the `ubrokerd` daemon.

<b>Step 2</b>	Using the superuser ID, remove all filesets of Workload Automation 5 for AIX by issuing the following command: <pre>installp -u UNVusp UNVuag UNVucr UNVutl UNVuem UNVudm UNVucm UNVubr UNVcom</pre> All entries in the <code>/etc/inittab</code> file that reference <code>ubroker</code> are removed.
<b>Step 3</b>	Delete the Agent user account ( <code>ubroker</code> ) and its home directory: <pre>userdel -r ubroker</pre>
<b>Step 4</b>	Delete the Agent group ( <code>ubroker</code> ): <pre>groupdel ubroker</pre>

## Workload Automation 5 for HP-UX Installation

- Installation Process
- Extracting Workload Automation for HP-UX Installation Files
  - PA-RISC Systems
  - IA64 Systems
- HP-UX Package
- Pre-Requisite to Installing Workload Automation 5 for HP-UX (PA-RISC Systems)
- Installing Workload Automation 5 for HP-UX
  - Starting the Installation Script
  - HP-UX Install Script Parameters
  - Installation Script Example
- Listing Workload Automation 5 for HP-UX Information
- Removing Workload Automation 5 for HP-UX

### Installation Process

Installation of Workload Automation for HP-UX is a three-step process:

<b>Step 1</b>	Download product distribution file (see <a href="#">Workload Automation 5 for UNIX - Distribution File</a> ).
<b>Step 2</b>	Extract the installation files from the distribution file.
<b>Step 3</b>	Install the extracted files.

### Extracting Workload Automation for HP-UX Installation Files

Stonebranch, Inc. provides separate product distribution files for the following HP-UX system:

- PA-RISC Systems
- IA64 Systems

Workload Automation for HP-UX product distribution files are in a compressed **tar** format.

#### PA-RISC Systems

To uncompress and extract the installation files from the distribution file, issue the following command:

```
zcat sb-5.1.0.0-hpux-11.11-hppa.tar.Z | tar xvf -
```

This command assumes the following:

- Name of the distribution file is **sb-5.1.0.0-hpux-11.11-hppa.tar.Z**.
- File is located in the current working directory.

#### Distribution File

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 [Distribution File Format](#) for distribution file naming conventions.)

The following table identifies the files contained in the distribution file.

File	Description
<b>Readme.unv</b>	Summary of the installation procedure.
<b>unv-5.1.0.0-hpux-11.11-hppa.depot</b>	HP-UX .depot file-format package.
<b>unvinst</b>	Installation script.
<b>upimerge.sh</b>	Script that uses the Universal Installation Merge (UIM) module.

If your Workload Automation 5 for HP-UX distribution file does not contain these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

## IA64 Systems

To uncompress and extract the installation files from the distribution file, issue the following command:

```
zcat sb-5.1.0.0-hpux-11.23-ia64.tar.Z | tar xvf -
```

This command assumes the following:

- Name of the distribution file is **sb-5.1.0.0-hpux-11.23-ia64.tar.Z**.
- File is located in the current working directory.

### Distribution File

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 [Distribution File Format](#) for distribution file naming conventions.)

The following table identifies the files contained in the distribution file.

File	Description
<b>Readme.unv</b>	Summary of the installation procedure.
<b>unv-5.1.0.0-hpux-11.23-ia64.depot</b>	HP-UX .depot file-format package.
<b>unvinst</b>	Installation script.
<b>upimerge.sh</b>	Script that uses the Universal Installation Merge (UIM) module.



#### Note

If your Workload Automation 5 for HP-UX distribution file does not contain all of these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

## HP-UX Package

The Workload Automation 5 for HP-UX is packaged as a depot file (extension **.depot**). The Workload Automation 5 depot contains product UCM.

The following table identifies the sub-products contained in the HP-UX package.

Subproduct Name	Description
<b>UNV.UNVcom</b>	Workload Automation 5 common files, such as message catalogs and translation tables.
<b>UNV.UNVctl</b>	Universal Control Manager
<b>UNV.UNVcts</b>	Universal Control Server
<b>UNV.UNVqry</b>	Universal Query
<b>UNV.UNVspl</b>	Universal Spool
<b>UNV.UNVuag</b>	Universal Automation Center Agent
<b>UNV.UNVubr</b>	Universal Broker
<b>UNV.UNVucm</b>	Universal Command Manager
<b>UNV.UNVucr</b>	Universal Certificate
<b>UNV.UNVucs</b>	Universal Command Server
<b>UNV.UNVudm</b>	Universal Data Mover Manager
<b>UNV.UNVuds</b>	Universal Data Mover Server
<b>UNV.UNVuem</b>	Universal Event Monitor Manager

<b>UNV.UNVues</b>	Universal Event Monitor Server
<b>UNV.UNVusp</b>	Universal Connector
<b>UNV.UNVutil</b>	Universal Utilities

### Pre-Requisite to Installing Workload Automation 5 for HP-UX (PA-RISC Systems)

In order to install and run Workload Automation 5 for HP-UX (PA-RISC Systems), 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

### Installing Workload Automation 5 for HP-UX

Workload Automation 5 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.

### Starting the Installation Script

To start the installation script, **unvinst**, and install all of the HP-UX packages, issue the following command:

```
sh ./unvinst [--user username [--userdir dir]] [--group groupname] [--convert_opsagent option
[--opsdir dir]] [--ac_transports list]
            [--ac_core core] [--info]
```

See [HP-UX Install Script Parameters](#) and [Installation Script Example](#), below, for a description of the optional parameters that you can issue with **unvinst** and an example of **unvinst** with these parameters.

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.

The Universal Broker daemon will be installed and run as the **username** and **groupname** specified with the [installation script parameters](#), below.



#### Note

With the Solaris, HP-UX, 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.

### HP-UX Install Script Parameters

The following table describes the optional parameters that are available in the installation script, **unvinst**, when installing Workload Automation 5.

Parameter	Description	Default
-----------	-------------	---------



<pre>-u -user --user</pre>	<p>Normal UNIX <b>username</b> that is used to execute the Universal Broker daemon. The install grants this user account ownership of all installed files, with the exception of the Workload Automation server components (for example: <b>ucmsrv</b>, <b>udmsrv</b>, and <b>uemsrv</b>) which, due to security requirements, are owned by root and will have their "set user ID on execution" bit set.</p> <p>If the user account that you want to use already exists, specify that user account.</p> <p>If this user account does not exist, the install creates it. If <b>--user</b> is omitted from <b>unvinst</b>, the default user account (<b>ubroker</b>) is used.</p> <p>If you want to change the user account for an installed Workload Automation 5 for HP-UX system, you must perform a re-installation and use this parameter to change the user account.</p>	<b>ubroker</b>
<pre>-userdir --userdir</pre>	<p>Home directory for the created user account specified by <b>--user</b>.</p> <p>If this directory does not exist, it is created when the specified user is created. If <b>--userdir</b> is omitted from <b>unvinst</b>, the default is used.</p>	/home/<username>
<pre>-g -group --group</pre>	<p>Normal UNIX <b>groupname</b>; the Universal Broker daemon will run as this specified group. All installed files will be assigned to this group.</p> <p>If the group that you want to use already exists, specify that group.</p> <p>If this group does not exist, the install creates it. If <b>--group</b> is omitted from <b>unvinst</b>, the default group (<b>ubroker</b>) is used.</p>	<b>ubroker</b>
<pre>-c -convert --convert_opsagent</pre>	<p>Specification (<b>yes</b> or <b>no</b>) that causes <b>unvinst</b> to execute <b>opsmerge.sh</b>, which resides in <b>/opt/universal/uagsrv/bin</b>.</p> <p><b>opsmerge.sh</b> performs the following tasks:</p> <ol style="list-style-type: none"> <li>1. Searches for an existing Opwise Agent install (1.5 or later) and converts configuration options stored in the <b>agent.props</b> file to corresponding options in <b>uags.conf</b>.</li> <li>2. Searches for an active Opwise Agent daemon process and attempts to stop it.</li> <li>3. Assigns the ID used by the Opwise Agent to UAG, by moving the <b>qname</b> file from the Opwise install directory to the <b>/var/opt/universal/uag/var</b> directory.</li> </ol>	<b>yes</b>
<pre>-d -opsdir --opsdir</pre>	<p>Identifies the primary install directory for the Opwise Agent when <b>--convert_opsagent</b> is specified.</p> <p>If <b>--opsdir</b> is omitted from <b>unvinst</b>, the default is used.</p>	/home/opswise
<pre>-t -transports --ac_transports</pre>	<p>Specifies a value, in the format <b>port@ipaddr[,port2@ipaddr2,...,portn@ipaddrn]</b>, for the port and network address of the Opwise Automation Center Transporter(s) used for network communication. The install uses this value to set the UAG <b>AUTOMATION_CENTER_TRANSPORTS</b> configuration option. The value specified in <b>--ac_transports</b> will override any currently configured options, including those imported from <b>agent.props</b> when <b>--convert_opsagent</b> is set to <b>yes</b>.</p> <p>If <b>--ac_transports</b> is omitted from <b>unvinst</b>, and no other value specifies the port and network address of the Transporter(s), the default is used.</p>	4803@127.0.0.1
<pre>-r -core --ac_core</pre>	<p>Specifies a value for the queue name of the Opwise Automation Center Message Hub. The install uses this value to set the UAG <b>AUTOMATION_CENTER_CORE</b> configuration option. The value specified in <b>--ac_core</b> will override any currently configured options, including those imported from <b>agent.props</b> when <b>--convert_opsagent</b> is set to <b>yes</b>.</p> <p>If <b>--ac_core</b> is omitted from <b>unvinst</b>, and no other value specifies the queue name of the Opwise Automation Center Message Hub, the default is used.</p>	HUB01

<pre>-? -h -help --info</pre>	Displays command line help.	n/a
-------------------------------	-----------------------------	-----

### Installation Script Example

The following example illustrates Workload Automation 5 for HP\_UX installed with the installation script, **unvinst**, and optional parameters.

```
sh ./unvinst --user user1 --userdir /homedir/user --group usergroup --convert_opsagent no --opsdir
/homedir/ops
--ac_transports 4803@127.0.0.1 --ac_core HUB01
```

### Listing Workload Automation 5 for HP-UX Information

On HP-UX, information on an installed product and sub-products is listed with the **swlist** command. The command must be executed with the superuser ID.

To list information for Workload Automation 5 for HP-UX, issue the following command:

```
swlist -l subproduct UNV
```

### Removing Workload Automation 5 for HP-UX

<b>Step 1</b>	Stop the <b>ubrokerd</b> daemon.
<b>Step 2</b>	<p>Using the superuser ID, remove all Workload Automation 5 for HP-UX sub-products by issuing the following command:</p> <pre>swremove UNV</pre> <p>System initialization files <code>/sbin/init.d/ubrokerd</code> and <code>{/sbin/rc3.d/s850ubrokerd</code> are removed.</p>
<b>Step 3</b>	<p>Delete the Agent user account (<b>ubroker</b>) and its home directory:</p> <pre>userdel -r ubroker</pre>
<b>Step 4</b>	<p>Delete the Agent group (<b>ubroker</b>):</p> <pre>groupdel ubroker</pre>

## Workload Automation 5 for Solaris Installation

- Installation Process
- Extracting the Workload Automation for Solaris Installation Files
  - Intel-Based Systems
  - SPARC-Based Systems
- Solaris Package
- Installing Workload Automation 5 for Solaris
  - Starting the Installation Script
  - Solaris Installation Script Parameters
  - Installation Script Example
- Listing Workload Automation 5 for Solaris Information
- Removing Workload Automation 5 for Solaris

### Installation Process

Installation of Workload Automation for Solaris is a three-step process:

<b>Step 1</b>	Download product distribution file (see <a href="#">Workload Automation 5 for UNIX - Distribution File</a> ).
<b>Step 2</b>	Extract the installation files from the distribution file.
<b>Step 3</b>	Install the extracted files.

### Extracting the Workload Automation for Solaris Installation Files

Stonebranch, Inc. provides a separate product distribution file for the following Solaris systems:

- Intel-based systems
- SPARC-based systems

Workload Automation 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 sb-5.1.0.0-solaris-10-x64.tar.Z | tar xvf -
```

This command assumes that:

- Name of the distribution file is **sb-5.1.0.0-solaris-10-x64.tar.Z**.
- File is located in the current working directory.

#### Distribution File

The following table identifies the files contained in the distribution file.

File	Description
<b>Readme.unv</b>	Summary of the installation procedure.
<b>unv-5.1.0.0-solaris-10-x64.pkg</b>	Solaris Intel-based file format package.
<b>unvinst</b>	Installation script.
<b>upimerge.sh</b>	Script that uses the Universal Installation Merge (UIM) module.



#### Note

If your Workload Automation 5 for Solaris (Intel-based systems) distribution file does not contain all of 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 sb-5.1.0.0-solaris-8-sparc.tar.Z | tar xvf -
```

This command assumes that:

- Name of the distribution file is **sb-5.1.0.0-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 [Distribution File Format](#) for distribution file naming conventions.)

### Distribution File

The following table identifies the files contained in the distribution file.

File	Description
<b>Readme.unv</b>	Summary of the installation procedure.
<b>unv-5.1.0.0-solaris-8-sparc.pkg</b>	Solaris SPARC-based file format package.
<b>unvinst</b>	Installation script.
<b>upimerge.sh</b>	Script that uses the Universal Installation Merge (UIM) module.



#### Note

If your Workload Automation 5 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

Workload Automation 5 for Solaris is packaged as a set of Solaris packages (extension **.pkg**).

The following table identifies the Workload Automation 5 for Solaris package names.

Package Name	Description
<b>UNVcom</b>	Workload Automation 5 common files, such as, message catalogs and translation tables.
<b>UNVctl</b>	Universal Control Manager
<b>UNVcts</b>	Universal Control Server
<b>UNVqry</b>	Universal Query
<b>UNVspl</b>	Universal Spool
<b>UNVuag</b>	Universal Automation Center Agent
<b>UNVubr</b>	Universal Broker
<b>UNVucm</b>	Universal Command Manager
<b>UNVucr</b>	Universal Certificate
<b>UNVucs</b>	Universal Command Server
<b>UNVudm</b>	Universal Data Mover Manager
<b>UNVuds</b>	Universal Data Mover Server
<b>UNVuem</b>	Universal Event Monitor Manager

<b>UNVues</b>	Universal Event Monitor Server
<b>UNVusp *</b>	Universal Connector
<b>UNVutil</b>	Universal Utilities

**Note \***

**UNVusp** is not provided with the Workload Automation 5 for Solaris *Intel-based* file format package.

## Installing Workload Automation 5 for Solaris

Workload Automation 5 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.

### Starting the Installation Script

To start the installation script, **unvinst**, and install all of the Solaris packages, issue the following command:

```
sh ./unvinst [--user username [--userdir dir]] [--group groupname] [--convert_opsagent option
[--opsdir dir]] [--ac_transports list]
                [--ac_core core] [--info]
```

See [Solaris Installation Script Parameters](#) and [Installation Script Example](#), below, for a description of the optional parameters that you can issue with **unvinst** and an example of **unvinst** with these parameters.

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.

The Universal Broker daemon will be installed and run as the **username** and **groupname** specified with the [installation script parameters](#), below.

**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.

### Solaris Installation Script Parameters

The following table describes the optional parameters that are available in the installation script, **unvinst**, when installing Workload Automation 5.

Parameter	Description	Default
-----------	-------------	---------

<pre>-u -user --user</pre>	<p>Normal UNIX <b>username</b> that is used to execute the Universal Broker daemon. The install grants this user account ownership of all installed files, with the exception of the Workload Automation server components (for example: <b>ucmsrv</b>, <b>udmsrv</b>, and <b>uemsrv</b>) which, due to security requirements, are owned by root and will have their "set user ID on execution" bit set.</p> <p>If the user account that you want to use already exists, specify that user account.</p> <p>If this user account does not exist, the install creates it. If <b>--user</b> is omitted from <b>unvinst</b>, the default user account (<b>ubroker</b>) is used.</p> <p>If you want to change the user account for an installed Workload Automation 5 for Solaris system, you must perform a re-installation and use this parameter to change the user account.</p>	<p><b>ubroker</b></p>
<pre>-userdir --userdir</pre>	<p>Home directory for the created user account specified by <b>--user</b>.</p> <p>If this directory does not exist, it is created when the specified user is created. If <b>--userdir</b> is omitted from <b>unvinst</b>, the default is used.</p>	<p><b>/export/home/&lt;username&gt;</b></p>
<pre>-g -group --group</pre>	<p>Normal UNIX <b>groupname</b>; the Universal Broker daemon will run as this specified group. All installed files will be assigned to this group.</p> <p>If the group that you want to use already exists, specify that group.</p> <p>If this group does not exist, the install creates it. If <b>--group</b> is omitted from <b>unvinst</b>, the default group (<b>ubroker</b>) is used.</p>	<p><b>ubroker</b></p>
<pre>-c -convert --convert_opsagent</pre>	<p>Specification (<b>yes</b> or <b>no</b>) that causes <b>unvinst</b> to execute <b>opsmerge.sh</b>, which resides in <b>/opt/universal/uagsrv/bin</b>.</p> <p><b>opsmerge.sh</b> performs the following tasks:</p> <ol style="list-style-type: none"> <li>1. Searches for an existing Opswise Agent install (1.5 or later) and converts configuration options stored in the <b>agent.props</b> file to corresponding options in <b>uags.conf</b>.</li> <li>2. Searches for an active Opswise Agent daemon process and attempts to stop it.</li> <li>3. Assigns the ID used by the Opswise Agent to UAG, by moving the <b>qname</b> file from the Opswise install directory to the <b>/var/opt/universal/uag/var</b> directory.</li> </ol>	<p><b>yes</b></p>
<pre>-d -opsdir --opsdir</pre>	<p>Identifies the primary install directory for the Opswise Agent when <b>--convert_opsagent</b> is specified.</p> <p>If <b>--opsdir</b> is omitted from <b>unvinst</b>, the default is used.</p>	<p><b>/export/home/opswise</b></p>
<pre>-t -transport --ac_transport</pre>	<p>Specifies a value, in the format <b>port@ipaddr[,port2@ipaddr2,...,portn@ipaddrn]</b>, for the port and network address of the Opswise Automation Center Transporter(s) used for network communication. The install uses this value to set the UAG <b>AUTOMATION_CENTER_TRANSPORTS</b> configuration option. The value specified in <b>--ac_transport</b> will override any currently configured options, including those imported from <b>agent.props</b> when <b>--convert_opsagent</b> is set to <b>yes</b>.</p> <p>If <b>--ac_transport</b> is omitted from <b>unvinst</b>, and no other value specifies the port and network address of the Transporter(s), the default is used.</p>	<p><b>4803@127.0.0.1</b></p>

<pre>-r -core --ac_core</pre>	<p>Specifies a value for the queue name of the Opswise Automation Center Message Hub. The install uses this value to set the UAG <code>AUTOMATION_CENTER_CORE</code> configuration option. The value specified in <code>--ac_core</code> will override any currently configured options, including those imported from <code>agent.props</code> when <code>--convert_opsagent</code> is set to <code>yes</code>.</p> <p>If <code>--ac_core</code> is omitted from <code>unvinst</code>, and no other value specifies the queue name of the Opswise Automation Center Message Hub, the default is used.</p>	<p><b>HUB01</b></p>
<pre>--? -h -help --info</pre>	<p>Displays command line help.</p>	<p>n/a</p>

### Installation Script Example

The following example illustrates Workload Automation 5 for Solaris installed with the installation script, `unvinst`, and optional parameters.

```
sh ./unvinst --user user1 --userdir /homedir/user --group usergroup --convert_opsagent no --opsdir /homedir/ops --ac_transports 4803@127.0.0.1 --ac_core HUB01
```

### Listing Workload Automation 5 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 Workload Automation 5 for Solaris, issue the following command:

```
pkginfo UNVuocr UNVusp UNVutl UNVues UNVuem UNVuds UNVudm UNVucs UNVuem UNVqry UNVctl UNVcts UNVuag UNVubr UNVspl UNVcom
```



#### Note

**UNVusp** is not provided with the Workload Automation 5 for Solaris Intel-based file format package.

### Removing Workload Automation 5 for Solaris

<p><b>Step 1</b></p>	<p>Stop the <code>ubrokerd</code> daemon.</p>
<p><b>Step 2</b></p>	<p>Using the superuser ID, remove all Workload Automation 5 for Solaris packages by issuing the following command:</p> <pre>pkgrm UNVuocr UNVusp UNVutl UNVues UNVuem UNVuds UNVudm UNVucs UNVuem UNVqry UNVctl UNVcts UNVuag UNVubr UNVspl UNVcom</pre> <p>System initialization files <code>/etc/init.d/ubrokerd</code> and <code>/etc/rc3.d/S85ubrokerd</code> are removed.</p> <p> <b>Note</b> <b>UNVusp</b> is not provided with the Workload Automation 5 for Solaris Intel-based file format package.</p>

<b>Step 3</b>	Delete the Agent user account (ubroker) and its home directory: <pre>userdel -r ubroker</pre>
<b>Step 4</b>	Delete the Agent group (ubroker): <pre>groupdel ubroker</pre>



## Workload Automation 5 for Linux Installation

- Installation Process
- Extracting the Workload Automation for Linux Installation Files
  - x86-Based Systems
  - x86\_64-Based Systems
  - IBM S/390 and zSeries Systems
  - Itanium Systems
- Installing Workload Automation 5 for Linux
  - Starting the Installation Script
  - Linux Installation Script Parameters
  - Installation Script Example
- Listing Workload Automation 5 for Linux Information
- Removing Workload Automation 5 for Linux
- Linux PAM Customization
  - PAM Configuration File
  - Examples of Customized PAM Configuration Files
- Configuring the Agent to Run a Task without a Password

### Installation Process

Installation of Workload Automation for Linux is a three-step process:

<b>Step 1</b>	Download product distribution file (see <a href="#">Workload Automation 5 for UNIX - Distribution File</a> ).
<b>Step 2</b>	Extract the installation files from the distribution file.
<b>Step 3</b>	Install the extracted files.

### Extracting the Workload Automation for Linux Installation Files

Stonebranch, Inc. provides separate product distribution files for the following Linux systems:

- x86-Based Systems
- x86\_64-Based Systems
- IBM S/390 and zSeries Systems
- Itanium Systems

Workload Automation for Linux product distribution files are in a compressed **tar** format.

#### **x86-Based Systems**

(The RPM package for x86-based systems contains the **i386** qualifier.)

To uncompress and extract the installation files from the product distribution file, issue the following command:

```
zcat sb-5.1.0.0-linux-2.4-i386.tar.Z | tar xvf -
```

This command assumes that the name of the distribution file is **sb-5.1.0.0-linux-2.4-i386.tar.Z**.

#### Distribution File

The following table identifies the files contained in the distribution file.

File	Description
<b>Readme.unv</b>	Summary of the installation procedure.
<b>unv-5.1.0.0-linux-2.4-i386.rpm</b>	Linux RPM file format package.
<b>unvinst</b>	Installation script.
<b>upimerge.sh</b>	Script that uses the Universal Installation Merge (UIM) module.

**Note**

If your Workload Automation 5 for Linux (x86-based systems) distribution file does not contain all of these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

**x86\_64-Based Systems**

(The RPM package for x86\_64-based systems contains the **x86\_64** qualifier.)

To uncompress and extract the installation files from the product distribution file, issue the following command:

```
zcat sb-5.1.0.0-linux-2.6-x86_64.tar.Z | tar xvf -
```

This command assumes that the name of the distribution file is **sb-5.1.0.0-linux-2.6-x86\_64.tar.Z**.

**Distribution File**

The following table identifies the files contained in the distribution file.

File	Description*
<b>Readme.unv</b>	Summary of the installation procedure.
<b>unv-5.1.0.0-linux-2.6-x86_64.rpm</b>	Linux RPM file format package.
<b>unvinst</b>	Installation script.
<b>upimerge.sh</b>	Script that uses the Universal Installation Merge (UIM) module.

**Note**

If your Workload Automation 5 for Linux (x86\_64-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 **s390x** qualifier.)

To uncompress and extract the installation files from the product distribution file, issue the following command:

```
zcat sb-5.1.0.1-linux-2.6-s390x.tar.Z | tar xvf -
```

This command assumes that the name of the distribution file is **sb-5.1.0.1-linux-2.6-s390x.tar.Z**.

**Distribution File**

The following table identifies the files contained in the distribution file.

File	Description
<b>Readme.unv</b>	Summary the installation procedure.
<b>unv-5.1.0.1-linux-2.6-s390x.rpm</b>	Linux RPM file format package.
<b>unvinst</b>	Installation script.
<b>upimerge.sh</b>	Script that uses the Universal Installation Merge (UIM) module.

**Note**

If your Workload Automation 5 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 Itanium systems contains the **ia64** qualifier.)

To uncompress and extract the installation files from the product distribution file, issue the following command:

```
zcat sb-5.1.0.0-linux-2.6-ia64.tar.Z | tar xvf -
```

This command assumes that the name of the distribution file is **sb-5.1.0.0-linux-2.6-ia64.tar.Z**.

**Distribution File**

The following table identifies the files contained in the distribution file.

File	Description
<b>Readme.unv</b>	Summary of the installation procedure.
<b>unv-5.1.0.0-linux-2.6-ia64.rpm</b>	Linux RPM file format package.
<b>unvinst</b>	Installation script.
<b>upimerge.sh</b>	Script that uses the Universal Installation Merge (UIM) module.

**Note**

If your Workload Automation 5 for Linux (Itanium systems) distribution file does not contain these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

**Installing Workload Automation 5 for Linux**

Workload Automation 5 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.

**Starting the Installation Script**

To start the installation script, **unvinst**, issue the following command:

```
sh ./unvinst [--user username [--userdir dir]] [--group groupname] [--convert_opsagent option
[--opsdir dir]] [--ac_transports list]
            [--ac_core core] [--info]
```

See [Linux Install Script Parameters](#) and [Installation Script Example](#), below, for a description of the optional parameters that you can issue with **unvinst** and an example of **unvinst** with these parameters.

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**.

The Universal Broker daemon will be installed and run as the **username** and **groupname** specified with the [installation script parameters](#), below.

**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.

**Linux Installation Script Parameters**

The following table describes the optional parameters that are available in the UNIX install script (`unvinst`) when installing Workload Automation 5.

Parameter	Description	Default
<pre>-u -user --user</pre>	<p>Normal UNIX <b>username</b> that is used to execute the Universal Broker daemon. The install grants this user account ownership of all installed files, with the exception of the Workload Automation server components (for example: <b>ucmsrv</b>, <b>udmsrv</b>, and <b>uemsrv</b>) which, due to security requirements, are owned by root and will have their "set user ID on execution" bit set.</p> <p>If the user account that you want to use already exists, specify that user account.</p> <p>If this user account does not exist, the install creates it. If <b>--user</b> is omitted from <code>unvinst</code>, the default user account (<b>ubroker</b>) is used.</p> <p>If you want to change the user account for an installed Workload Automation 5 for Linux system, you must perform a re-installation and use this parameter to change the user account.</p>	<b>ubroker</b>
<pre>-userdir --userdir</pre>	<p>Home directory for the created user account specified by <b>--user</b>.</p> <p>If this directory does not exist, it is created when the specified user is created. If <b>--userdir</b> is omitted from <code>unvinst</code>, the default is used.</p>	<code>/home/&lt;username&gt;</code>
<pre>-g -group --group</pre>	<p>Normal UNIX <b>groupname</b>; the Universal Broker daemon will run as this specified group. All installed files will be assigned to this group.</p> <p>If the group that you want to use already exists, specify that group.</p> <p>If this group does not exist, the install creates it. If <b>--group</b> is omitted from <code>unvinst</code>, the default group (<b>ubroker</b>) is used.</p>	<b>ubroker</b>
<pre>-c -convert --convert_opsagent</pre>	<p>Specification (<b>yes</b> or <b>no</b>) that causes <code>unvinst</code> to execute <code>opsmerge.sh</code>, which resides in <code>/opt/universal/uagsrv/bin</code>.</p> <p><code>opsmerge.sh</code> performs the following tasks:</p> <ol style="list-style-type: none"> <li>1. Searches for an existing Opswise Agent install (1.5 or later) and converts configuration options stored in the <code>agent.props</code> file to corresponding options in <code>uags.conf</code>.</li> <li>2. Searches for an active Opswise Agent daemon process and attempts to stop it.</li> <li>3. Assigns the ID used by the Opswise Agent to UAG, by moving the <code>qname</code> file from the Opswise install directory to the <code>/var/opt/universal/uag/var</code> directory.</li> </ol>	<b>yes</b>
<pre>-d -opsdir --opsdir</pre>	<p>Identifies the primary install directory for the Opswise Agent when <b>--convert_opsagent</b> is specified.</p> <p>If <b>--opsdir</b> is omitted from <code>unvinst</code>, the default is used.</p>	<code>/home/opswise</code>

<pre>-t -transport --ac_transports</pre>	<p>Specifies a value, in the format <code>port@ipaddr[,port2@ipaddr2,...,portn@ipaddrn]</code>, for the port and network address of the Opwise Automation Center Transporter(s) used for network communication. The install uses this value to set the UAG <code>AUTOMATION_CENTER_TRANSPORTS</code> configuration option. The value specified in <code>--ac_transports</code> will override any currently configured options, including those imported from <code>agent.props</code> when <code>--convert_opsagent</code> is set to <b>yes</b>.</p> <p>If <code>--ac_transports</code> is omitted from <code>unvinst</code>, and no other value specifies the port and network address of the Transporter(s), the default is used.</p>	<p>4803@127.0.0.1</p>
<pre>-r -core --ac_core</pre>	<p>Specifies a value for the queue name of the Opwise Automation Center Message Hub. The install uses this value to set the UAG <code>AUTOMATION_CENTER_CORE</code> configuration option. The value specified in <code>--ac_core</code> will override any currently configured options, including those imported from <code>agent.props</code> when <code>--convert_opsagent</code> is set to <b>yes</b>.</p> <p>If <code>--ac_core</code> is omitted from <code>unvinst</code>, and no other value specifies the queue name of the Opwise Automation Center Message Hub, the default is used.</p>	<p>HUB01</p>
<pre>-? -h -help --info</pre>	<p>Displays command line help.</p>	<p>n/a</p>

### Installation Script Example

The following example illustrates Workload Automation 5 for Linux installed with the installation script, `unvinst`, and optional parameters.

```
sh ./unvinst --user user1 --userdir /homedir/user --group usergroup --convert_opsagent no --opsdir /homedir/ops --ac_transports 4803@127.0.0.1 --ac_core HUB01
```

### Listing Workload Automation 5 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 Workload Automation 5 for Linux, issue the following command:

```
rpm -qi unv
```

### Removing Workload Automation 5 for Linux

<p><b>Step 1</b></p>	<p>Stop the <code>ubrokerd</code> daemon.</p>
<p><b>Step 2</b></p>	<p>Using the superuser ID, remove all Workload Automation 5 for Linux packages by issuing the following command:</p> <pre>rpm -e unv</pre>
<p><b>Step 3</b></p>	<p>Delete the Agent user account (<code>ubroker</code>) and its home directory:</p> <pre>userdel -r ubroker</pre>

**Step 4** Delete the Agent group (ubroker):

```
groupdel ubroker
```

## Linux PAM Customization

Linux installations utilize Pluggable Authentication Modules (PAM) for user authentication. Many of the Workload Automation 5 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 Workload Automation 5 to be copied to directory **/etc/pam.d**:

```
cp /opt/universal/ucmdsrv/samp/ucmd.pam /etc/pam.d/ucmd
```

### PAM Configuration File

All Workload Automation 5 components utilize the same PAM configuration file.

#### 32-Bit Linux Systems

For 32-bit Linux systems (X\_86-Based systems), 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

#### 64-Bit Linux Systems

For 64-bit Linux systems (x86\_64-Based systems; Itanium systems; S/390 and z/Series systems) its contents are:

auth	required	/lib64/security/pam_pwdb.so shadow nullok
auth	required	/lib64/security/pam_nologin.so
account	required	/lib64/security/pam_pwdb.so

Your Administrator must modify this sample PAM file to meet your local configuration.

### Examples of Customized PAM Configuration Files

Workload Automation 5 for Redhat systems 5.0 and greater

auth	include	system-auth
auth	required	pam_nologin.so
account	include	system-auth

Workload Automation 5 for SUSE-based systems 10.0 and greater

auth	required	pam_unix2.so nullok
auth	required	pam_nologin.so

account	include	common-account
---------	---------	----------------

Alternative Workload Automation 5 for SUSE 10.1

auth	required	pam_unix2.so nullok
auth	required	pam_nologin.so
account	include	common-account

### Configuring the Agent to Run a Task without a Password

Prior to release 5.1, Opswise Automation Center used the **ops\_suexec.nopass file**, which listed all trusted users.

In release 5.1, this file no longer is used. To apply your desired security configuration, access the UAG **SECURITY** configuration option in the **uags.conf** configuration file:

- If you do not want security, set the value to **none**.
- If you want security, set the value to **pam** and update the following files:
  - Check the **/etc/pam.d/ucmd** configuration file to ensure that it contains the minimum PAM security settings (see [Examples of Customized PAM Configuration Files](#), above).
  - Add the following to **/etc/universal/uac.conf** for each user: **uag\_work\_request [username],allow,noauth**  
Also, verify that the user has a **/home** directory defined.

## Workload Automation 5 for UNIX Installation (Generic)

- Workload Automation for UNIX Installation (Generic)
- Installation Process
- Extracting the Workload Automation for UNIX Installation Files
  - MP-RAS
  - Tru64
- Distribution File
- Installing Workload Automation 5 for UNIX
  - Starting the Installation Script
  - UNIX (Generic) Installation Script Parameters
  - Installation Script Example
  - Installation Environment Variables
- Removing Workload Automation 5 for UNIX

### Workload Automation for UNIX Installation (Generic)

Installation of Workload Automation for UNIX on operating systems other than AIX, HP-UX, Solaris, and Linux are performed with the generic Workload Automation for UNIX installation script. Currently, Workload Automation for UNIX installs on two such operating systems:

- MP-RAS
- Tru64

### Installation Process

Installation of these Workload Automation for UNIX is a three-step process:

<b>Step 1</b>	Download product distribution file (see <a href="#">Workload Automation 5 for UNIX - Distribution File</a> ).
<b>Step 2</b>	Extract the installation files from the distribution file.
<b>Step 3</b>	Install the extracted files.

### Extracting the Workload Automation for UNIX Installation Files

The Workload Automation 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:

#### MP-RAS

```
zcat sb-5.1.0.0-mpras-3.02-i386.tar.Z | tar xvf -
```

(This command assumes that the name of the distribution file is **sb-5.1.0.0-mpras-3.02-i386.tar.Z**.)

#### Tru64

```
zcat sb-5.1.0.0-tru64-5.1-alpha.tar.Z | tar xvf -
```

(This command assumes that the name of the distribution file is **sb-5.1.0.0-tru64-5.1-alpha.tar.Z**.)

### Distribution File

The following table identifies the files contained in each of these distribution files.

File	Description
<b>Readme.unv</b>	Summary of the installation procedure.
<b>sb.tar</b>	<b>tar</b> file format package.



<code>unvinst</code>	Installation script.
<code>upimerge.sh</code>	Script that uses the Universal Installation Merge (UIM) module.

**Note**

If your Workload Automation 5 for UNIX distribution file does not contain all of these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

## Installing Workload Automation 5 for UNIX

Workload Automation 5 for UNIX (generic) is installed with the 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.

### Starting the Installation Script

To start the installation script, `unvinst`, issue the following command:

```
sh ./unvinst [--user username [--userdir dir]] [--group groupname]
```

See [UNIX \(Generic\) Installation Script Parameters](#) and [Installation Script Example](#), below, for a description of the optional parameters that you can issue with `unvinst` and an example of `unvinst` with these parameters.

When you issue the command, an installation log file is written to `./install.log`. The Universal Broker daemon will be installed and run as the `username` and `groupname` specified with the [installation script parameters](#), below.

**Note**

With installation of Workload Automation 5 for UNIX using the generic installation script, the previous installation directories — default or user-specified — are not removed.

### UNIX (Generic) Installation Script Parameters

The following table describes the optional parameters that are available in the UNIX installation script, `unvinst`, when installing Workload Automation 5.

Parameter	Description	Default
<pre>-u -user --user</pre>	<p>Normal UNIX <b>username</b> that is used to execute the Universal Broker daemon. The install grants this user account ownership of all installed files, with the exception of the Workload Automation server components (for example: <code>ucmsrv</code>, <code>udmsrv</code>, and <code>uemsrv</code>) which, due to security requirements, are owned by root and will have their "set user ID on execution" bit set.</p> <p>If the user account that you want to use already exists, specify that user account.</p> <p>If this user account does not exist, the install creates it. If <code>--user</code> is omitted from <code>unvinst</code>, the default user account (<b>ubroker</b>) is used.</p> <p>If you want to change the user account for an installed Workload Automation 5 for UNIX (Generic) system, you must perform a re-installation and use this parameter to change the user account.</p>	<b>ubroker</b>
<pre>--userdir --userdir</pre>	<p>Home directory for the created user account specified by <code>--user</code>.</p> <p>If this directory does not exist, it is created when the specified user is created. If <code>--userdir</code> is omitted from <code>unvinst</code>, the default is used.</p>	<code>/home/&lt;username&gt;</code>

<pre>-g -group --group</pre>	<p>Normal UNIX <b>groupname</b>; the Universal Broker daemon will run as this specified group. All installed files will be assigned to this group.</p> <p>If the group that you want to use already exists, specify that group.</p> <p>If this group does not exist, the install creates it. If <b>--group</b> is omitted from <b>unvinst</b>, the default group (<b>ubroker</b>) is used.</p>	<p><b>ubroker</b></p>
<pre>-? -h -help --info</pre>	<p>Displays command line help.</p>	<p>n/a</p>

### Installation Script Example

The following example illustrates Workload Automation 5 for UNIX (generic) installed with the installation script, **unvinst**, and optional parameters.

```
sh ./unvinst --user user1 --userdir /homedir/user --group usergroup
```

### Installation Environment Variables

Environment variables can be defined to set default values for certain installation options.

The following table identifies the environment variables used and their install-script defaults.

Variable Name	Default Value	Notes
UNV_INSTLOG	<b>./install.log</b>	Installation log file.
UNV_CFGDIR	<b>/etc/universal</b>	Default configuration directory.
UNV_START_DAEMON	<b>Y</b>	Default response to start the Universal Broker daemon or not.
UNV_USRBIN	<b>/opt/universal/bin</b>	Default directory in which symbolic links are created to user programs.

### Removing Workload Automation 5 for UNIX

<b>Step 1</b>	Stop the <b>ubrokerd</b> daemon.
<b>Step 2</b>	<p>Remove Workload Automation 5 for UNIX (generic) by deleting the relevant directories:</p> <ul style="list-style-type: none"> <li>• <b>/opt/universal</b></li> <li>• <b>/var/opt/universal</b></li> <li>• <b>/etc/universal</b></li> </ul>
<b>Step 3</b>	<p>Delete the Agent user account (<b>ubroker</b>) and its home directory:</p> <pre>userdel -r ubroker</pre>
<b>Step 4</b>	<p>Delete the Agent group (<b>ubroker</b>):</p> <pre>groupdel ubroker</pre>

## Migrating an Opswise Agent to UAG for Workload Automation 5 for UNIX

- [Overview](#)
- [UAG Crossgrade Script](#)
  - [Crossgrade Script Actions](#)
  - [Crossgrade Script Command Options](#)

### Overview

The Universal Automation Center Agent (UAG) provided by the Workload Automation 5 installation replaces all previous versions (1.7 and earlier) of the Opswise Automation Center Agent.

When instructed to do so, the Workload Automation 5 installation will convert an Opswise Agent's existing settings to the UAG configuration file (**uags.conf**). The install also can disable the existing Opswise Agent in favor of the UAG server.

This migration commonly is referred to throughout this page as a UAG crossgrade, or simply, crossgrade.



#### Note

The installation procedures for each version of UNIX may contain information pertinent to this upgrade (see [Workload Automation 5 for UNIX - Installation Procedures Overview](#)).

### UAG Crossgrade Script

The UAG crossgrade is performed by a script named **opsmerge.sh**, which is installed to **/opt/universal/uagsrv/bin**.

The command line syntax for executing a UAG crossgrade using the **unvinst** script is:

```
sh ./unvinst --convert_opsagent=value [--opsdir=opsagentdir]
```

The **opsmerge.sh** script is executed from **unvinst** only when the **--convert\_opsagent** value is **yes**. Optionally, **--opsdir** can be specified to set the install location of the Opswise agent. **--opsdir** is ignored if **--convert\_opsagent** is not specified.

For example:

```
sh ./unvinst --convert_opsagent=yes [--opsdir=opsagentdir]
```

### Crossgrade Script Actions

The script performs the following actions:

- Verifies the user running the script is root
- Stops an active Opswise Automation Center agent daemon (it will **not** remove it from **/etc/init.d** or platform-specific equivalent; this must be done after the Workload Automation install).  
If the agent cannot be stopped gracefully, the script will wait for 60 secs and then issue a SIGKILL.
- Copies an existing **./cache/qname** (pre-1.6) or **./var/qname** (1.6 and later) file from the Opswise agent install directory to **/var/opt/universal/uag/var**. The target directory is created if it does not already exist.
- Renames the old Opswise agent **qname** file to **qname.yymmdd-hhmm**.
- Checks the existing **./etc/agent.props** file for the properties listed below. Each property is mapped to the **uags.conf** name listed beside it.

<b>agent.props Property Name</b>	<b>uags.conf Configuration Keyword</b>
config.loglvl	loglvl
config.txtdebug	txtdebug
config.clusters	agent_clusters
network.core	automation_center_core
network.name	netname

network.transports	automation_center_transports
--------------------	------------------------------

The script can handle any combination of whitespace in agent.props before/after property names and before/after values. If the value itself contains a space, it is enclosed in quotes inside the UAG configuration file.

The `opsmerge.sh` script maps the last uncommented instance of an agent property to its corresponding UAG configuration keyword and writes the converted property to a file named `uags.conf.yymmdd-hhmm`. This file becomes input to the Universal Products Install Merge utility (`upimerge`), which eventually combines the contents of the converted file with the installed UAG configuration file, `/etc/universal/uags.conf`.

If the `opsmerge.sh` script fails, it will NOT stop the install, though it may set the `unvinst` return code.

### ***Crossgrade Script Command Options***

The `opsmerge.sh` script accepts the following command line options:

Parameter	Description	Default
opmdir	UAG install directory.	<code>/home/opswise</code>  ( <code>/export/home/opswise</code> on Solaris systems.)
arcdir	Location of the <code>uags.conf.yymmdd-hhmm</code> file, used as input to the UPIMERGE utility.	<code>/etc/universal/.archive</code>
qnamedir	Location where the <code>qname</code> file that UAG will use is stored.	<code>/var/opt/universal/uag/var</code>

## Workload Automation 5 for UNIX - Customization

- Overview
- Universal Broker Customization
  - Universal Broker Configuration
  - Universal Broker System Initialization
- Universal Automation Center Agent Customization
  - Universal Automation Center Agent Configuration
- Universal Command Manager Customization
  - Universal Command Manager Configuration
- Universal Command Server Customization
  - Universal Command Server Configuration
- Universal Connector Customization
  - Universal Connector Configuration
  - Universal Connector SAP RFC Configuration
- Universal Control Manager Customization
  - Universal Control Manager Configuration
- Universal Control Server Customization
  - Universal Control Server Configuration
- Universal Data Mover Manager Customization
  - Universal Data Mover Manager Configuration
- Universal Data Mover Server Customization
  - Universal Data Mover Server Configuration
- Universal Event Monitor Manager Customization
  - Universal Event Monitor Manager Configuration
- Universal Event Monitor Server Customization
  - Universal Event Monitor Server Configuration
- Universal Query Customization
  - Universal Query Configuration

### Overview

This page provides the following information for the customization of Workload Automation components:

- Configuration
- System initialization (Universal Broker)

(For information on applying product licenses to installed Workload Automation 5 for UNIX components, see [UNIX Installation - Licensing](#).)

### Universal Broker Customization

#### Universal Broker Configuration

Configuration options for Universal Broker are stored in configuration file, **ubroker.conf**, in directory **/etc/universal** by default.

See the [Universal Broker 5.1.0 Reference Guide](#) for details on configuring Universal Broker.

#### Universal Broker System Initialization

A Broker daemon start-up script is provided in file **/opt/universal/ubroker/ubrokerd**. A single command line argument — either **start**, **stop**, **status**, or **restart** — instructs the script on the action to take.

See the [Universal Broker 5.1.0 Reference 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-5.1.0/ubrokerd** 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.

## Universal Automation Center Agent Customization

### Universal Automation Center Agent Configuration

Configuration options for Universal Automation Center Agent (UAG) are stored in configuration file, **uags.conf**, in directory `/etc/universal` by default. See the [Universal Automation Center Agent 5.1.0 Reference Guide](#) for details on configuring UAG.

UAG runs as a component managed by Universal Broker. UAG provides a component definition file, **uag**, that Universal Broker uses to start it and establish its runtime environment. **uag** is located in directory `/etc/universal/comp` by default.

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.

UAG uses the Universal Access Control List (UACL) configuration file as a level of product security. How UAG utilizes the UACL file is described in the [Indesca 5.1.0 User Guide](#).

## Universal Command Manager Customization

### Universal Command Manager Configuration

Configuration options for Universal Command Manager are stored in configuration file, **ucmd.conf**, in directory `/etc/universal` by default.

See the [Universal Command 5.1.0 Reference Guide](#) for details on configuring Universal Command 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.

## Universal Command Server Customization

### Universal Command Server Configuration

Configuration options for Universal Command Server are stored in configuration file, **ucmds.conf**, in directory `/etc/universal` by default.

See the [Universal Command 5.1.0 Reference 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/universal/comp` by default.

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.

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 [Indesca 5.1.0 User Guide](#).

## Universal Connector Customization

### Universal Connector Configuration

Configuration options for Universal Connector are stored in configuration file **usap.conf**, in directory `/etc/universal`, by default.

See the [Universal Connector 5.1.0 Reference Guide](#) for details on configuring Universal Connector.

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.

### Universal Connector 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.

## Universal Control Manager Customization

### Universal Control Manager Configuration

Configuration options for Universal Control Manager are stored in configuration file, **uctl.conf**, in directory **/etc/universal** by default.

See the [Workload Automation Utilities 5.1.0 Reference Guide](#) for details on configuring Universal Control 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.

## Universal Control Server Customization

### Universal Control Server Configuration

Configuration options for Universal Control Server are stored in configuration file, **uctls.conf**, in directory **/etc/universal** by default.

See the [Workload Automation Utilities 5.1.0 Reference 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, **uctl**, that Universal Broker uses to start the Server and establish its runtime environment. **uctl** is located in directory **/etc/universal/comp** by default

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.

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 [Indesca 5.1.0 User Guide](#).

## Universal Data Mover Manager Customization

### Universal Data Mover Manager Configuration

Configuration options for UDM Manager are stored in configuration file, **udm.conf**, in directory **/etc/universal** by default.

See the [Universal Data Mover 5.1.0 Reference Guide](#) for details on configuring UDM 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.

## Universal Data Mover Server Customization

### Universal Data Mover Server Configuration

Configuration options for UDM Server are stored in configuration file, **udms.conf**, in directory **/etc/universal** by default.

See the [Universal Data Mover 5.1.0 Reference Guide](#) for details on configuring Universal Command Server.

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/universal/comp** by default

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.

UDM Server uses the Universal Access Control List (UACL) configuration file as a level of product security. How Universal Data Mover Server utilizes the UACL file is described in the [Infitran 5.1.0 User Guide](#).

## Universal Event Monitor Manager Customization

### Universal Event Monitor Manager Configuration

Configuration options for UEM Manager are stored in configuration file **uem.conf**, in directory **/etc/universal** by default.

See the [Universal Event Monitor 5.1.0 Reference 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.

## Universal Event Monitor Server Customization

### Universal Event Monitor Server Configuration

Configuration options for UEM Server are stored in configuration file, `uems.conf`, in directory `/etc/universal` by default.

See the [Universal Event Monitor 5.1.0 Reference Guide](#) for details on configuring UEM Server.

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. How Universal Event Monitor Server utilizes the UACL file is described in the [Indesca 5.1.0 User Guide](#).

## Universal Query Customization

### Universal Query Configuration

Configuration options for Universal Query are stored in configuration file, `uquery.conf`, in directory `/etc/universal` by default.

See the [Workload Automation Utilities 5.1.0 Reference 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.



## Workload Automation 5 for UNIX - File Inventory Lists

- Workload Automation 5 for UNIX - File Inventory Lists
- Universal Automation Center Agent
- Universal Broker
- Universal Command Manager
- Universal Command Server
- Universal Connector
- Universal Control Manager
- Universal Control Server
- Universal Data Mover Manager
- Universal Data Mover Server
- Universal Event Monitor Manager
- Universal Event Monitor Server
- Universal Certificate
- Universal Query
- Universal Spool Utilities

### Workload Automation 5 for UNIX – File Inventory Lists

The Workload Automation installation for UNIX includes the files required for:

- Universal Automation Center Agent
- 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 page identifies the files installed with each of these Workload Automation 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 Automation Center Agent

File	Description
<b>/opt/universal/uagsrv/bin/uagsrv</b>	UAG component program.
<b>/opt/universal/uagsrv/samp</b>	UAG sample directory.
<b>/opt/universal/nls</b>	Code page files used for text translation between different operating systems and product message catalogs.
<b>/etc/universal/uags.conf</b>	UAG configuration file.
<b>/etc/universal/comp/uag</b>	UAG component definition file.
<b>/var/opt/universal/trace</b>	UAG trace file directory.

<code>/opt/universal/ubroker/tmpl/uagcfg</code>	Template file for the UAG configuration.
<code>/opt/universal/ubroker/tmpl/uagcmp</code>	Template file for the UAG component definition.

## Universal Broker

File	Description
<code>/opt/universal/ubroker/ubrinstd.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/tmpl</code>	XML template files used by I-Management Console for remotely configuring Workload Automation 5
<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.

## Universal Command Manager

File	Description
<code>/opt/universal/ucmdmgr/ucmcinst.src</code>	Manager installation source file (included in Generic UNIX packaging only).

<b>/opt/universal/ucmdmgr/bin/ucmd</b>	Universal Command Manager program.
<b>/opt/universal/ucmdmgr/bin/uencrypt</b>	Universal Encrypt program file.
<b>/opt/universal/ucmdmgr/samp</b>	Manager sample file directory.
<b>/opt/universal/bin/ucmd</b>	Symbolic link to the Manager program.
<b>/opt/universal/bin/uencrypt</b>	Symbolic link to the Universal Encrypt program.
<b>/opt/universal/nls</b>	Code page files used for text translation between different operating systems and product message catalogs.
<b>/etc/universal/ucmd.conf</b>	Manager configuration file.

## Universal Command Server

<b>File</b>	<b>Description</b>
<b>/opt/universal/ucmdsrv/ucmsinst.src</b>	Server installation source file (included in Generic UNIX packaging only).
<b>/opt/universal/ucmdsrv/bin/ucmsrv</b>	Server component program.
<b>/opt/universal/ucmdsrv/bin/ucopy</b>	Utility used for binary file copies. Similar to the UNIX cat command.
<b>/opt/universal/ucmdsrv/bin/umet</b>	Universal Message Translator program.
<b>/opt/universal/ucmdsrv/samp</b>	Server sample directory.
<b>/etc/universal/comp</b>	Universal Command Server component definition file.
<b>/opt/universal/bin/ucopy</b>	Symbolic link to the Universal Copy program.
<b>/opt/universal/bin/umet</b>	Symbolic link to the Universal Message Translator program.
<b>/opt/universal/nls</b>	Code page files used for text translation between different operating systems and product message catalogs.
<b>/var/opt/universal/trace</b>	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.

## 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.

## 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.

## Universal Control Server

File	Description
<code>/opt/universal/uctlsrv/uctsinst.src</code>	Server installation source file (include in Generic UNIX packaging only).

<b>/opt/universal/ucltsrv/bin/ucltsrv</b>	Server component program.
<b>/opt/universal/ucltsrv/samp</b>	Server sample file directory.
<b>/etc/universal/comp</b>	Universal Control Server component definition file.
<b>/opt/universal/nls</b>	Code page files used for text translation between different operating systems and product message catalogs.
<b>/var/opt/universal/trace</b>	Server trace file directory.
<b>/etc/universal/uclts.conf</b>	Server configuration file.

### Universal Data Mover Manager

File	Description
<b>/opt/universal/udmmgr/udmminst.src</b>	Manager installation source file (included in Generic UNIX packaging only).
<b>/opt/universal/udmmg/bin/udm</b>	Manager program.
<b>/opt/universal/udmmgr/samp</b>	Manager sample file directory.
<b>/opt/universal/bin/udm</b>	Symbolic link to the Manager program.
<b>/opt/universal/nls</b>	Code page files used for text translation between different operating systems and product message catalogs.
<b>/etc/universal/udm.conf</b>	Manager configuration file.

### Universal Data Mover Server

File	Description
<b>/opt/universal/udmsrv/udmsinst.src</b>	Server installation source file (include in Generic UNIX packaging only).
<b>/opt/universal/udmsrv/bin/udmsrv</b>	Server program.
<b>/opt/universal/udmsrv/samp</b>	Server sample file directory.
<b>/opt/universal/nls</b>	Code page files used for text translation between different operating systems

	and product message catalogs.
<b>/etc/universal/udms.conf</b>	Server configuration file.
<b>/etc/universal/comp</b>	Universal Data Mover Server component definition file.

## Universal Event Monitor Manager

File	Description
<b>/opt/universal/uemmgr/uemminst.src</b>	Manager installation source file (included in Generic UNIX packaging only).
<b>/opt/universal/uemmgr/bin/uem</b>	Manager program.
<b>/opt/universal/uemmgr/samp</b>	Manager sample file directory.
<b>/opt/universal/bin/uem</b>	Symbolic link to the Manager program.
<b>/opt/universal/nls</b>	Code page files used for text translation between different operating systems and product message catalogs.
<b>/etc/universal/uem.conf</b>	Manager configuration file.

## Universal Event Monitor Server

File	Description
<b>/opt/universal/uemsvr/uemsinst.src</b>	Server installation source file (included in Generic UNIX packaging only).
<b>/opt/universal/uemsvr/bin/uemsvr</b>	Server program.
<b>/opt/universal/uemsvr/bin/uemload</b>	UEM database load utility.
<b>/opt/universal/uemsvr/samp</b>	Server sample file directory.
<b>/opt/universal/bin/uemload</b>	Symbolic link to the UEM database load utility.
<b>/opt/universal/nls</b>	Code page files used for text translation between different operating systems and product message catalogs.
<b>/etc/universal/uems.conf</b>	Server configuration file.

<b>/etc/universal/comp/uems</b>	Event-driven Universal Event Monitor Server component definition file.
<b>/etc/universal/comp/uemd</b>	Demand-driven Universal Event Monitor Server component definition file.

## Universal Certificate

File	Description
<b>/opt/universal/ucert/bin/ucert</b>	UCERT program

## Universal Query

File	Description
<b>/opt/universal/uquery/uqryinst.src</b>	Query installation source file (included in Generic UNIX packaging only).
<b>/opt/universal/uquery/bin/uquery</b>	Universal Query program.
<b>/opt/universal/uquery/samp</b>	Query sample file directory.
<b>/opt/universal/bin/uquery</b>	Symbolic link to the Query program.
<b>/opt/universal/nls</b>	Code page files used for text translation between different operating systems and product message catalogs.
<b>/etc/universal/uquery.conf</b>	Query configuration file.

## Universal Spool Utilities

File	Description
<b>/opt/universal/uspool/bin/uslist</b>	Used to list the contents of Universal Spool files.
<b>/opt/universal/uspool/bin/uslrm</b>	Used to remove records from Universal Spool files.
<b>/opt/universal/uspool/bin/udb_archive</b> <b>/opt/universal/uspool/bin/udb_checkpoint</b> <b>/opt/universal/uspool/bin/udb_deadlock</b> <b>/opt/universal/uspool/bin/udb_dump</b> <b>/opt/universal/uspool/bin/udb_load</b>	Miscellaneous spool file utilities. Should be used only for debugging purposes, and only at the request of Stonebranch, Inc. Customer Support.

<code>/opt/universal/uspool/bin/udb_printlog</code>	
<code>/opt/universal/uspool/bin/udb_recover</code>	
<code>/opt/universal/uspool/bin/udb_stat</code>	
<code>/opt/universal/uspool/bin/udb_upgrade</code>	
<code>/opt/universal/uspool/bin/udb_verify</code>	



## **Workload Automation 5 for SOA for UNIX**

Error formatting macro: redirect: java.lang.NullPointerException

## Workload Automation 5 for SOA for UNIX - Installation Overview

- Installation Package
  - Package Components
  - Component Compatibility
- System Requirements
  - UNIX Versions
- Platform Requirements
- Licensing

### Installation Package

#### Package Components

The Workload Automation 5 for SOA 5.1.0 for UNIX package includes the following product components:

- Universal Application Container Server
- Universal Application Container
- Universal Application Interface

#### Component Compatibility

The following table identifies the compatibility of Workload Automation 5 for SOA for UNIX 5.1.0 with previous component / product versions.

Component	Compatibility
Workload Automation 5 for SOA 5.1.0	Universal Command Manager 4.3.0, 4.2.0, 4.1.0, and 3.2.0.

The component references pertain to all supported platforms for that version.

### System Requirements

#### UNIX Versions

To install Workload Automation 5 for SOA for UNIX, you must have one of the following versions of UNIX:

- Supported UNIX operating system:
  - AIX 5.3 TL9 and above.
  - Linux 2.4 kernel and greater:
    - RedHat Package Manager (RPM).
    - Intel (x86) Compatible Systems.
- 512MB RAM minimum, 1 GB or more preferred.
- 150 MB free disk space.
- TCP/IP socket implementation.
- Superuser (root) access.
- Bourne shell or compatible.
- Workload Automation 5.1.0.0 or later (32-bit packages only).

#### Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opswise Automation Center 5.1.1](#) and [Indesca-Infitrans 5.1.0](#) page to make sure that your platform is supported before performing an installation.

#### Licensing

Workload Automation 5 for SOA allows operation by connector and endpoint count, based on the license information it receives from the UAC Server component (see [UNIX Installation - Licensing](#) for details).

For licensing information, see [UNIX Installation - Licensing](#).

## Workload Automation 5 for SOA for UNIX - Deployment Options

- Deployment Options
- Single-Server Deployment (SSD)
  - SSD Deployment Flow of Events
- Distributed Server Deployment (DSD)
  - DSD Deployment Flow of Events

### Deployment Options

Deployment of Workload Automation 5 for SOA has been designed to be flexible in order to fit the needs of your Enterprise IT.

There are two main deployment options:

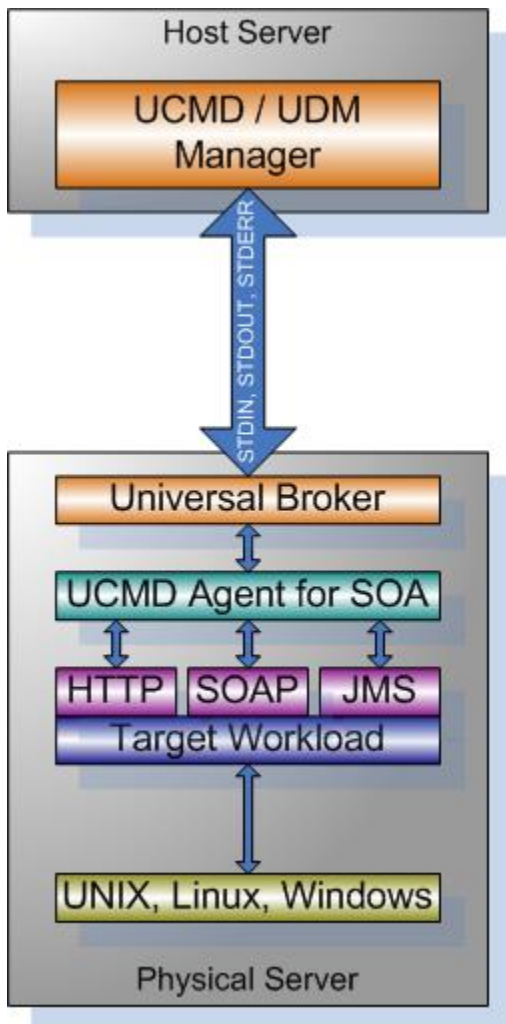
1. Single-Server Deployment (SSD)
2. Distributed Server Deployment (DSD)

**SSD is the default deployment.**

### Single-Server Deployment (SSD)

Single-Server Deployment (SSD) is where all components, with the exception of the Workload Automation 5 managers, are deployed to the same physical server. This includes Workload Automation 5 for SOA, Universal Broker and associated components, and your target workload(s).

The following figure illustrates the Single Server Deployment (SSD).



## SSD Deployment Flow of Events

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. 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 Workload Automation 5 for SOA.
3. Workload Automation 5 for SOA will execute the workload deployed on the same physical server and return any messages or data back to the Universal Command Manager.

## Distributed Server Deployment (DSD)

Distributed Server Deployment (DSD) is where:

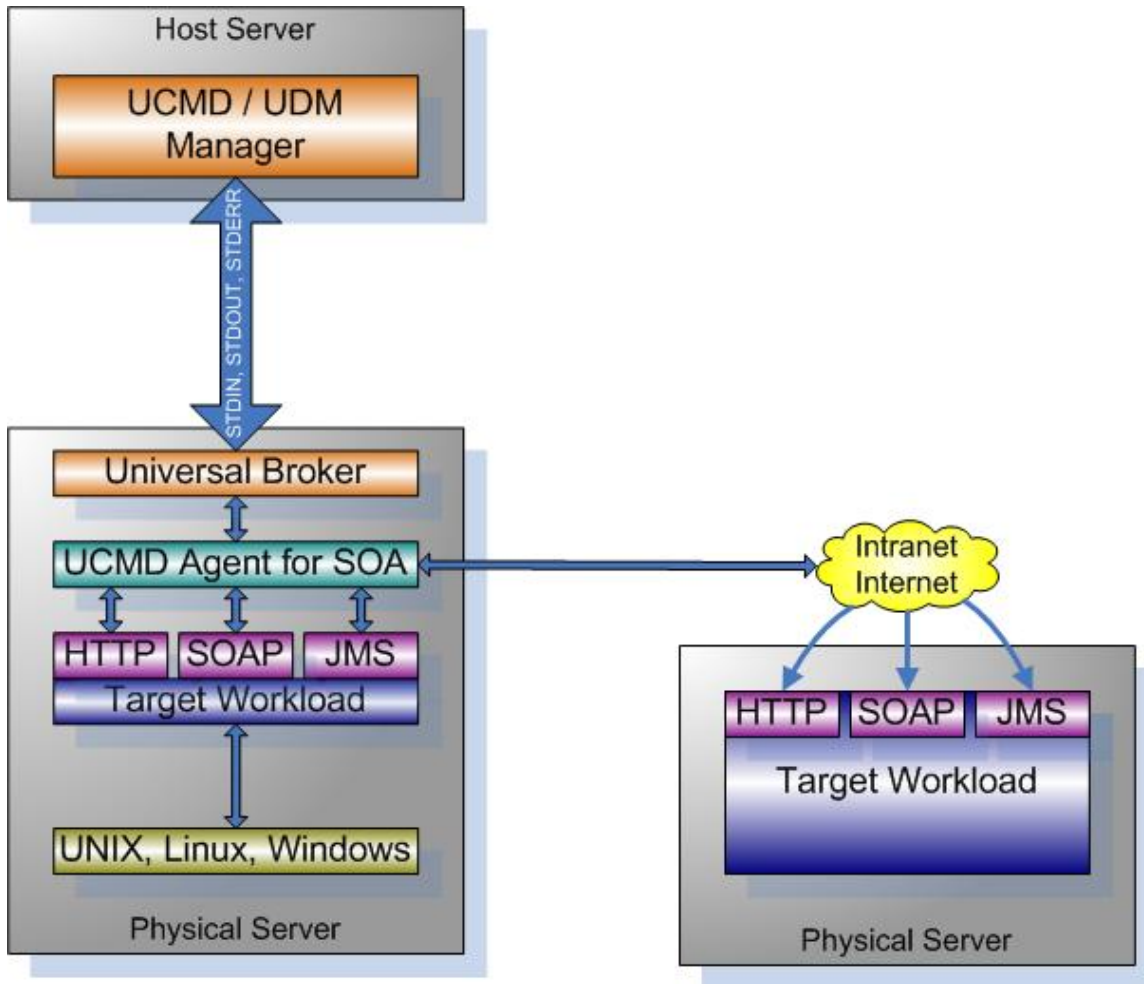
- Universal Broker and Workload Automation 5 for SOA are located together on one physical server.
- Target workload is located on a different physical server.



### Note

The target workload location is user-defined. The `SERVICE_URL` option specifies how Workload Automation 5 for SOA knows where to look for the target workload.

The following figure illustrates the Distributed Server Deployment (DSD).



### DSD Deployment Flow of Events





The flow is nearly the same as with the Single Server Deployment except that the location of the target workload is not **localhost**.

## Workload Automation 5 for SOA for UNIX - Pre-Installation-Upgrade Backups

### Workload Automation 5 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.

File	Location
UAC.xml File	<code>/etc/universal</code>
Log4jConfiguration.xml File *	<code>/etc/universal (UAC)</code> <code>/etc/universal (UAI)</code> <div style="background-color: #ffffcc; padding: 5px; margin-top: 10px;">  <b>Note</b>            The installation process does not overwrite Log4jConfiguration.xml files.         </div>
JMS Provider Client Jar Files	<code>/opt/universal/uac/container/webapps/axis2/WEB-INF/lib</code> <div style="background-color: #ffffcc; padding: 5px; margin-top: 10px;">  <b>Note</b>            The names of the jar files differ depending on which JMS Provider you are using.         </div>
JMS Provider Client Properties Files	<code>/opt/universal/uai/xml</code> <div style="background-color: #ffffcc; padding: 5px; margin-top: 10px;">  <b>Note</b>            These are suggested locations; you can place these files anywhere on the files system. If you have located these files under the <b>uai</b> directory, you should back them up.         </div>
Payload Files	<p>Normally, payload files should be located on the host system where Universal Command Manager is installed.</p> <p>If you have chosen to store them elsewhere, the suggested location is:</p> <code>/opt/universal/uai/xml</code> <div style="background-color: #ffffcc; padding: 5px; margin-top: 10px;">  <b>Note</b>            You can place these files anywhere on the files system. If you have located these files under the <b>uai</b> directory, you should back them up.         </div>

## Workload Automation 5 for SOA for UNIX - Distribution File

- [UNIX Distribution File](#)
- [Obtaining the Distribution File](#)
- [Distribution File Format](#)

### UNIX Distribution File

Stonebranch, Inc. provides different Workload Automation for SOA for UNIX packages for different types of UNIX operating systems.

### Obtaining the Distribution File

To obtain the Workload Automation for SOA for UNIX package for your type of UNIX operating system, you must download the corresponding product distribution file from the Stonebranch [Customer Portal](#).



#### Note

A customer user name and password — provided by Stonebranch, Inc. — are required to access the Customer Portal.

After a distribution file has been downloaded, the installation files contained in that file must be extracted before the product can be installed (see [Workload Automation 5 for UNIX - Installation Procedures](#)).

### Distribution File Format

The name of each Workload Automation for SOA for UNIX distribution file has the following format:

```
sb-soa-Version.Release.Modification Level.Maintenance Level-operating
system-version(.release)(-platform).tar.Z
```

For example: `sb-soa-5.1.0.0-linux-2.4-i386.tar.Z`

In this format:

- **Version** is the current version of Workload Automation 5 for SOA.
- **Release** is the current release of Workload Automation 5 for SOA.
- **Modification Level** is the current Workload Automation 5 for SOA feature set.
- **Maintenance Level** is the Workload Automation 5 for SOA build level.
- **operating system** is the name of the operating system (for example, AIX or Linux).
- **version(.release)** is the supported version and, optionally, the release of the operating system.
- **platform** is the targeted hardware platform (for example, i386). It is included in the file name only if there is more than one platform available for the specified operating system.

## Workload Automation 5 for SOA for UNIX - Installation Procedures

Error formatting macro: redirect: java.lang.NullPointerException



## **Workload Automation 5 for SOA for UNIX - Installation Procedures Overview**

### **Workload Automation 5 for SOA for UNIX Installation Procedures**

The following procedures are provided for the installation and modification of Workload Automation 5 for SOA for UNIX:

- [Workload Automation 5 for SOA for AIX Installation](#)
- [Workload Automation 5 for SOA for Linux Installation](#)

### **Installation Process**

Installation is a straightforward process using the platform-specific package. The installed environment is self-contained and follows current Workload Automation 5 installation standards.

### ***Installation Prerequisite***

Before installing Workload Automation 5 for SOA 5.1.0, you must install Universal Products 3.2.0 or later.

## Workload Automation 5 for SOA for AIX Installation

- [Product Distribution File](#)
- [Unpacking and Installation Procedures](#)
- [Removing Workload Automation 5 for SOA for AIX](#)
- [Listing Workload Automation 5 for SOA for AIX Information](#)

### Product Distribution File

The Workload Automation 5 for SOA for AIX product distribution file is in a compressed **tar** format.

The name of the Workload Automation 5 for SOA for AIX distribution file has the following format:

```
sb-soa-5.1.0.0-aix-5.3.tar.Z
```

### Unpacking and Installation Procedures

To unpack and install Workload Automation 5 for SOA, perform the following steps:

<b>Step 1</b>	Create a directory (or select an existing directory) in which to save the package file.
<b>Step 2</b>	Save the package file into that directory.
<b>Step 3</b>	<p>Uncompress and extract the installation files in the current working directory. The command to extract the files is:</p> <pre>zcat sb-soa-5.1.0.0-aix-5.3.tar.Z   tar xvf -</pre> <p>If your operating system does not support the <b>zcat</b> command, use the following command:</p> <pre>gunzip sb-soa-5.1.0.0-aix-5.3.tar.Z</pre> <p>The output of the <b>gunzip</b> command provides the following <b>tar</b> file:  <pre>tar -xvf sb-soa-5.1.0.0-aix-5.3.tar</pre></p>
<b>Step 4</b>	<p>After the extraction is complete, run the installation script, <b>upsinst</b>, which executes the <b>installp</b> command:</p> <pre>./upsinst</pre> <p>An installation log is written to file <b>install.log</b> in the current directory. <b>upsinst</b> automatically restarts the Universal Broker daemon, <b>ubrokerd</b>, at the end of the install.</p>
<b>Step 5</b>	<p>From the license file that was sent to you by Stonebranch, Inc., add the license information to the following file:  <pre>/etc/universal/uacs.conf</pre></p>

**Step 6** Recycle **ubroker** using the following commands (cd to **/opt/universal/ubroker**)

First:

```
./ubrokerd stop
```

Then:

```
./ubrokerd start
```

**Step 7** Use the **uquery** command (cd to **/opt/universal/bin**) to validate that the Universal Application Container Server component of Universal Command Agent for SOA 5.1.0 is running:

**uquery -i localhost** (or the name of your server)

The output for Universal Application Container Server should have the following format:

```
Component ID.....: 1360109684
Component Name.....: uac (Server)
Component Description....: Universal Application Container Server
Component Version.....: 5.1.0 Level 0 Release Build 101
Component Type.....: uac
Component Process ID.....: 23331000
Component Start Time.....: 18:14:42
Component Start Date.....: 02/05/13
Component Command ID.....: uac
Component State.....: REGISTERED
Component MGR UID.....:
Component MGR Work ID.....:
Component MGR Host Name...:
Component MGR IP Address..:
Component MGR Port.....:
Component Comm State.....: ESTABLISHED
Component Comm State Time.: 18:14:44
Component Comm State Date.: 02/05/13
Component MGR Restartable.: NO
Component Comment.....:
```

**Removing Workload Automation 5 for SOA for AIX****Note**

Before removing Workload Automation 5 for SOA for AIX, stop the **ubrokerd** daemon. Also, it is strongly recommended that you back up existing data before removing Workload Automation 5 for SOA for AIX.

Workload Automation 5 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 Workload Automation 5 for SOA for AIX Information**

On AIX, information on an installed product is listed with the **lspp** command. The command must be executed with the superuser ID.

Issue the following command to list information for Workload Automation 5 for SOA for AIX:

```
lslpp -La 'UPS*'
```

## Workload Automation 5 for SOA for Linux Installation

- [Product Distribution File](#)
- [Unpacking and Installation Procedures](#)
- [Removing Workload Automation 5 for SOA for Linux](#)
- [Listing Workload Automation 5 for SOA for Linux Information](#)

### Product Distribution File

Workload Automation 5 for SOA 5.1.0 is packaged as an RPM file (extension **.rpm**). It is installed using the Linux **rpm** command.

The name of the Workload Automation 5 for SOA for Linux distribution file has the following format:

```
sb-soa-5.1.0.0-linux-2.4-i386.tar.z
```

### Unpacking and Installation Procedures

To unpack and install Workload Automation 5 for SOA, perform the following steps:

<b>Step 1</b>	Create a directory (or select an existing directory) in which to save the package file.
<b>Step 2</b>	Save the package file into that directory.
<b>Step 3</b>	<p>Uncompress and extract the installation files in the current working directory. The command to extract the files is:</p> <pre>zcat sb-soa-5.1.0.0-linux-2.4-i386.tar.Z   tar xvf -</pre> <p>If your operating system does not support the <b>zcat</b> command, use the following command:  <b>gunzip sb-soa-5.1.0.0-linux-2.4-i386.tar.Z</b></p> <p>The output of the <b>gunzip</b> command provides the following <b>tar</b> file:  <b>tar -xvf sb-soa-5.1.0.0-linux-2.4-i386.tar</b></p>
<b>Step 4</b>	<p>After the extraction is complete, run the installation script, <b>ups*inst*</b>, which executes the <b>rpm</b> command:</p> <pre>./upsinst</pre> <p>An installation log is written to file <b>install.log</b> in the current directory. <b>upsinst</b> automatically restarts the Universal Broker daemon, <b>ubrokerd</b>, at the end of the install.</p>
<b>Step 5</b>	From the license file that was sent to you by Stonebranch, Inc., add the license information to the <b>/etc/universal/uacs.conf</b> file.

**Step 6** Recycle ubroker using the following commands (cd to `/opt/universal/ubroker`):

First:

```
./ubrokerd stop
```

Then:

```
./ubrokerd start
```

**Step 7** Use the **uquery** command (cd to `/opt/universal/bin`) to validate that the Universal Application Container Server component of Universal Command Agent for SOA 5.1.0 is running:

**uquery -i localhost** (or the name of your server)

The output for Universal Application Container Server should have the following format:

```
Component ID.....: 1360109684
Component Name.....: uac (Server)
Component Description.....: Universal Application Container Server
Component Version.....: 5.1.0 Level 0 Release Build 101
Component Type.....: uac
Component Process ID.....: 23331000
Component Start Time.....: 18:14:42
Component Start Date.....: 02/05/13
Component Command ID.....: uac
Component State.....: REGISTERED
Component MGR UID.....:
Component MGR Work ID.....:
Component MGR Host Name...:
Component MGR IP Address..:
Component MGR Port.....:
Component Comm State.....: ESTABLISHED
Component Comm State Time.: 18:14:44
Component Comm State Date.: 02/05/13
Component MGR Restartable.: NO
Component Comment.....:
```

## Removing Workload Automation 5 for SOA for Linux



### Note

Before removing Workload Automation 5 for SOA for Linux, stop the **ubrokerd** daemon. Also, it is strongly recommended that you back up existing data before removing Workload Automation 5 for SOA for Linux.

Workload Automation 5 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 Workload Automation 5 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 Workload Automation 5 for SOA for Linux, issue the following command:

```
rpm -qi ups
```

## Workload Automation 5 SOA for UNIX Installation - File Inventory Lists

- [Workload Automation 5 for SOA for UNIX - File Inventory Lists](#)
- [Parent Directories](#)
- [Product Directories and Files](#)
  - [/opt/universal Parent Directory](#)
  - [/var/opt/universal Parent Directory](#)
  - [/etc/universal Parent Directory](#)

### Workload Automation 5 for SOA for UNIX – File Inventory Lists

This page identifies the Workload Automation 5 for SOA file system hierarchy and its contents.

The parent directories under which Workload Automation 5 for SOA operates are based on the existing Workload Automation 5 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.

### Parent Directories

For UNIX, the parent directories are:

- **/opt/universal**
- **/var/opt/universal**

The following sections identify the directories and files located under each of these parent directories.

### Product Directories and Files

#### /opt/universal Parent Directory

The following table identifies the Workload Automation 5 for SOA for UNIX product directories and files located under the **/opt/universal** parent directory.

Directory / File	Description
<b>uac</b>	Directory containing artifacts for the UAC component. It includes the following sub-directories and files.
<b>bin</b>	Subdirectory containing the <b>uacsvr</b> executable. It requires no user interaction.
<b>lib</b>	Subdirectory containing the <b>uacsvr.jar</b> file needed for communication between the <b>uacsvr</b> and <b>uac</b> components. It requires no user interaction.
<b>container</b>	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.
<b>shutdownUAC.sh</b>	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.



<b>startUAC.sh</b>	Startup script for UAC. You usually will not need to use this script, as it is the responsibility of Universal Broker to start UAC.
<b>uacValidateInbound.sh</b>	Script that validates the contents of the UAC.xml file using the schema definition contained in <b>UAC.xsd</b> .
<b>UAC.xsd</b>	File used by UAC to validate content of the <b>UAC.xml</b> inbound configuration file. It requires no user interaction.
<b>uai</b>	Directory containing the artifacts for the UAI component. It includes the following sub-directories and files.
<b>lib</b>	Subdirectory containing the libraries needed for UAI operation. It usually requires no user interaction.
<b>UAI.sh</b>	Shell script used to invoke the UAI component. It usually requires no user interaction.
<b>jre</b>	Directory containing the Java Runtime Environment for the UAC, UAI, and UAC Server components. It usually requires no user interaction.
<b>nls/uacmceng.umc</b>	Universal Application Container Server English message catalog.
<b>Readme.ups</b>	Workload Automation 5 for SOA for UNIX installation README file.
<b>ubroker/tmpl/uaccfg</b>	Universal Application Container Server configuration template file.
<b>ubroker/tmpl/uaccmp</b>	Universal Application Container Server component definition template file.

### **/var/opt/universal Parent Directory**

The following table identifies the Workload Automation 5 for SOA for UNIX product directories and files located under the **/var/opt/universal** parent directory.

<b>Directory / File</b>	<b>Description</b>
<b>log/uac</b>	Directory containing logging and work artifacts for the UAC component. It includes the following sub-directories and files.
<b>temp</b>	Subdirectory used by UAC for temporary file persistence. It requires no user interaction.
<b>work</b>	Subdirectory used by the web services framework for temporary operations. It requires no user interaction.

<b>catalina.out</b>	Container engine log file.
<b>container.log</b>	Web services framework log file.
<b>derby.log</b>	Database log file.
<b>uac.log</b>	UAC log file.
<b>log/uai</b>	Directory containing the logging and work artifacts for the UAI component. It is empty by default. If <b>trace</b> or <b>info</b> is enabled, the output would be written to this directory.
<b>uac_db</b>	Directory containing the database components, log, property, and lock files. It requires no user interaction.

### **/etc/universal Parent Directory**

The following table identifies the Workload Automation 5 for SOA for UNIX product directories and files located under the **/etc/universal** parent directory.

<b>Directory / File</b>	<b>Description</b>
<b>uac_log4jConfiguration.xml</b>	Logger configuration file that you may need to modify if you want to change the logging level of the UAC component.
<b>uai_log4jConfiguration.xml</b>	Logger configuration file that you may need to modify if you want to change the logging level of the UAI component.
<b>uacs.conf</b>	Universal Application Container Server configuration file.
<b>UAC.xml</b>	Inbound operation configuration file that you may need to modify to support your inbound workload operations.
<b>comp/uac</b>	Universal Application Container Server component definition file.

## IBM i Installation

Error formatting macro: redirect: java.lang.NullPointerException

## IBM i Installation - Overview

- [IBM i Installation](#)
- [Installation Summary](#)
- [Naming Conventions](#)

## IBM i Installation

These pages provide information on the installation of Stonebranch Inc's Workload Automation 5 on the IBM i operating system. Unless otherwise specified, all references to Workload Automation for IBM i in these pages refer to version 5.1.0.

All Workload Automation for IBM i 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 Workload Automation component is running, including manager components. The local Broker maintains product configuration data for all components that have a configuration file.

## Installation Summary



### Warning

This installation procedure is intended only for default installations into simple environments. Installing into high availability environments may require a customized installation. For customized installation, parallel installation or multiple system roll out see the Installation Guide for IBM i.

<b>Step 1</b>	Download the distribution file from the Stonebranch <a href="#">Customer Portal</a> to a computer running a supported version of Windows or UNIX.
<b>Step 2</b>	Extract the Licensed Program Save File from the distribution file.
<b>Step 3</b>	If transferring the installation file to the native, QSYS file system, create a save file, UNV510, on the target IBM I system in library QGPL. This step is not necessary if transferring the file to the IFS.
<b>Step 4</b>	FTP the Licensed Program Save File to either the save file created in step 1 or to /QSYS.LIB/QGPL.LIB/UNV510.SAVF.
<b>Step 5</b>	Execute the command <code>RSTLICPGM LICPGM(0UNV510) DEV(*SAVF) SAVF(QGPL/UNV510)</code> to install Workload Automation to the default libraries.
<b>Step 6</b>	Verify the correctness of the install by executing the following command: <code>CHKPRDOPT 0UNV510</code> .
<b>Step 7</b>	Optionally, rename the UNVUBR510 subsystem to UBROKER for consistent operation across future installations.

## Naming Conventions

In Workload Automation 5 for IBM i, some library names and object names include a **vrnm** (version, release, and modification) suffix, **510**. This **vrnm** suffix will change, as appropriate, for subsequent releases.

Additionally, some library names can be changed. These pages use the default names for these libraries:

- **UNVPRD510** (product library)
- **UNVTMP510** (temporary library)
- **UNVSPL510** (spool file library)
- **UNVCMDREF** (command reference library)

You can change the **UNVPRD510**, **UNVTMP510**, and **UNVSPL510** library names at installation time.

You can change **UNVCMDREF** only via the Universal Command Server configuration file.

## IBM i Installation - Installation Package

### Package Components

The Workload Automation for IBM i package includes the following Workload Automation components:

- Universal Broker 5.1.0
- Universal Command (Manager and Server) 5.1.0
- Universal Control (Manager and Server) 5.1.0
- Universal Copy 5.1.0
- Universal Data Mover (Manager and Server) 5.1.0
- Universal Display Log File for AS/400 log files 5.1.0
- Universal Encrypt 5.1.0
- Universal Message Translator 5.1.0
- Universal Query 5.1.0
- Universal Submit Job with z/OS WTO support 5.1.0

Workload Automation for IBM i is packaged with product identifier **0UNV510**. The product can be managed using the IBM i commands for working with licensed programs (such as **RSTLICPGM**, **SAVLICPGM**, and **DLTLICPGM**).



**Note**

For the 5.1.0 release, Universal Command and Universal Encrypt are packaged as one IBM i licensed program.

### Component Compatibility

The following table identifies the compatibility of Workload Automation for IBM i with previous component / product versions.

Component	Compatibility
Universal Broker 5.1.0	Stonebranch Solutions / Universal Products 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Command 5.1.0	Universal Command 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Control 5.1.0	Universal Control 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Copy 5.1.0	Universal Copy 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Data Mover 5.1.0	Universal Data Mover 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Display Log File for IBM i log files 5.1.0	Universal Display Log File for OS/400 log files 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Encrypt 5.1.0	Universal Encrypt 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Message Translator 5.1.0	Universal Message Translator 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.
Universal Query 5.1.0	Universal Broker 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.

Universal Submit Job with z/OS WTO support 5.1.0

Universal Submit Job with z/OS WTO support 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, 3.1.0, and 1.2.1.

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

## IBM i Installation - Installation Requirements

- System Requirements
- Platform Requirements
- Libraries and Logs
  - Universal Broker Job Log
  - Command Reference Library
  - Trace File Location Library
  - Spool Library

### System Requirements

The requirements for installation of Workload Automation for IBM i are:

- IBM i V5R4M0 or above.
  - TCP/IP.
  - User profile with \*ALLOBJ, \*SPLCTL, \*JOBCTL, and \*SECADM authorities.
  - About 360 megabytes of disk space.
  - Network-attached workstation.
  - PTF SI27629 installed on V5R4 systems.
- For additional information, see the documentation for APAR SE28859 and PTF SI27629 at both:
- [https://www-912.ibm.com/n\\_dir/nas4apar.nsf/aaf5d88f9cc2ee10862571020058635c/a2b5a3b3ac874a9d862572d8003c7457?Op](https://www-912.ibm.com/n_dir/nas4apar.nsf/aaf5d88f9cc2ee10862571020058635c/a2b5a3b3ac874a9d862572d8003c7457?Op)
  - <https://www-912.ibm.com/a&#95;dir/as4ptf.nsf/c2fd98f5d2eccb83862574ce00520341/7529dd654f63941b862572eb0058109>

The Workload Automation installation process creates a product user profile, **UNVUBR510**, that is given **\*ALLOBJ** special authority. **UNVUBR510**, 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. Workload Automation 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 [Indesca 5.1.0 User Guide](#) and [Infitran 5.1.0 User Guide](#).

The **\*BASE** option contains the complete installation. This includes all of the components and utilities listed above.

### Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opwise Automation Center 5.1.1](#) and [Indesca-Infitran 5.1.0](#) page to make sure that your platform is supported before performing an installation.

### Libraries and Logs

Under the IBM i native file system, Workload Automation 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 IBM i, Universal Broker writes its messages to the **UBROKER** job log.

#### Space

Job log file growth is dependent on use of the Workload Automation 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 **UNVCMDFREF**. 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 **UNVCMDFREF** 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 IBM i, Workload Automation 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 (**UNVTMP510**).

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 IBM i 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 **UNVUBR510** user profile requires at least **\*CHANGE** authority to the **UNVTMP510** 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 Workload Automation 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 **UNVSPL510** library.

### Space

The spool files are located in the **UNVSPL510** 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:

**MAX-PROCESSES x AVERAGE NUMBER-OF-RECORDS x 32764 = 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 [Indesca 5.1.0 User Guide](#).

## Security

The **UNVUBR510** user profile requires at least **\*CHANGE** authority to the **UNVSPL510** library to create and use the spool files. No general user access is required.

No other Workload Automation components access the spool library.

## IBM i Installation - Distribution File

- IBM i Distribution File
- Obtaining the Distribution File
- Distribution File Format
  - Full Release
  - Maintenance Release
- Distribution File Contents
  - Full Release
  - Maintenance Release

### IBM i Distribution File

The Workload Automation for IBM i product distribution file contains all of the files required for the installation of the Workload Automation for IBM i package.

### Obtaining the Distribution File

To obtain the Workload Automation for IBM i package, you must download the corresponding product distribution file from the Stonebranch Customer Portal.



#### Note

A customer user name and password — provided by Stonebranch, Inc. — are required to access the Customer Portal.

### Distribution File Format

The format of the Workload Automation for IBM i 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:

`sb-Version.Release.Modification Level.Maintenance Level-operating system-version.release.tar.Z`

For example: `sb-5.1.0.0-as400-5.4.tar.Z`

In this format:

- **Version** is the current version of Workload Automation.
- **Release** is the current release of Workload Automation.
- **Modification Level** is the current Workload Automation feature set.
- **Maintenance Level** is the Workload Automation 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: `sb-5.1.0.1-as400-ptfs.tar.Z`

(See [Transferring for a Maintenance Release](#) for information about PTFs in maintenance release distribution.)

### Distribution File Contents

The distribution file contains the installation files required for the installation of Workload Automation for IBM i.

The Workload Automation Licensed Program for IBM i is distributed as an IBM i Save File.

Depending on whether the installation is for a full release or a maintenance release of Workload Automation for IBM i, the distribution file contains different installation files.

## Full Release

The following table lists the installation files included in the distribution file for a full release of Workload Automation for IBM i.

File Name	Description
<b>UNV510.SAVF</b>	Workload Automation Licensed Program Save File.
<b>README.TXT</b>	Summary of the installation procedure.

## Maintenance Release

The following table lists the installation files included in the distribution file for a maintenance release of Workload Automation for IBM i.

File Name	Description
<b>CUMUNV510</b>	Workload Automation Cumulative PTF Save File.
<b>README.TXT</b>	Summary of the installation procedure.

## IBM i Installation - Transferring to IBM i

- [Workload Automation for IBM i - Transferring Workload Automation to IBM i](#)
- [Transferring for a Full Release](#)
- [Transferring for a Maintenance Release](#)

### Workload Automation for IBM i – Transferring Workload Automation to IBM i

In order to install Workload Automation for IBM i, you must first transfer the Workload Automation Licensed Program for IBM i to the IBM i operating system.

The transfer is comprised of three procedures:


1. Download the product distribution file from the Stonebranch [Customer Portal](#) (see [IBM i Installation - Distribution File](#)) to Windows, UNIX, or IBM i shell with tar and zcat utilities.
2. Extract all files from the distribution file.
3. Transfer the IBM i Save File (the Workload Automation Licensed Program) to a library on an iSeries.

This page provides separate transferring procedures for:

- [Transferring for a Full Release](#)
- [Transferring for a Maintenance Release](#)

### Transferring for a Full Release

To transfer a full release package to IBM i — whether for a new installation of a full release or an upgrade installation to a new release — perform the following steps:

<b>Step 1</b>	Download the distribution file to a computer running a supported version of Windows or UNIX.
<b>Step 2</b>	<p>Extract the installation files from the distribution file:</p> <ul style="list-style-type: none"> <li>• To extract the installation files on UNIX, QShell, or PASE shell, run:</li> </ul> <pre style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;">zcat *.Z   tar xvf -</pre> <div style="background-color: #ffffcc; padding: 10px; border: 1px solid #ccc; margin: 10px 0;"> <p> <b>Note</b> The space and the - character after <b>xvf</b> must be included.</p> </div> <ul style="list-style-type: none"> <li>• 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.</li> </ul>
<b>Step 3</b>	<p>Create a save file on the target IBM i system in library <b>QGPL</b>. On the IBM i command line, execute the following command:</p> <pre style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;">CRTSAVF FILE(QGPL/UNV510)</pre>

**Step 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\Workload Automation for as400`

```
ftp your_as400
Name: your_name
Password: your_password
ftp> cd QGPL
ftp> bin
ftp> lcd "c:\temp\Workload Automation for as400"
ftp> put UNV510.SAVF UNV510
ftp> quit
```

## Transferring for a Maintenance Release

To upgrade Workload Automation from a currently installed 5.1.0 release to 5.1.0 maintenance release, cumulative PTFs are used. These cumulative PTFs are distributed as IBM i Save Files and can be downloaded from the Stonebranch [Customer Portal](#). PTFs should be installed as user **QSECOFR** or a user with equivalent authority.

To transfer a maintenance release package to IBM i, perform the following steps:

**Step 1** Download the distribution file to a computer running a supported version of Windows or UNIX.

**Step 2** Extract the distribution file:

- To extract the \*.tar.Z file on UNIX, run:

```
zcat *.Z | tar xvf -
```



**Note**

The space and the - character after **xvf** must be included.

- 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.

**Step 3** Create a save file on the target IBM i system in library **QGPL**. On the IBM i command line, execute the following command:

```
CRTSAVF FILE(QGPL/CUMUNV510)
```

**Step 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 CUMUNV510 CUMUNV510
ftp> quit
```

## IBM i Installation - Installation Procedures

Error formatting macro: redirect: java.lang.NullPointerException

## IBM i Installation - Installation Procedures Overview

### Installation Processes

There are six different processes for installing Workload Automation on an IBM i system:

1. [IBM i Installation - New Installation \(Default\)](#)
2. [IBM I Installation - New Installation \(Custom\)](#)
3. [IBM I Installation - Re-Installation of Same Release](#)
4. [IBM I Installation - Upgrade Installation to New Release](#)
5. [IBM i Installation - Propagating New Release to Additional Systems](#)
6. [IBM i Installation - Upgrade Installation for Maintenance Release](#)

Your installation environment, currently installed Workload Automation (if any), and Workload Automation release to be installed determine the process to use.

There are two additional procedures related to installation:

- [IBM i Installation - UCHGRLS \(Change Release Tag\) Program](#) for changing Workload Automation for IBM i command names in the IBM i **QSYS** library.
- [IBM i Installation - Product Removal](#) procedures that are required or optional, depending on the installation process.



## IBM i Installation - New Installation (Default)

### New Installation (Default)

The default process for a new installation of Workload Automation for IBM i installs to the following libraries:

- Product library (**UNVPRD510**)
- Temporary library (**UNVTMP510**)
- Spool file library (**UNVSPL510**)
- Command reference library (**UNVCMDREF**)



#### V5R4 systems

Before installing Workload Automation for IBM i, either download and apply PTF SI27629 or verify that it is applied to the system.

<b>Step 1</b>	<p>Execute the following command to install Workload Automation to the default libraries:</p> <pre style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">RSTLICPGM LICPGM(0UNV510) DEV(*SAVF) SAVF(QGPL/UNV510)</pre>
<b>Step 2</b>	<p>Verify that the installation was correct by executing the following command:</p> <pre style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">CHKPRDOPT 0UNV510</pre> <p>The following message should display: <b>No errors detected by CHKPRDOPT.</b></p>
<b>Step 3</b>	<p>Optionally, rename the UNVUBR510 subsystem to UBROKER for consistent operation across future installations.</p>
<b>Step 4</b>	<p>Optionally, use the <a href="#">UCHGRSL (Change Release Tag) Program</a> to change the names of the Workload Automation for IBM i commands in the IBM i <b>QSYS</b> library.</p>

## IBM I Installation - New Installation (Custom)


### New Installation (Custom)

The custom method for a new installation of Workload Automation for IBM i allows installation to libraries other than **UNVPRD510**, **UNVTMP510**, and **UNVSPL510**, as required by your environment.



#### V5R4 systems

Before installing Workload Automation for IBM i, either download and apply PTF SI27629 or verify that it is applied to the system.

<b>Step 1</b>	<p>Execute the following command to install Workload Automation to user-specified libraries:</p> <pre style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">RSTLICPGM LICPGM(0UNV510) DEV(*SAVF) SAVF(QGPL/UNV510) LIB(prodLib tmpLib spoolLib)</pre> <div style="background-color: #ffffcc; padding: 5px; margin: 10px 0;"> <p> <b>Note</b> Replace <b>prodLib</b> with the desired product library name, <b>tmpLib</b> with the desired temporary library name, and <b>spoolLib</b> with the desired spool library name.</p> </div>
<b>Step 2</b>	<p>Verify that the installation was correct by executing the following command:</p> <pre style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">CHKPRDOPT 0UNV510</pre> <p>The following message should display: <b>No errors detected by CHKPRDOPT.</b></p>
<b>Step 3</b>	<p>Optionally, rename the UNVUBR510 subsystem to UBROKER for consistent operation across future installations.</p>
<b>Step 4</b>	<p>Optionally, use the <a href="#">UCHGRS (Change Release Tag) Program</a> to change the names of the Workload Automation for IBM i commands in the IBM i <b>QSYS</b> library.</p>

## IBM I Installation - Re-Installation of Same Release

### Re-Installation Using Same Libraries

If you are reinstalling a Workload Automation for IBM i release (5.1.0 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.

<b>Step 1</b>	Back up all file objects in all of the Workload Automation libraries – product, temporary, spool, and command reference – prior to re-installation (in case errors occur during the re-installation process).
<b>Step 2</b>	End the Universal Broker subsystem, <b>UNVUBR510 (UBROKER if renamed)</b> .
<b>Step 3</b>	Run <b>RSTLICPGM</b> to install Workload Automation using the library names used during the initial 5.1.0 installation.
<b>Step 4</b>	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.
<b>Step 5</b>	Restart <b>UNVUBR510</b> .
<b>Step 6</b>	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.
<b>Step 7</b>	If <b>UCHGRLS</b> was previously run to remove 510 tags from the Workload Automation commands in <b>QSYS</b> , manually delete the newly created, 510-tagged Workload Automation commands in <b>QSYS</b> . A list of those commands can be found in <a href="#">QSYS Library</a> .
	<div style="background-color: #ffffcc; padding: 10px;"> <b>Note</b> Do not remove the <b>UNVUBR510</b> user profile from <b>QSYS</b>.         </div>

### Re-Installation Using Different Library Names

Before reinstalling a Workload Automation for IBM i release (5.1.0 and later) into different libraries, you must remove the currently installed release.

<b>Step 1</b>	Save copies of all configuration files and component definition files; otherwise, the configuration and component information will be lost.
<b>Step 2</b>	Remove the currently installed release (see <a href="#">Product Removal</a> ).
<b>Step 3</b>	Re-install the 5.1.0 release. See <a href="#">New Installation (Default)</a> and <a href="#">New Installation (Custom)</a> to determine which process is most appropriate for your environment.
<b>Step 4</b>	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.
<b>Step 5</b>	Optionally, use the <a href="#">UCHGRLS (Change Release Tag) Program</a> to change the names of the Workload Automation for IBM i commands in the IBM i <b>QSYS</b> library.

## IBM I Installation - Upgrade Installation to New Release

- Upgrade Procedures
  - Print Files Removal
  - Currently Installed Release
- Install (Default or Custom)
- Post-Install Tasks
- Pre-Production Tasks
  - Change Release Tags
  - Change Port Address

### Upgrade Procedures

To upgrade Workload Automation for IBM i from a currently installed pre-5.1.0 release (Stonebranch Solutions release 4.3.0 or earlier) to a 5.1.0 release, the following procedures are required:

1. Install (Default or Custom)
2. Post-Install Tasks
3. Pre-Production Tasks



#### V5R4 systems

Before installing Workload Automation for IBM i, either download and apply PTF SI27629 or verify that it is applied to the system.

### Print Files Removal

In releases 4.1.0 - 4.3.0, Job Log Copy Keep used the **USBMJPRTF** and **UCMSVRPRTF** print files.

Starting with release 5.1.0, these print files are no longer used and will be removed during the installation process. The spooled job logs will not be lost because they are spooled to the **QEZJOBLOG** output queue.

To print job logs from a previous release, use the **CHGSPLFA** command to associate the job logs with a different printer file.

### Currently Installed Release

You are not required to delete a currently installed pre-5.1.0 release (Stonebranch Solutions 4.3.0 or earlier) of Workload Automation for IBM i before installing a 5.1.0 release. You can do so either before or after installing the 5.1.0 release (see [Product Removal](#)).

However, if you want to install a 5.1.0 release to the previously used installation libraries of a pre-5.1.0 release, you must first uninstall that pre-5.1.0 release:

<b>Step 1</b>	Save configuration, component, NLS, and any other files from the product library ( <b>UNVPRD*** / UNIVERSAL</b> ) that may contain information needed after the new installation.
<b>Step 2</b>	Save any files from the command reference library ( <b>UNVCMDREF</b> ) and temporary library ( <b>UNVTMP*** / UNVTMP</b> ) that are needed, as these libraries will be deleted by the uninstall process.
<b>Step 3</b>	Follow the release-appropriate uninstall process (see <a href="#">Product Removal</a> ) and the New Release installation procedure (see <a href="#">Install (Default or Custom)</a> ), below, to install the new release.
<b>Step 4</b>	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 Workload Automation release (5.1.0) are the same as the processes for a new installation.

See [New Installation \(Default\)](#) and [New Installation \(Custom\)](#) to determine which process is most appropriate for your environment.

### Post-Install Tasks

After the installation upgrade of Stonebranch Solutions 4.3.0 (or previous release) to Workload Automation, perform the following steps:

<b>Step 1</b>	Copy any customized code page files from the <b>UNVNLS</b> file members from the previous release to the newly installed <b>UNVNLS</b> file.
	 <b>Note</b> Do not simply copy the old <b>UNVNLS</b> file to replace the new <b>UNVNLS</b> file. The new files contain many updates.
<b>Step 2</b>	If needed, manually update the newly installed configuration file, <b>UNVCONF</b> , from the file of the previous release. The new file contains many new configuration options. At a minimum, the Workload Automation license entries must be entered into the configuration file members for the managers of each licensed product.
<b>Step 3</b>	If the original component file ( <b>UNVCOMP</b> ) was modified, update the newly installed <b>UNVCOMP</b> file for each of the modified members.
<b>Step 4</b>	If either or both of the exit programs ( <b>UCMSJOB1</b> and <b>UCMSJOB2</b> ) 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, <b>UNVPRD510</b> .
	 <b>Note</b> You cannot replace <b>UCMSJOB1</b> for 5.1.0 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 <b>UCMSJOB1</b> in the <a href="#">Universal Command 5.1.0 Reference Guide</a> for additional information.

## Pre-Production Tasks

### Change Release Tags


When you are ready to take the new release to production, you can use the **UCHGRLS** (Change Release Tag) program to:

- Rename the Workload Automation for IBM i commands in **QSYS** to the untagged command names in **UNVPRD510**.
- Tag the pre-5.1.0 release command names in **QSYS** with the version / release / modification number of that release (for example, **320**).

See [UCHGRLS \(Change Release Tag\) Program](#) for information on using **UCHGRLS**.

### Change Port Address

No two Workload Automation 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. Workload Automation 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.

## IBM i Installation - Propagating New Release to Additional Systems

### Propagating New Release to Additional Systems

You may want to install Workload Automation on a pre-production system, implement needed changes such as customizing configuration files, and then propagate these changes to other systems.

One method to propagate the customized product is via the **SAVLICPGM** and **RSTLICPGM** commands (see [SAVLICPGM](#) and [RSTLICPGM Method](#), below).



#### Warning

A number of Stonebranch objects are created during the installation process. These include, but are not limited to:

- Commands in QSYS
- UBROKER job queue and class
- Job definitions in the product library
- Universal Broker Subsystem
- Print files in the product library.

The default command reference library, **UNVCMDREF**, is created during installation unless a library of that name exists on the system. This process only recreates the objects previously identified. It will not propagate changes made to those objects.



#### V5R4 systems

Before installing Workload Automation for IBM i, either download and apply PTF SI27629 or verify that it is applied to the system.

### SAVLICPGM and RSTLICPGM Method

The following steps are recommended:

<b>Step 1</b>	Install Workload Automation (see <a href="#">New Installation (Default)</a> and <a href="#">New Installation (Custom)</a> ).
<b>Step 2</b>	Download the latest product maintenance from the Stonebranch <a href="#">Customer Portal</a> and temporarily apply the PTFs.  Allow the system to automatically handle any permanent PTF application via the <b>APYPTF</b> command. See <a href="#">Upgrade Installation for Maintenance Release</a> for detailed instructions on applying a maintenance release.
<b>Step 3</b>	Customize the Workload Automation installation as described in <a href="#">Customization</a> .
<b>Step 4</b>	Verify that the 5.1.0 Universal Broker Subsystem, <b>UNVUBR510</b> , is not active.
<b>Step 5</b>	To create a save file in which to save the customized product, invoke:  <pre>CRTSAVF FILE(QGPL/UNV510CUST) TEXT('Customized installation file for SB Prod 5.1.0')</pre>
<b>Step 6</b>	Invoke:  <pre>SAVLICPGM LICPGM(0UNV510) DEV(*SAVF) SAVF(QGPL/UNV510CUST)</pre>

<b>Step 7</b>	If needed, restart the Universal Broker Subsystem:  <pre>STRSBS UNVPRD510/UNVUBR510</pre>
<b>Step 8</b>	Via what ever mechanism you prefer, move the save file to each system on which you want to restore the product.
<b>Step 9</b>	Invoke:  <pre>RSTLICPGM LICPGM(0UNV510) DEV(*SAVF) SAVF(QGPL/UNV510CUST)</pre>

**Note**

Although this procedure describes use of a save file, any other available save and restore device can be used.

## IBM i Installation - Upgrade Installation for Maintenance Release

### Upgrade Installation for Maintenance Release

To upgrade Workload Automation for IBM i from a currently installed release to a maintenance release, perform the following steps to install the cumulative PTFs:

<p><b>Step 1</b></p>	<p>If it is running, end the Universal Broker subsystem (<b>UNVUBR510</b>) on the IBM i system. On the IBM i, enter the following command:</p> <pre>===&gt; ENDSBS SBS(UNVUBR510)</pre>
<p><b>Step 2</b></p>	<p>Load the PTF. On the IBM i, enter the following command:</p> <pre>===&gt; LODPTF LICPGM(0UNV510) DEV(*SAVF) SAVF(QGPL/CUMUNV510)</pre>
<p><b>Step 3</b></p>	<p>Apply the PTF. On the IBM i, enter the following command:</p> <pre>===&gt; APYPTF LICPGM(0UNV510)</pre> <p>This command applies all PTFs contained in the cumulative PTF file.</p> <p>If you want to apply only selected PTFs, enter the following command:</p> <pre>===&gt;APYPTF LICPGM(0UNV510) SELECT(ptf1 ptf2 ...)</pre> <p>This command applies the PTF temporarily.</p> <p>If you want to apply the PTFs permanently after successful testing, re-execute the command with the parameter <b>APY(*PERM)</b>.</p>
<p><b>Step 4</b></p>	<p>Restart <b>UNVUBR510</b>.</p>



## IBM i Installation - UCHGRLS (Change Release Tag) Program

- [Overview](#)
- [Using UCHGRLS](#)
- [UCHGRLS Examples](#)

### Overview

All installations of Workload Automation for IBM i, except an upgrade for maintenance, place the Workload Automation for IBM i commands in the product library (**UNVPRD510**) and the IBM i **QSYS** library.

Starting with the 3.2.0 release of Universal Products for IBM i, the command names in **QSYS** are tagged with the Workload Automation for IBM i 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 IBM i **QSYS** library from tagged to untagged.

### Using UCHGRLS

**UCHGRLS** resides in **UNVPRD510**. 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 = **510**.

This changes the tagged command names in **QSYS** to the untagged command names in **UNVPRD510**.

For an upgrade installation (from a pre-5.1.0 release to 5.1.0), **UCHGRLS** tags the pre-5.1.0 release command names in **QSYS** with the version / release / modification number of that release (for example, **510**). See [Object Inventory Lists](#) for tagged command name details.

### UCHGRLS Examples

This example runs **UCHGRLS** with the default values, following installation of **0UNV510** with **0UNV410** existing on the system.

```
UNVPRD510/UCHGRLS
```

This example runs **UCHGRLS** to revert back to **0UNV320** for the default names in **QSYS** with **0UNV510** installed. (This should be done prior to re-installing **0UNV510**, assuming that **0UNV320** remains installed on the system.)

```
UNVPRD510/UCHGRLS SETTAGVRM(510) RMVTAGVRM(320)
```

This example runs **UCHGRLS** to change the **0UNV510** commands in **QSYS** to the default names and to tag the **0UCM121** and **0UEN121** commands in **QSYS** with 121.

```
UNVPRD510/UCHGRLS SETTAGVRM(121) RMVTAGVRM(510)
```

This example runs **UCHGRLS** to set the **0UNV510** commands in **QSYS** to the default names, where **0UNV510** is the first Workload Automation installation on an IBM i system.

```
UNVPRD510/UCHGRLS SETTAGVRM(*NEW) RMVTAGVRM(510)
```

## IBM i Installation - Product Removal

- [Overview](#)
- [Universal Encrypt 1.2.1](#)
- [Universal Command 1.2.1](#)
- [Universal Products 3.1.1](#)
- [Universal Products 3.2.0](#)
- [Universal Products 4.1.0](#)
- [Stonebranch Solutions 4.2.0](#)
- [Stonebranch Solutions 4.3.0](#)
- [Workload Automation 5.1.0](#)

### Overview

This page provides processes for the removal of current (5.1.0) and previous (Stonebranch Solutions 4.3.0 or earlier) releases of Workload Automation for IBM i.

If you remove the currently installed release before installing or re-installing a 5.1.0 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 5.1.0 release, **UNVCMDFREF** is not deleted.

To keep your command reference files from being deleted, rename **UNVCMDFREF** and change the library owner. Release 5.1.0 will let you specify this library – or any library – as your command reference library via the `CMD_REFERENCE_DIRECTORY` option.

Before removing a 5.1.0 release, check `CMD_REFERENCE_DIRECTORY` in the Universal Command server member (**UCMDS**) of the Workload Automation configuration file (**UNVCONF**) – located in the product library (**UNVPRD510**); 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 5.1.0 release.



#### Warning

If the **UCHGRSL** command was used to rename the 510-tagged names to normal production names (see [UCHGRSL \(Change Release Tag\) Program](#)), you must perform the following steps prior to uninstalling either 5.1.0, 4.3.0, 4.2.0, 4.1.0, 3.2.0, 3.1.1, or 1.2.1:

<b>Step 1</b>	Run the <b>UCHGRSL</b> command to temporarily rename the 5.1.0 Workload Automation commands in <b>QSYS</b> to the 510-tagged names and the tagged names of the older product to the normal production names. Without this step, Workload Automation may no longer run as expected.
<b>Step 2</b>	After removal of the older product is complete, use the <b>*NEW</b> option with the <b>UCHGRSL</b> command to change the 510-tagged names back to the normal, production names.

### Universal Encrypt 1.2.1



#### Note

Before performing this uninstall, read the [Overview](#).

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:

<b>Step 1</b>	To remove the licensed program, execute the following command: <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <pre>====&gt;DLTLICPGM LICPGM(OUEN121)</pre> </div>
---------------	---

<b>Step 2</b>	If the following message displays, use the <b>DSPJOBLOG</b> command to identify the problem: <b>Objects for product OUBEN121 option *ALL release *ONLY not deleted.</b> Correct the problem and repeat Step 1.
---------------	---

## Universal Command 1.2.1



**Note**  
Before performing this uninstall, read the [Overview](#).

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:

<b>Step 1</b>	If it is active, end the Universal Broker subsystem ( <b>UBROKER</b> ). On the IBM i command line, execute the following command:
	<pre>===&gt;ENDSBS UBROKER</pre>
<b>Step 2</b>	To remove the licensed program, execute the following command:
	<pre>===&gt;DLTLICPGM LICPGM(OUCM121)</pre>
<b>Step 3</b>	If the following message displays, use the <b>DSPJOBLOG</b> command to identify the problem: <b>Objects for product OUCM121 option *ALL release *ONLY not deleted.</b> Correct the problem and repeat Step 2.

## Universal Products 3.1.1



**Note**  
Before performing this uninstall, read the [Overview](#).

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:

<b>Step 1</b>	If it is active, end the Universal Broker subsystem ( <b>UBROKER</b> ). On the IBM i command line, execute the following command:
	<pre>===&gt;ENDSBS UBROKER</pre>
<b>Step 2</b>	To remove the licensed program, execute the following command:
	<pre>===&gt;DLTLICPGM LICPGM(OUNV311)</pre>
<b>Step 3</b>	If the following message displays, use the <b>DSPJOBLOG</b> command to identify the problem: <b>Objects for product OUNV311 option *ALL release *ONLY not deleted.</b> Correct the problem and repeat Step 2.

## Universal Products 3.2.0



### Note

Before performing this uninstall, read the [Overview](#).

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:

<b>Step 1</b>	<p>If it is active, end the Universal Broker subsystem (<b>UNVUBR320</b>). On the IBM i command line, execute the following command:</p> <pre>===&gt;ENDSBS UNVUBR320</pre>
<b>Step 2</b>	<p>To remove the licensed program, execute the following command:</p> <pre>===&gt;DLTLICPGM LICPGM(OUNV320)</pre>
<b>Step 3</b>	<p>If the following message displays, use the <b>DSPJOBLOG</b> command to identify the problem: <b>Objects for product OUNV320 option *ALL release *ONLY not deleted.</b> Correct the problem and repeat Step 2.</p>

## Universal Products 4.1.0



### Note

Before performing this uninstall, read the [Overview](#).

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:

<b>Step 1</b>	<p>If it is active, end the Universal Broker subsystem (<b>UNVUBR410</b>). On the IBM i command line, execute the following command:</p> <pre>===&gt;ENDSBS UNVUBR410</pre>
<b>Step 2</b>	<p>To remove the licensed program, execute the following command:</p> <pre>===&gt;DLTLICPGM LICPGM(OUNV410)</pre>
<b>Step 3</b>	<p>If the following message displays, use the <b>DSPJOBLOG</b> command to identify the problem: <b>{Objects for product OUNV410 option *ALL release *ONLY not deleted.}}</b> Correct the problem and repeat Step 2.</p>

## Stonebranch Solutions 4.2.0



### Note

Before performing this uninstall, read the [Overview](#).

The user ID used for uninstalling Stonebranch Solutions 4.2.0 must have \*ALLOBJ and \*SECADM special authorities.

To uninstall Stonebranch Solutions 4.2.0, perform the following steps:

<b>Step 1</b>	<p>If it is active, end the Universal Broker subsystem (<b>UNVUBR420</b>). On the IBM i command line, execute the following command:</p> <pre>===&gt;ENDSBS UNVUBR420</pre>
<b>Step 2</b>	<p>To remove the licensed program, execute the following command:</p> <pre>===&gt;DLTLICPGM LICPGM(OUNV420)</pre>
<b>Step 3</b>	<p>If the following message displays, use the <b>DSPJOBLOG</b> command to identify the problem: <b>Objects for product OUNV510 option *ALL release *ONLY not deleted.</b> Correct the problem and repeat Step 2.</p>

## Stonebranch Solutions 4.3.0



### Note

Before performing this uninstall, read the [Overview](#).

The user ID used for uninstalling Stonebranch Solutions 4.3.0 must have \*ALLOBJ and \*SECADM special authorities.

To uninstall Stonebranch Solutions 4.3.0, perform the following steps:

<b>Step 1</b>	<p>If it is active, end the Universal Broker subsystem (<b>UNVUBR430</b>). On the IBM i command line, execute the following command:</p> <pre>===&gt;ENDSBS UNVUBR430</pre>
<b>Step 2</b>	<p>To remove the licensed program, execute the following command:</p> <pre>===&gt;DLTLICPGM LICPGM(OUNV430)</pre>
<b>Step 3</b>	<p>If the following message displays, use the <b>DSPJOBLOG</b> command to identify the problem: <b>Objects for product OUNV430 option *ALL release *ONLY not deleted.</b> Correct the problem and repeat Step 2.</p>

## Workload Automation 5.1.0



### Note

Before performing this uninstall, read the [Overview](#).

The user ID used for uninstalling Workload Automation 5.1.0 must have \*ALLOBJ and \*SECADM special authorities.

To uninstall Workload Automation 5.1.0, perform the following steps:

<b>Step 1</b>	If it is active, end the Universal Broker subsystem ( <b>UNVUBR510</b> ). On the IBM i command line, execute the following command:  <pre>===&gt;ENDSBS UNVUBR510</pre>
<b>Step 1</b>	To remove the licensed program, execute the following command:  <pre>===&gt;DLTLICPGM LICPGM(OUNV510)</pre>
<b>Step 1</b>	If the following message displays, use the <b>DSPJOBLOG</b> command to identify the problem: <b>Objects for product OUNV510 option *ALL release *ONLY not deleted.</b> Correct the problem and repeat Step 2.

## IBM i Installation - Customization

- Workload Automation for IBM i - Customization
- Modify the IPL Start-up Program
- Multiple-Installation Implementation
  - Universal Broker
  - Active Jobs
- Universal Broker Customization
  - Configuration
  - System Initialization
- Universal Command Manager Customization
  - Configuration
- Universal Command Server Customization
  - Configuration
- Universal Control Manager Customization
  - Configuration
- Universal Control Server Customization
  - Configuration
- Universal Data Mover Manager Customization
  - Configuration
- Universal Data Mover Server Customization
  - Configuration
- Universal Query Customization
  - Configuration
- National Language Customization

## Workload Automation for IBM i – Customization

(For information on applying product licenses to installed Workload Automation 5 for IBM i components, see [IBM i Installation - Licensing](#).)

### Modify the IPL Start-up Program

Modify the IPL Start-up Program so that the Universal Broker subsystem (**UNVUBR510**) is started at IPL time.

The subsystem start command is:

```
QSYS/STRSBS UNVPRD510/UNVUBR510
```

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.

### Multiple-Installation Implementation

Workload Automation for IBM i 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 Workload Automation commands in the **QSYS** library are tagged with the \*v\*ersion / \*r\*elease / \*m\*odification number, **510**. This renaming allows currently installed pre-5.1.0 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 5.1.0 command names and tag the pre-5.1.0 command names with that pre-5.1.0 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, **UNVPRD510**.

The **UNVUBR510** subsystem and Workload Automation job descriptions use the **UNVPRD510 / UBROKER** class and job queue instead of the **QGPL / QBATCH** class and job queue.

To revert back to the **QGPL / QBATCH** class:

<b>Step 1</b>	Invoke the CHGRTGE command using SBSD( <b>UNVPRD510 / UNVUBR510</b> ), SEQNBR( <b>10</b> ), and CLS( <b>QGPL / QBATCH</b> ).
<b>Step 2</b>	Invoke the CHGPJE command using SBSD( <b>UNVPRD510 / UNVUBR510</b> ), PGM( <b>QSYS / QP0ZSPWP</b> ), JOB( <b>UCMSINIT</b> ), and CLS( <b>QGPL / QBATCH</b> ).
<b>Step 3</b>	Invoke the CHGPJE command using SBSD( <b>UNVPRD510 / UNVUBR510</b> ), PGM( <b>QSYS / QP0ZSPWT</b> ), JOB( <b>UNVSRV</b> ), and CLS( <b>QGPL / QBATCH</b> ).
<b>Step 4</b>	Use the RMVJOBQE command with SBSD( <b>UNVPRD510 / UNVUBR510</b> ) and JOBQ( <b>UNVPRD510 / UBROKER</b> ) to remove the job queue entry from the <b>UNVUBR510</b> subsystem.

To revert back to the **QGPL/QBATCH** job queue, invoke the CHGJOB command specifying JOBQ(**QGPL / QBATCH**) for the **UNVPRD510 / UBROKER**, **UNVPRD510 / UCMSINIT**, and **UNVPRD510 / 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 **UNVUBR510 / UBROKER** job queue has no limit. Thus, it allows many additional Workload Automation 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 **UNVUBR510** 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.

## Universal Broker Customization

### Configuration

Configuration options for Universal Broker are stored in configuration file **UNVCONF(UBROKER)**, in product library **UNVPRD510**.

See the [Universal Broker 5.1.0 Reference 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 (**UNVUBR510**) located in product library **UNVPRD510**.

To start Universal Broker, enter the following on the command line: **STRSBS UNVPRD510/UNVUBR510**. There are two sets of pre-start jobs initiated: **UNVSRV** and **UCMSINIT**:

- **UNVSRV** is the pre-start job for any of the Workload Automation 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.

## Universal Command Manager Customization

## Configuration

Configuration options for Universal Command Manager are stored in its configuration file, member **UNVCONF(UCMD)** in product library **UNVPRD510**.

See the [Universal Command 5.1.0 Reference Guide](#) for details on configuring Universal Command Manager.

The file can be edited using the Source Entry Utility (SEU).

## Universal Command Server Customization

### Configuration

Configuration options for Universal Command Server are stored in configuration file **UNVCONF(UCMDS)**, in product library **UNVPRD510**.

See the [Universal Command 5.1.0 Reference 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 **UNVPRD510**.

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 **UNVPRD510**. See the [Indesca 5.1.0 User Guide](#) for information on how Universal Command Server utilizes the UACL file.

The file can be edited using the Source Entry Utility (SEU).

## Universal Control Manager Customization

### Configuration

Configuration options for Universal Control Manager are stored in configuration file **UNVCONF(UCTL)**, in product library **UNVPRD510**.

See the [Workload Automation Utilities 5.1.0 Reference Guide](#) for details on configuring Universal Control Manager.

The file can be edited using the Source Entry Utility (SEU).

## Universal Control Server Customization

### Configuration

Configuration options for Universal Control Server are stored in configuration file **UNVCONF(UCTLS)**, in product library **UNVPRD510**.

See the [Workload Automation Utilities 5.1.0 Reference 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 **UNVPRD510** 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 **UNVPRD510**. See the [Indesca 5.1.0 User Guide](#) for information on how Universal Control Server utilizes the UACL file.

The file can be edited using the Source Entry Utility (SEU).

## Universal Data Mover Manager Customization

### Configuration

Configuration options for Universal Data Mover Manager are stored in its configuration file, member **UNVCONF(UDM)** in product library **UNVPRD510**.

See the [Universal Data Mover 5.1.0 Reference Guide](#) for details on configuring Universal Data Mover Manager.

The file can be edited using the Source Entry Utility (SEU).

## Universal Data Mover Server Customization

### Configuration

Configuration options for Universal Data Mover Server are stored in configuration file **UNVCONF(UDMS)**, in product library **UNVPRD510**.

See the [Universal Data Mover 5.1.0 Reference Guide](#) for details on configuring Universal Data Mover Server.

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 **UNVPRD510**.

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 **UNVPRD510**. See the [Infitran 5.1.0 User Guide](#) for information on how Universal Data Mover Server utilizes the UACL file.

The file can be edited using the Source Entry Utility (SEU).

## Universal Query Customization

### Configuration

Configuration options for Universal Query are stored in configuration file **UNVCONF(UQRY)**, in product library\***UNVPRD510\***.

See the [Workload Automation Utilities 5.1.0 Reference Guide](#) for details on configuring Universal Query.

The file can be edited using the Source Entry Utility (SEU).

## National Language Customization

The default IBM i 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, Workload Automation 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 Workload Automation software cannot recognize. As a result, Workload Automation components, including Universal Broker, fail to start.

To resolve this, either:

- Use a Universal Broker job CCSID of 65535. Both the Workload Automation 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.
- Use 037 as the CCSID for the Universal Broker user profile (**UNVUBR510**). 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 IBM i 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.

## IBM i Installation - Licensing

- Licensing Workload Automation 5 for IBM i Components
- Product License File
  - Format
  - Sample
- Entering License Information
- Restart Universal Broker

### Licensing Workload Automation 5 for IBM i Components

After Workload Automation 5 for IBM i has been installed, you must configure the following components with product licenses before they can be used:

- Universal Command Manager
- Universal Data Mover Manager

### Product License File

Product license information (license parameter keywords and their values) is contained in a text file provided by your Stonebranch, Inc. account representative.

#### Format

The format of the product license file name is: *<component name>\_<customer name>\_<operating system>\_<schedule or solution>.txt*. For example: **Indesca\_Stonebranch\_OS400\_A1.txt**.

(For Universal Command Manager, **Indesca** is used as the *<component name>* in the product license file name, and as the name of the product in the product license file itself; for Universal Data Mover Manager, **Infitran** is used as the *<component name>* in the product license file name, and as the name of the product in the product license file itself - see below.)

#### Sample

The following is a sample Universal Command Manager for IBM i product license file:

```
License_Product "INDESCA"
License_Customer "STONEBRANCH"
License_OS_Type "OS400"
License_Type "PERPETUAL"
License_Expiration_Date 2029.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

### Entering License Information

In the **UNVPRD510** product library:

- Enter the Universal Command Manager license parameters into the Universal Command Manager configuration file, member **UNVCONF(UCMD)**.
- Enter the Universal Data Mover Manager license parameters into the Universal Data Mover Manager configuration file, member **UNVCONF(UDM)**.

It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)

### Restart Universal Broker

For Universal Broker to read the license information, you must stop and restart it:

<b>Stop Universal Broker</b>	<pre>ENDSBS UBROKER *CNTRLD</pre> <p>(*CNTRLD performs a controlled shutdown.)</p>
<b>Start Universal Broker</b>	<pre>STRSBS UBROKER</pre>

## IBM i Installation - Object Inventory Lists

- Workload Automation for IBM i Objects
- Installation Libraries
  - QSYS Library Command Names
- Source File Record Lengths
- Product Library (UNVPRD510)
- Temporary Library (UNVTMP510)
- Spool Library (UNVSPL510)
- Command Reference Library (UNVCMDREF)
- QSYS Library

### Workload Automation for IBM i Objects

The Workload Automation for IBM i 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

### Installation Libraries

These objects are contained in four user-designated installation libraries:

- Product library **UNVPRD510** (formerly **UNIVERSAL**)  
Library containing all Workload Automation for IBM i products.
- Temporary library **UNVTMP510** (formerly **UNVTMP**)  
Library used as a temporary work area by Workload Automation. It may contain such items as trace files and temporary job log files.
- Spool library **UNVSPL510** (formerly **UNVSPPOOL**)  
Library containing a spool database of all Workload Automation 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 IBM i **QSYS** library:

- Copy of the Workload Automation for IBM i commands contained in **UNVPRD510**.
- Universal Broker user profile, **UNVUBR510**.

### QSYS Library Command Names

The names of the commands in **QSYS** are tagged with the Workload Automation for IBM i version / release / modification number, **510**. However, since command names can have a maximum nine characters, the following names are shortened:

- **UDSPLOGF** becomes **UDSPLF510**.
- **USBMJOB** becomes **USBMJ510**.
- **UMSGHNDLR** becomes **UMSGH510**.
- **UJOBINIT** becomes **UJOB510**.



#### Note

For further information on library and object names, see [Naming Conventions](#).

### Source File Record Lengths

Workload Automation for IBM i 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 Workload Automation installation library (**UNVPRD510**, by default) – are editable by SEU as they are initially installed.

**Product Library (UNVPRD510)**

<b>Name</b>	<b>Type</b>	<b>Description</b>
<b>PRCDSPLOGF</b>	*PGM	Universal Display Log File program.
<b>PUMSGHNDLR</b>	*PGM	Universal Submit Job, WTO msg callback program.
<b>STRUCP</b>	*PGM	UCOPY command program.
<b>STRUEN</b>	*PGM	Universal Encrypt command program.
<b>STRUME</b>	*PGM	UMET command program.
<b>UBROKER</b>	*PGM	Universal Broker program.
<b>UCMCP</b>	*PGM	UCMD Manager command processing program.
<b>UCHGRS</b>	*PGM	Universal Change Release program.
<b>UCMD</b>	*PGM	UCMD Manager program.
<b>UCMDEF01</b>	*PGM	UCMD object definition program.
<b>UCMDEXIT01</b>	*PGM	UCMD install exit program.
<b>UCMMSG01</b>	*PGM	UCMD message definition program.
<b>UCMSINIT</b>	*PGM	UCMD Server Initiator program.
<b>UCMSJOBI</b>	*PGM	UCMD Server Initiator job initialization program.
<b>UCMSJOB</b>	*PGM	UCMD Server Initiator job termination program.
<b>UCMSRV</b>	*PGM	Universal Command Server program.
<b>UCOPY</b>	*PGM	Universal Copy program.
<b>UCTCPP</b>	*PGM	UCTL Manager command processing program.

<b>UCTL</b>	*PGM	Universal Control Manager program.
<b>UCTSRV</b>	*PGM	Universal Command Server program.
<b>UDM</b>	*PGM	Universal Data Mover Manager program.
<b>UDMCP</b>	*PGM	UDM Manager command processing program.
<b>UDMSRV</b>	*PGM	UDM Server program.
<b>UDSPLOGF</b>	*PGM	Universal Display Log File cmd processing program.
<b>UENCRYPT</b>	*PGM	Universal Encrypt program.
<b>UJOBINIT</b>	*PGM	USBMJOB initialization program.
<b>ULSTSECPP</b>	*PGM	Universal Spool List utility command processing program.
<b>UMET</b>	*PGM	Universal Message Translator program.
<b>UMSGHDLR</b>	*PGM	Universal Submit Job, PUMSGHDLR interface.
<b>UQRCP</b>	*PGM	Universal Query command program.
<b>UQUERY</b>	*PGM	Universal Query program.
<b>URMVSECPP</b>	*PGM	Universal Spool Remove utility command processing program.
<b>USBMJOB</b>	*PGM	Universal Submit Job program.
<b>USBMUSRJOB</b>	*PGM	Universal Submit User Job program.
<b>USLIST</b>	*PGM	Universal Spool List Job program.
<b>USLRM</b>	*PGM	Universal Spool Remove Job program.
<b>UCMINST</b>	*MSGF	UCMD install message file.
<b>UNVMSG</b>	*MSGF	Workload Automation message file.



<b>UNVSEQMSGF</b>	*MSGF	Universal Products Sequential Message File.
<b>USBMJOB</b>	*MSGF	Universal Submit Job message file.
<b>CP2CCSID_X</b>	*FILE	Example code page to CCSID mapping file.
<b>INSTL.INFO</b>	*FILE	Installation Information file for library locations.
<b>UCMNLSTMP</b>	*FILE	Temporary National Language Support file.
<b>UCMSVRPRTF</b>	*FILE	Universal Command server job log print file.
<b>UNVCLSRC</b>	*FILE	CL source.
<b>UNVCOMP</b>	*FILE	Universal component definition members.
<b>UNVCONF</b>	*FILE	Universal configuration members.
<b>UNVDDSSRC</b>	*FILE	DDS Src for Stonebranch Products.
<b>UNVNLS</b>	*FILE	National Language Support.
<b>UNVTMPL</b>	*FILE	Universal template members.
<b>USBMJPRTF</b>	*FILE	USBMJOB job log print file.
<b>UBROKER</b>	*JOBDEF	Job Definition for Universal Broker.
<b>UCMSINIT</b>	*JOBDEF	Job Definition for UCMD Server Initiator.
<b>UNVSRV</b>	*JOBDEF	Job Definition for Universal Servers.
<b>UBROKER</b>	*JOBQ	Universal Broker job queue.
<b>UBROKER</b>	*CLS	Universal Broker class.
<b>STRUCM</b>	*CMD	Universal Command Manager command.
<b>STRUCP</b>	*CMD	Universal Copy command.

<b>STRUCT</b>	*CMD	Universal Control Manager command.
<b>STRUDM</b>	*CMD	Universal Data Mover Manager command.
<b>STRUEN</b>	*CMD	Universal Encrypt command.
<b>STRUME</b>	*CMD	Universal Message Translator command.
<b>STRUQR</b>	*CMD	Universal Query command.
<b>UCHGRLS</b>	*CMD	Universal Change Release command.
<b>UDSPLOGF</b>	*CMD	Universal Display Log File command.
<b>UJOBINIT</b>	*CMD	USBMJOB initialization program command.
<b>ULSTSE</b>	*CMD	Universal Spool List utility command.
<b>UMSGHNDLR</b>	*CMD	WTO message callback program command.
<b>URMVSE</b>	*CMD	Universal Spool Remove utility command.
<b>USBMJOB</b>	*CMD	Universal Submit Job command.
<b>UNVUBR510</b>	*SBSD	Subsystem Definition for Universal Broker
<b>STRUCM</b>	*PNLGRP	UCMD Manager panel definition.
<b>STRUCP</b>	*PNLGRP	UCOPY panel definition.
<b>STRUCT</b>	*PNLGRP	UCTL Manager panel definition.
<b>STRUDM</b>	*PNLGRP	UDM Manager panel definition.
<b>STRUEN</b>	*PNLGRP	Universal Encrypt panel definition.
<b>STRUME</b>	*PNLGRP	UMET panel definition.

<b>STRUQR</b>	*PNLGRP	Universal Query panel definition.
<b>UCHGRLS</b>	*PNLGRP	Universal Change Release panel definition.
<b>UDSPLOGF</b>	*PNLGRP	Universal Display Log File panel definition.
<b>UJOBINIT</b>	*PNLGRP	USBMJOB initialization program panel definition.
<b>ULSTSE</b>	*PNLGRP	Universal Spool List utility panel definition.
<b>UMSGHNDLR</b>	*PNLGRP	WTO message callback program panel definition.
<b>URMVSE</b>	*PNLGRP	Universal Spool Remove utility panel definition.
<b>USBMJOB</b>	*PNLGRP	Universal Submit Job panel definition.
<b>UNV510</b>	*PRDDFN	Workload Automation product definition.
<b>UNV510</b>	*PRDLOD	Workload Automation product load.

### Temporary Library (UNVTMP510)

Name	Type	Description
<b>UCMDEF02</b>	*PGM	UNVTMP object definition program.
<b>UCMDEXIT02</b>	*PGM	UNVTMP install exit program.

### Spool Library (UNVSPL510)

Name	Type	Description
<b>UNVDEFSP</b>	*PGM	<b>UNVSPL510</b> object definition program.
<b>UNVEXITSPL</b>	*PGM	<b>UNVSPL510</b> install exit program.

### Command Reference Library (UNVCMDREF)

Name	Type	Description

<b>UNVCRFSRC</b>	*FILE	Universal Command command reference file.
------------------	-------	---

## QSYS Library

Name	Type	Description
<b>STRUCM510</b>	*CMD	Universal Command Manager command.
<b>STRUCP510</b>	*CMD	Universal Copy command.
<b>STRUCT510</b>	*CMD	Universal Control Manager command.
<b>STRUDM510</b>	*CMD	Universal Data Mover Manager command.
<b>STRUEN510</b>	*CMD	Universal Encrypt command.
<b>STRUME510</b>	*CMD	Universal Message Translator command.
<b>STRUQR510</b>	*CMD	Universal Query command.
<b>UDSPLF510</b>	*CMD	Universal Display Log File command.
<b>UJOBI510</b>	*CMD	Universal Job Initializer command.
<b>ULSTSE510</b>	*CMD	Universal Spool List utility command.
<b>UMSGH510</b>	*CMD	Universal Message Handler command.
<b>URMVSE510</b>	*CMD	Universal Spool Remove utility command.
<b>USBMJ510</b>	*CMD	Universal Submit Job command.
<b>UNVUBR510</b>	*USRPRF	Universal Broker user profile.

## HP NonStop Installation

Error formatting macro: redirect: java.lang.NullPointerException

## HP NonStop Installation - Overview



Currently, HP NonStop runs Universal Command 2.1.1. These pages provide information for that version.

### HP NonStop Installation

These pages provide information on the installation of Stonebranch, Inc.'s Universal Command 2.1.1 on the HP NonStop operating system.

All Universal Products 2.1.1 for HP NonStop (see [HP NonStop Components](#)) are included as part of Universal Command 2.1.1.

### Installation Summary

<b>Step 1</b>	Download the distribution file from the Stonebranch <a href="#">Customer Portal</a> .
<b>Step 2</b>	Extract the HP NonStop installation files from the windows self-extracting executable.
<b>Step 3</b>	Transfer the extracted files to the HP NonStop Guardian environment using ftp.
<b>Step 4</b>	Logon to the Guardian environment of the HP NonStop system as super.super.
<b>Step 5</b>	Alter the file code of the installation script and set its file code to 180.
<b>Step 6</b>	Convert the installation script to a Guardian EDIT style file.
<b>Step 7</b>	Run the installation obey script.

## HP NonStop Installation - HP NonStop Components

### Components

The Universal Command 2.1.1 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

### Product Compatibility

The following table 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.

## HP NonStop Installation - Installation Requirements

- System Requirements
- Platform Requirements
- \$SYSTEM Volume
  - Log Directory
  - Trace Directory

### System Requirements

System requirements for Universal Command 2.1.1 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 Universal Command 2.1.1, you must be able to write to the directory from which the installation is launched.

### Platform Requirements

Since platform requirements may change with new releases of a product, please consult the [Platform Support for Opswise Automation Center 5.1.1](#) and [Indesca-Infitrans 5.1.0](#) page to make sure that your platform is supported before performing an installation.

### \$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.

## HP NonStop Installation - Distribution File

- HP NonStop Distribution File
- Obtaining the Distribution File
- NSK S-series System
  - Extracting the Installation Files
- Integrity System
  - Extracting the Installation Files
- Distribution File Contents

### HP NonStop Distribution File

The product distribution file contains the installation files required for the installation of Universal Command 2.1.1 for HP NonStop.

### Obtaining the Distribution File

To obtain the distribution file, you must download it from the Stonebranch [Customer Portal](#).



#### Note

A customer user name and password – provided by Stonebranch, Inc. – are required to access the Customer Portal.

Stonebranch, Inc. provides a separate product distribution file for the following HP NonStop systems:

- NSK S-series system
- Integrity system

### NSK S-series System

The Universal Command 2.1.1 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.

### Integrity System

The Universal Command 2.1.1 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.

## Distribution File Contents

The following table 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
<b>Readme.unv</b>	Summary of the installation procedure.
<b>ucmdins</b>	Installation script.
<b>ucmdpkg</b>	Installation package file.

If your Universal Command 2.1.1 for HP NonStop distribution file does not contain these files, contact Stonebranch, Inc. Customer Support to obtain a correct distribution file.

## HP NonStop Installation - Installation Procedures

### 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:

<b>Step 1</b>	Transfer the installation files (in a BINARY transfer mode, not ASCII) to the HP NonStop operating system.
<b>Step 2</b>	<p>To start the install as user <b>super.super</b>, the installation script file (<b>ucmdins</b>) must be converted to a Guardian EDIT file by using the <b>ctoedit</b> function.</p> <p>Execute the following commands from the subvolume in which the files are located:</p> <pre data-bbox="263 625 1443 709">fup alter ucmdins, code 180</pre> <pre data-bbox="263 785 1443 869">ctoedit ucmdins, ucmdinst</pre> <p>This prepares the script to be used with the <b>obey</b> command.</p>
<b>Step 3</b>	<p>To start the installation, execute the following <b>obey</b> command:</p> <pre data-bbox="263 1037 1443 1121">obey ucmdinst</pre> <p>This copies the files to the proper locations, after which the system will be ready to configure.</p>

## HP NonStop Installation - Customization

- Introduction
- Universal Broker Customization
  - Configuration
  - System Initialization
- Universal Command Manager Customization
  - Configuration
- Universal Command Server Customization
  - Configuration
- Universal Control Manager Customization
  - Configuration
- Universal Control Server Customization
  - Configuration
- Universal Query Customization
  - Configuration

### Introduction

This page provides information on customizing the Universal Products components that comprise Universal Command 2.1.1 for HP NonStop.

(For information on applying product licenses to installed Universal Command 2.1.1 for HP NonStop components, see [HP NonStop Installation - Licensing](#).)

### Universal Broker Customization

#### Configuration

Configuration options for Universal Broker are stored in configuration file, **UBRCFG**, in subvolume **\$SYSTEM.UNVCONF**.

See the [Universal Broker 5.1.0 Reference 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.

#### TZ Environment Variable Values

Value	Description
<b>North America</b>	
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
<b>Europe</b>	

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

**Note**

Other common time zone abbreviations can be used; all possible values are not shown.

## 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 [Indesca 5.1.0 User Guide](#) for details on the Universal Broker daemon script.

## 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 5.1.0 Reference Guide](#) for details on configuring Universal Command Manager.

The product executable files intended for command line use are located in the subvolume **\$\$SYSTEM.UNVBIN**.

## 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 5.1.0 Reference 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 5.1.0 Reference Guide](#) for information on how Universal Command Server utilizes the UACL file.

## Universal Control Manager Customization

### Configuration

Configuration options for Universal Control Manager are stored in configuration file, **UCTLCFG**, in subvolume **\$\$SYSTEM.UNVCONF**.

See the [Workload Automation Utilities 5.1.0 Reference Guide](#) for details on configuring Universal Control Manager.

The product executable files intended for command line use are located in the subvolume **\$\$SYSTEM.UNVBIN**.

## Universal Control Server Customization

### Configuration

Configuration options for Universal Control Server are stored in configuration file, **UCTLSCFG**, in subvolume **\$\$SYSTEM.UNVCONF**.

See the [Workload Automation Utilities 5.1.0 Reference 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 [Indesca 5.1.0 User Guide](#) for information on how Universal Control Server utilizes the UACL file.

## Universal Query Customization

### Configuration

Configuration options for Universal Query are stored in configuration file, **UQRYCFG**, in subvolume **\$\$SYSTEM.UNVCONF**.

See the [Workload Automation Utilities 5.1.0 Reference Guide](#) for details on configuring Universal Query.

The product executable files intended for command line use are located in the subvolume **\$\$SYSTEM.UNVBIN**.

## HP NonStop Installation - Licensing

- Licensing Workload Automation 5 for HP NonStop Components
- Product License File
  - Format
  - Sample
- Entering License Information

### Licensing Workload Automation 5 for HP NonStop Components

After Universal Command 2.1.1 for HP NonStop has been installed, you must configure the following component with a product license before it can be used:

- Universal Command Manager

#### Product License File

Product license information (license parameter keywords and their values) is contained in a text file provided by your Stonebranch, Inc. account representative.

#### Format

The format of the product license file name is: *<component name>\_<customer name>\_<operating system>\_<schedule or solution>.txt*. For example: **Indesca\_Stonebranch\_Tandem\_A1.txt**.

(For Universal Command Manager, **Indesca** is used as the *<component name>* in the product license file name, and as the name of the product in the product license file itself - see below.)

#### Sample

The following is a sample Universal Command Manager for IBM i product license file:

```
License_Product "INDESCA"
License_Customer "STONEBRANCH"
License_OS_Type "Tandem"
License_Type "PERPETUAL"
License_Expiration_Date 2029.12.31          YYYY.MM.DD
License_NT_Servers 100
License_UNIX_Servers 100
License_OS400_Servers 10000
License_OS390_Servers 10000
License_Tandem_Servers 10000
License_OS390_Unix_Servers 10000
License_Key ABCD-1234-EFGH-5678-IJKL-MNOP-9999
```

### Entering License Information

In subvolume **\$\$SYSTEM.UNVCONF**, enter the Universal Command Manager license parameters into the Universal Command Manager configuration file, member **UCMDCFG**

It is recommended that you enter license information at the end of the file. (The values are specified in the same syntax as all other configuration options.)



#### Note

You do not have to restart Universal Broker in order for it to read the license information.



## HP NonStop Installation - File Inventory Lists

- Universal Command for HP NonStop - File Inventory Lists
- Universal Broker
- Universal Command Manager
- Universal Command Server
- Universal Control Manager
- Universal Control Server
- Universal Query

### 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
<b>\$\$SYSTEM.UNVBIN.UBROKERD</b>	Broker daemon start script.
<b>\$\$SYSTEM.UNVBIN.UBROKER</b>	Console broker program.
<b>\$\$SYSTEM.UNVBIN.UBRD</b>	Daemon broker program.
<b>\$\$SYSTEM.UNVCOMP</b>	Component definition directory.
<b>\$\$SYSTEM.UNVCOMP.UCMD</b>	Universal Command Server component definition file.
<b>\$\$SYSTEM.UNVCOMP.UCTL</b>	Universal Control Server component definition file.
<b>\$\$SYSTEM.UNVNLS</b>	Code page files used for text translation between different operating systems and platforms and product message catalogs.
<b>\$\$SYSTEM.UNVLOG</b>	Broker message log directory.
<b>\$\$SYSTEM.UNVTRACE</b>	Broker trace file directory.
<b>\$\$SYSTEM.UNVCONF.UBRCFG</b>	Broker configuration file.

## Universal Command Manager

File	Description
<b>\$\$SYSTEM.UNVBIN.UCMD</b>	Universal Command Manager program.
<b>\$\$SYSTEM.UNVBIN.UENCRYPT</b>	Universal Encrypt program file.
<b>\$\$SYSTEM.UNVNLS</b>	Code page files used for text translation between different operating systems and platforms and product message catalogs.
<b>\$\$SYSTEM.UNVCONF.UCMDCFG</b>	Manager configuration file.

## Universal Command Server

File	Description
<b>\$\$SYSTEM.UNVBIN.UCMSRV</b>	Server component program.
<b>\$\$SYSTEM.UNVBIN.UCOPY</b>	Utility used for binary file copies. Similar to the UNIX cat command.
<b>\$\$SYSTEM.UNVBIN.UMET</b>	Universal Message Translator program.
<b>\$\$SYSTEM.UNVNLS</b>	Code page files used for text translation between different operating systems and platforms and product message catalogs.
<b>\$\$SYSTEM.UNVTRACE</b>	Server trace file directory.
<b>\$\$SYSTEM.UNVCONF.UCMDSCFG</b>	Server configuration file.

## Universal Control Manager

File	Description
<b>\$\$SYSTEM.UNVBIN.UCTL</b>	Manager program.
<b>\$\$SYSTEM.UNVNLS</b>	Code page files used for text translation between different operating systems and platforms and product message catalogs.
<b>\$\$SYSTEM.UNVCONF.UCTLCFG</b>	Manager configuration file.

## Universal Control Server

File	Description
<b>\$\$SYSTEM.UNVBIN.UCTSRV</b>	Server component program.
<b>\$\$SYSTEM.UNVNLS</b>	Code page files used for text translation between different operating systems and platforms and product message catalogs.
<b>\$\$SYSTEM.UNVTRACE</b>	Server trace file directory.
<b>\$\$SYSTEM.UNVCONF.UCTLSCFG</b>	Server configuration file.

## Universal Query

File	Description
<b>\$\$SYSTEM.UNVBIN.UQUERY</b>	Universal Query program.
<b>\$\$SYSTEM.UNVNLS</b>	Code page files used for text translation between different operating systems and platforms and product message catalogs.
<b>\$\$SYSTEM.UNVCONF.UQRYCFG</b>	Query configuration file.

## Workload Automation 5 Administration

Error formatting macro: redirect: java.lang.NullPointerException

## **Workload Automation 5 Administration - Overview**

### **Workload Automation 5 Administration**

The following information is provided for the administration of Workload Automation 5 components:

- [Workload Automation 5 - Databases](#)
- [Workload Automation 5 - Security](#)

## Workload Automation 5 - Databases

Error formatting macro: redirect: java.lang.NullPointerException

## Databases - Overview

### Workload Automation 5 Databases

Some Workload Automation 5 components provide features that rely upon a set of databases for their implementation. Such features include fault tolerance, managed configuration, event subsystem (UES) data collection, and event monitoring.

Unless otherwise noted, the Universal Broker owns all databases and performs all direct database access. The Universal Broker processes and responds to all database access requests it receives from individual Workload Automation 5 components.

## Databases - Component Information Database

- [Component Information Database](#)
- [Universal Spool](#)

### Component Information Database

The component information database records information about all Workload Automation 5 server components that the Universal Broker manages. It is opened during Universal Broker start-up processing.

The information captured by the Universal Broker includes, but is not limited to, the component's process ID, start time, current state, and end time.

One important aspect of this database is its ability to record the current state of an Workload Automation 5 server component. Each time a component's state changes, it sends a notification to Universal Broker, which updates that component's record.

For the Workload Automation 5 components that offer it, this component state provides the basis for reconnect or restart functionality, otherwise known as network or manager fault tolerance (for Indesca: see [Indesca - Fault Tolerance Implementation](#) or network fault tolerance (for Infitran: see [Infitran - Fault Tolerance Implementation](#)).

When a Workload Automation 5 server process finishes executing and its component state indicates that it has completed, Universal Broker deletes that component's information from the database.

Universal Broker stores Workload Automation 5 component information in the **bcomponent.db** and **scomponent.db** database files.

<b>IBM i</b>	The database, <b>UBR_CMP_DB</b> , is located in the spool library, <b>UNVSPL510</b> .
<b>UNIX</b>	The database file resides in the <b>/var/opt/universal/spool</b> directory.
<b>Windows</b>	The database file default location is: <ul style="list-style-type: none"> <li>• <b>C:\Program Files\Universal</b> (32-bit Windows systems)</li> <li>• <b>C:\Program Files\Universal</b> (64-bit Windows systems)</li> <li>• <b>C:\Program Files (x86)\Universal</b> (32-bit installs on 64-bit Windows systems)</li> </ul>
<b>z/OS</b>	Workload Automation 5 components access this file via an HFS- or zFS-allocated dataset, which is mounted on the z/OS Unix System Services (USS) file system. Universal Broker is capable of dynamically mounting this database during start-up, if it is not already mounted.

### Universal Spool

The Universal Spool – or more simply, the spool – supports storage of the standard file I/O that Universal Command Server redirects from user processes. When Universal Command Server executes with spooling enabled, it captures a user process' standard input, standard output, and standard error and writes them to the spool.

Universal Spool is implemented as a set of databases. Universal Broker and Universal Command Server remove the database records automatically when they are no longer required. No database maintenance jobs are required.

Universal Broker and server components are the only programs that access the spool. No user access is required. The operating system's file system security should be used to prevent all access to the spool except for the broker and server. The Broker and server require full control permissions to the spool in order to add, delete, update, and read database files.

All standard I/O files written to the spool are encrypted to insure data privacy.

The spool database files that store redirected standard input, standard output, and standard error have the format **spool.stdin.compид.db**, **spool.stdout.compид.db**, and **spool.stderr.compид.db**, respectively. In each of the names, **compид** is replaced by the server's actual component ID.



<b>IBM i</b>	<p>These files are located in the spool library, <b>UNVSPL510</b>, in the following format:</p> <ul style="list-style-type: none"><li>• <b>SExxxxxxx</b> (SE = standard error)</li><li>• <b>SOxxxxxxx</b> (SO = standard output)</li><li>• <b>SIxxxxxxx</b> (SI = standard input)</li></ul> <p>In each file name, <b>xxxxxxx</b> is the component ID expressed in hexadecimal. (The component ID is the Component ID used to identify a Stonebranch process. This is the same Component ID output by Universal Query.)</p>
<b>UNIX</b>	<p>These files reside in the <b>/var/opt/universal/spool</b> directory.</p>
<b>Windows</b>	<p>The database files' default location is:</p> <ul style="list-style-type: none"><li>• <b>C:\Program Files\Universal</b> (32-bit Windows systems)</li><li>• <b>C:\Program Files\Universal</b> (64-bit Windows systems)</li><li>• <b>C:\Program Files (x86)\Universal</b> (32-bit installs on 64-bit Windows systems)</li></ul>
<b>z/OS</b>	<p>Workload Automation 5 components access these files via an HFS- or zFS-allocated dataset, which is mounted on the z/OS Unix System Services (USS) file system. Universal Broker is capable of dynamically mounting this database during start-up, if it is not already mounted.</p>

## Databases - Universal Event Monitor Databases

Error formatting macro: redirect: java.lang.NullPointerException

## Universal Event Monitor Databases - Overview

### Universal Event Monitor Databases

To support the Universal Event Monitor (UEM) component, the Universal Broker provides and manages the following databases:

- [Event Definition Database](#)
- [Event Handler Database](#)
- [Event Spool Database](#)

These database files are local to each system. The Workload Automation 5 install script is responsible for creating the database directory. If the Universal Broker attempts to open a database file that does not exist, it will create that database.

<b>UNIX</b>	The default database directory is <code>/var/opt/universal/spool</code> .
<b>Windows</b>	The default UEM database directory is: <ul style="list-style-type: none"><li>• <code>C:\Program Files\Universal</code> (32-bit Windows systems)</li><li>• <code>C:\Program Files\Universal</code> (64-bit Windows systems)</li><li>• <code>C:\Program Files (x86)\Universal</code> (32-bit installs on 64-bit Windows systems)</li></ul>

**Note**

UEM Server is only available for Windows and UNIX. The UEM databases are used only on those operating systems.

## Universal Event Monitor Event Definition Database

### Event Definition Database

The Universal Event Monitor (UEM) Server stores information about the events that it monitors in the event definition database.

An event definition record describes a system event and provides the information that UEM uses to track an event occurrence and test for its completion. An event definition record also may contain information that UEM uses to respond to (that is, "handle") an event's successful completion, its failure to complete, and even its failure to occur.

An event-driven UEM Server relies upon stored event definitions for its input. When an event-driven UEM Server starts, it asks the Broker for all event definitions assigned to it. If no event definitions are assigned to a particular event-driven Server, that Server continues to execute but will not do any actual event monitoring.

A demand-driven UEM Server also may obtain its input from a stored event definition record, but it is not required. Typically, a demand-driven Server receives its input from the UEM Manager's command line parameters.

Event definition records are added and maintained with the [UEMLoad Utility](#).

## Universal Event Monitor Event Handler Database

### Event Handler Database

An event handler record describes the action that Universal Event Monitor (UEM) should take in response to a monitored event's outcome. This action, or response, is simply a system command or script that UEM executes upon an event's completion, failure to complete, or failure to occur. The UEM Server executes these processes in a secure context, using user account credentials stored in the event handler record.

**A Stonebranch Tip**

Security is a primary concern within all Workload Automation components.

Whenever the user account information stored in an event handler record includes a password, that password is encrypted using the Data Encryption Standard (DES) algorithm.

An event-driven UEM Server relies upon stored event handlers to determine its response to the events that it monitors. The event definition records that describe events to UEM also contain the IDs of event handler records that UEM should use to respond to those events.

A demand-driven UEM Server also may respond to an event using a stored event handler record, but it is not required. Typically, a demand-driven Server relies upon the UEM Manager's command line parameters to describe the actions that it should take in response to the event that it monitors.

When a UEM Server needs to use a stored event handler record, it sends a request to the Universal Broker to retrieve the record using the ID specified in the event definition record or provided from the UEM Manager command line. The Universal Broker returns the event handler record to the UEM Server, which then executes the specified system command or script.

Event handler records are added and maintained with the [UEMLoad Utility](#).

## Universal Event Monitor Event Spool Database

### Event Spool Database

Universal Event Monitor (UEM) records its monitoring activity in the event spool database.

It is possible for UEM to detect multiple occurrences of any single event that it monitors. UEM creates a record in the spool database for each event occurrence that it detects and tracks. UEM maintains the current state of an event occurrence from initial detection through the completion of any event handlers.

If an event definition goes inactive before UEM detects any occurrences of that event, UEM creates a single spool entry to record the expired event.

Universal Broker applies all updates to the event spool database. A UEM Server is responsible for sending the Universal Broker all relevant information, along with the required database operation (add, update, or delete).

Typically, any spool records created for an event are deleted when the Broker detects the completion of the UEM Server that monitored the event. However, when an event-driven UEM Server completes, any records that indicate work in progress (for example, tracking of an event occurrence, execution of an event handler) are retained for possible recovery when the event-driven Server is restarted. For additional information on recovery of event spool records, see [Universal Event Monitor Server](#).



#### A Stonebranch Tip

An option can be set in the Universal Broker's configuration to prevent it from deleting any event spool records when the UEM Server component completes. Setting the **comp\_info\_retention** option to a value greater than 0 causes the event spool record to be preserved.

Because there is currently no database cleanup routine available, this option should be set only following a recommendation from, and with the assistance of, Stonebranch Inc. Customer Support.

Feedback from a demand-driven UEM Server is returned to the UEM Manager that initiated the monitoring request. In this situation, event spool records are simply another means of following the progress of the event and any detected occurrences.

However, for an event-driven UEM Server that has no client, the records in the event spool database are the best way to monitor the status of the work performed by that UEM Server. Because an event-driven UEM Server typically is a long-running process, an adequate history of the UEM Server's activity can be obtained by viewing the spool records.

Currently, event spool records can only be viewed with the Universal Spool List utility (**uslist**). Information on using Universal Spool List to view event spool records can be found in [Universal Event Monitor Server](#). For information on all Workload Automation 5 utilities, see the [Workload Automation Utilities 5.1.0 Reference Guide](#).

## Controlling Universal Event Monitor Database Access

### Controlling Universal Event Monitor Database Access

Universal Broker is responsible primarily for providing access to the Workload Automation databases. However, there are utilities provided, including the Universal Spool List (**uslist**) and Universal Spool Remove (**uslrm**), that can be used to access the databases directly. While these utilities should be used only following a recommendation from and with the assistance of Stonebranch, Inc. Customer Support, they are documented fully in the [Workload Automation Utilities 5.1.0 Reference Guide](#).

To protect the database contents, operating system permissions on the database files themselves should be set so that only accounts with super-user or administrative privileges has access to them.

UEM provides its own command line utility, UEMLoad, to maintain the event definition and event handler databases. While the contents of these databases can be viewed using the Universal Spool List utility, it is recommended that all access be done using UEMLoad. The ability to remove event definition and event handler records is only provided with UEMLoad.

UEMLoad only can manage event definition and event handler databases that are local to the system on which it resides. To process a request, UEMLoad sends a request to the Universal Broker running on that system to start a demand-driven UEM Server. Next, UEMLoad sends the database request to the UEM Server, so that the UEM Server can validate the request and provide any required default values. The UEM Server then forwards the request to the Universal Broker, so that the changes can be applied to the appropriate database.

UEMLoad executes in the security context of the user account that started it. Since the Universal Broker applies changes to the event definition and event handler databases, any user with the authority to execute UEMLoad will, effectively, have access to a secure resource. It is therefore strongly recommended that the privileges on UEMLoad be set such that only those user accounts with super-user or administrative privileges be allowed to execute it.

Application support also is provided to further limit access to the event definition and event handler databases. A type of Universal Access Control List (UACL) is provided by UEM to grant or deny local user accounts the authority to access these databases.

To fully secure the event definition and event handler databases, a UACL entry can be defined to deny access to all user accounts. Then, additional entries can be defined to grant database access to those user accounts with the appropriate authority. Whenever UEMLoad is executed, the entries in the UACL will be checked. If a match cannot be found which indicates that the user account that started UEMLoad is allowed to access the database, the application will terminate with an error.

[Universal Access Control List \(UACL\)](#) provides a more thorough overview of the UACL feature. For information on the specific UACL used to control access to the event definition and event handler databases, see the [DATABASE\\_MAINTENANCE\\_ACL](#) UACL entry in the [Universal Event Monitor 5.1.0 Reference Guide](#).

The event spool records generated by a UEM Server only can be viewed with the Universal Spool List utility.

## Universal Enterprise Controller Databases

- [Overview](#)
- [Database Files](#)
- [Database Management](#)
  - [Automated Database Cleanup](#)
  - [Memory Management](#)

### Overview

Universal Enterprise Controller (UEC) uses databases to maintain agent, user, configuration, and event data.

### Database Files

The UEC databases reside in three files:

Database Name	File Name	Contents
UEC database	uec.db	Definitions of agents, groups, users, SAP systems, and a record of updates to distributed components' configurations in a managed environment.
UEC transient database	uec.tmp.db	UES monitor events and component information that is temporary to support I-Activity Monitor. This file is deleted and created upon restart of UEC.
UES database	uec.evm.db	Universal Event Subsystem (UES) persistent events.

### Database Management

#### Automated Database Cleanup

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

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

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

The following UEC configuration options control database record retention:

- `COMMIT_COMPLETE_EXPIRATION`
- `COMMIT_INCOMPLETE_EXPIRATION`
- `MONITOR_EVENT_EXPIRATION`
- `PERSISTENT_EVENT_EXPIRATION`

#### Memory Management

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

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

```
UNV5101D 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:

<b>Step 1</b>	For z/OS installations, mount the <b>UECDB</b> HFS or zFS dataset.
<b>Step 2</b>	Inside the UEC database directory (or, on z/OS, the mount point), create a text file named <b>DB_CONFIG</b> .
<b>Step 3</b>	Inside the <b>DB_CONFIG</b> file, add the following string: <pre>set_tmp_dir *dbpath*</pre> In this string, <b>dbpath</b> is the path to the location in which the database files reside.



<b>Step 4</b>	Start / restart UEC.
---------------	----------------------

## Database Backup and Recovery

- [Overview](#)
- [Database Backups](#)
- [General Database Recovery Procedures](#)

### Overview

Workload Automation 5 databases, on operating system's other than IBM i, are implemented using Oracle's Berkeley Database product.

Recovering from database corruption requires the following steps:

1. Dump the corrupted database to a file using the [Universal Database Dump](#) utility.
2. Reload the database from the dump file using the [Universal Database Load](#) utility.

Database corruption can occur if the system or address space that is managing the databases ends abnormally. A Workload Automation program that utilizes databases should not be terminated abnormally.

Abnormal methods of termination include:

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

### Database Backups

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

### General Database Recovery Procedures

Generally speaking, database recovery follows the same steps independent of platform and database file.

Multiple attempts may be necessary in order to successfully recover from database corruption. Stonebranch Inc. recommends that you begin with the least aggressive recovery method and only proceed to more aggressive methods if necessary.

For the first recovery attempt, execute the [Universal Database Dump](#) utility with the **-r** (lowercase) command line option. This option instructs the utility to recover as much data as possible. Depending on the extent of database corruption, this may result in a recovered database with some incomplete key/data pairs.

Reload the database using the [Universal Database Load](#) utility, and specify the **-o** option. This option instructs the utility to remove the underlying database file, which results in a clean reload from the dump file.

If the database passes validation when you restart the application, it is likely that all data was successfully recovered and no additional recovery attempts are necessary.

If the database fails validation, rerun the [Universal Database Dump](#) utility and omit the **-r** option. This results in a dump of only the most complete data. While this improves the chances for successful recovery, some data loss is likely. Rerun the [Universal Database Load](#) utility and restart the application.

If both recovery attempts fail, you may delete the corrupted database and restart the application. This results in a total loss of data, but will allow the application to execute. The application will create the missing database during startup.

## Database Recovery for Universal Broker

- [Database Recovery for Universal Broker](#)

### Database Recovery for Universal Broker

Universal Broker uses databases to maintain component information, configuration information, and event data. A corrupted database will prevent the Broker from executing.

Database recovery procedures depend partly on the operating system on which the Broker is executing. This page describes the procedures for each operating system.

<b>IBM i</b>	<p>The Universal Broker subsystem, <b>UNVUBR510</b> (by default), must be down in order to perform database recovery. Use standard IBM i database recovery procedures and attempt to restart the Universal Broker subsystem.</p> <p>If the problem persists, restore the failing database file. The entire Universal Spool file library may be required if restoring individual files fails to correct the problem. As a last resort, delete all files in the Universal Spool file library and restart <b>UNVUBR510</b>.</p> <p>Deleting the files from the Universal Spool library will result in loss of all data stored in those files, including spooled output for Manager Fault Tolerant jobs. All affected jobs may need to be re-run.</p>
<b>UNIX</b>	<p>The Broker daemon must be down to perform database recovery. A backup of either the database file being recovered or the entire directory should be created before recovery is attempted.</p> <p>A sample database recovery script is provided in file <b>ubrdbrec</b> in the <code>/opt/universal/ubroker/bin</code> directory. The script uses the Universal Database Utilities to dump and reload a database file. The default location of all Universal Broker databases is the <code>/var/opt/universal/spool</code> directory.</p> <p>The user ID with which the recovery script runs requires appropriate permissions to the database directory and to the database file. Write access is required to the directory and read and write access is required to the database file.</p> <p>The <b>ubrdbrec</b> script accepts an optional argument: the database file name to recover. If no database file name is specified, the <b>ues.db</b> database is recovered. The script ends with exit code 0 if successful and a non-zero exit code if it failed.</p>
<b>Windows</b>	<p>The Broker service must be stopped to perform database recovery. A backup of either the database file being recovered or the entire directory should be created before recovery is attempted.</p> <p>A sample database recovery batch file is provided in file <b>ubrdbrec.bat</b> in the "<code>Program Files\Universal\UBroker\bin</code>" directory. The batch file uses the Universal Database Utilities to dump and reload a database file.</p> <p>The default location of all Universal Broker databases is directory "<code>Program Files\Universal\spool\ubroker</code>".</p> <p>The user ID with which the recovery script runs requires appropriate permissions to the database directory and to the database file. Write access is required to the directory and read and write access is required to the database file.</p> <p>The <b>ubrdbrec.bat</b> batch file accepts an optional argument: the database file name to recover. If no database file name is specified, the <b>ues.db</b> database is recovered. The batch file ends with exit code 0 if successful and a non-zero exit code if it failed.</p>
<b>z/OS</b>	<p>The Universal Broker started task must be down to perform database recovery. A backup of either the database file being recovered or the entire HFS or zFS data set should be created before recovery is attempted.</p> <p>A sample database recovery job is provided in member <b>UBRDBREC</b> in the <b>SUNVSAMP</b> library. The job uses the Universal Database Utilities to dump and reload a database file.</p> <p>All databases are located in the HFS or zFS product data set <b>#HLQ.UNV.UNVDB</b>. The HFS or zFS data set must be mounted prior to running <b>UBRDBREC</b>. See <a href="#">Mounting and Unmounting the Databases</a> for information on mounting the HFS or zFS data set, if necessary.</p> <p>The user ID with which the recovery job runs requires appropriate permissions to the root directory of the HFS or zFS data set and to the database file. Write access is required to the directory and read and write access is required to the database file.</p> <p>Customize <b>UBRDBREC</b> to meet local JCL and installation requirements. Specify the database file name to recover on the PARM keyword of the <b>EXEC</b> statement of both steps (the dump and load steps). When all modifications are complete, submit the job. All steps should end with return code 0.</p>

## Database Recovery for Universal Enterprise Controller

- Database Recovery for Universal Enterprise Controller


### Database Recovery for Universal Enterprise Controller

If Universal Enterprise Controller (UEC) terminates abnormally, it creates the file **uec.hf** in the database directory, which prompts UEC to initiate database verification upon restart.

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

The Universal Database Dump (**UDBDUMP**) utility and the Universal Database Load (**UDBLOAD**) utility enable recovery from a corrupted Berkeley database. (For detailed information on these utilities, see the [Workload Automation Utilities 5.1.0 Reference Guide](#).)

Database recovery procedures depend partly on the operating system on which UEC is executing: z/OS or Windows. This page describes the procedures for each operating system.

<p><b>Windows</b></p>	<p>The UEC service must be stopped to perform database recovery. A backup of either the database file being recovered or the entire directory should be created before recovery is attempted.</p> <p>A sample database recovery batch file is provided in file <b>uecdbrec.bat</b> in the "<b>Program Files\Universal\UECtlr\bin</b>" directory. The batch file uses the Universal Database Utilities to dump and reload a database file.</p> <p>The default location of all UEC databases is "<b>Program Files\Universal\UECtlr</b>".</p> <div data-bbox="293 989 1446 1220" style="background-color: #ffffcc; padding: 10px; margin: 10px 0;"> <p> <b>Note</b>            Stonebranch has identified an issue with upgrades <i>from</i> releases earlier than UEC 3.2.0.0 (such as 3.1.0.x or 3.1.1.x) <i>to</i> releases 3.2.0.0 and later. Following the upgrade, UEC databases reside in the location specified by the user's currently configured <b>working_directory</b> location. This location defaults to "<b>Program Files\Universal\UECtlr\bin</b>". If the current UEC install was not an upgrade, it may be necessary to pass the path to the <b>uec_evm.db</b> file as a command line argument to the script. You can provide an absolute path or a path relative to the <b>uecdbrec.bat</b> script's location.</p> </div> <p>The user ID with which the recovery script runs requires appropriate permissions to the database directory and to the database file. Write access is required to the directory and read and write access is required to the database file.</p> <p>The <b>uecdbrec.bat</b> batch file accepts an optional argument-the database file name to recover. If no database file name is specified, the <b>uec_evm.db</b> database is recovered. The batch file ends with exit code 0 if successful and a non-zero exit code if it failed.</p>
<p><b>z/OS</b></p>	<p>The UEC started task must be down to perform database recovery. A backup of either the database file being recovered or the entire HFS or zFS data set should be created before recovery is attempted.</p> <p>A sample database recovery job is provided in member <b>UECDBREC</b> in the <b>SUNVSAMP</b> library. The job uses the Universal Database Utilities to dump and reload a database file.</p> <p>All databases are located in the HFS or zFS product data set <b>#HLQ.UNV.UECDB</b>. The HFS data set is allocated to the <b>UNVDB</b> ddname in both the dump and load steps. The HFS or zFS data set must be mounted prior to running <b>UECDBREC</b>. See <a href="#">Mounting and Unmounting the Databases</a> for additional information on mounting the HFS or zFS data set.</p> <p>The user ID with which the recovery job runs requires appropriate permissions to the root directory of the HFS data set and to the database file. Write access is required to the directory and read and write access is required to the database file.</p> <p>Customize <b>UECDBREC</b> to meet local JCL and installation requirements. All UEC databases are recovered by the job. When all modifications are complete, submit the job. All steps should end with return code 0.</p>

## Listing Workload Automation 5 Database Records Examples

### Listing Workload Automation 5 Database Records Examples

- [List Universal Broker Database](#)
- [List Universal Command Server Database Records](#)
- [List Universal Broker Detail for a Component](#)
- [List Standard Out for a Component](#)

## List Universal Broker Database

### List Universal Broker Database

The following figures illustrate how to execute Universal Spool List (USLIST) with all defaults. (No options are required to issue USLIST.)

<b>UNIX</b>	<pre>cd /opt/universal/bin uslist</pre>
<b>Windows</b>	<pre>cd c:\program files\universal\uspool\bin uslist</pre>

Since no USLIST options are supplied, this example, by default, lists the contents of the Universal Broker Component database (UBROKER). A summary of all records is produced; no detail component records are written.

The broker database must be located in the default directory.

### Components

[Universal Spool List](#)

## List Universal Command Server Database Records

### List Universal Command Server Database Records

The following figures illustrate how to list the database records for a specific component (in this case, Universal Command Server).

<b>UNIX</b>	<pre>cd /opt/universal/bin uslist -list ucmd -ucmdspooldir "c:\program files\universal\spool2"</pre>
<b>Windows</b>	<pre>cd c:\program files\universal\uspool\bin uslist -list ucmd -ucmdspooldir "c:\program files\universal\spool2"</pre>

These examples list the contents of the Universal Command Server Component database. A summary of all records is written.

These examples specify the directory location in which the Universal Command Server Component database is located. If the directory contains spaces, it must be enclosed in double ( " ) quotes.

### Command Line Options

The command line options used in this example are:

Option	Description
-list	Type of database from which to select record to write.
-ucmdspooldir	Directory location in which the Universal Command Server Component database ( <b>scomponent.db</b> ) is located.

### Components

Universal Spool List

## List Universal Broker Detail for a Component

- [List Universal Broker Detail for a Component](#)
  - [Command Line Options](#)
  - [Components](#)

### List Universal Broker Detail for a Component

The following figures illustrate how to list the Universal Broker detail for a specific component ID.

<b>UNIX</b>	<pre style="border: 1px solid #0070C0; padding: 5px;">cd /opt/universal/bin uslist -component 123456789</pre>
<b>Windows</b>	<pre style="border: 1px solid #0070C0; padding: 5px;">cd c:\program files\universal\uspool\bin uslist -component 123456789</pre>

Since the **-list** option is not supplied, these examples, by default, list the contents of the Universal Broker Component database (UBROKER). Because a component ID is specified, this will cause detail broker records to be written for component ID **123456789**.

### Command Line Options

The command line option used in this example is:

Option	Description
-component	Component identifier for which records will be selected to write.

### Components

[Universal Spool List](#)



## List Standard Out for a Component

- [List Standard Out for a Component](#)
  - [Command Line Options](#)
  - [Components](#)

### List Standard Out for a Component

The following figures illustrate how to list the standard output spool file for a specific component ID.

<b>UNIX</b>	<pre>cd /opt/universal/bin uslist -list STDOUT -component 123456789</pre>
<b>Windows</b>	<pre>cd c:\program files\universal\uspool\bin uslist -list STDOUT -component 123456789</pre>

The standard output spool file is written for component **123456789**.

### Command Line Options

The command line options used in this example are:

Option	Description
-list	Type of database from which to select records to write.
-component	Component identifier for which records will be selected to write.

### Components

[Universal Spool List](#)

## Removing Workload Automation 5 Database Records Examples

### Removing Workload Automation 5 Database Records Examples

- [Remove Component Records](#)
- [Remove Component Records - Change Database Directory](#)

## Remove Component Records

- [Remove Component Records](#)
  - [Command Line Options](#)
  - [Components](#)

### Remove Component Records

The following figures illustrate how to execute Universal Spool Remove (USLRM) with defaults.

<b>UNIX</b>	<pre style="border: 1px solid #4a7ebb; padding: 5px;">cd /opt/universal/bin uslrm -component 123456789</pre>
<b>Windows</b>	<pre style="border: 1px solid #4a7ebb; padding: 5px;">cd c:\program files\universal\uspoolbin uslrm -component 123456789</pre>

The only required option is **-component** (the component ID; you can execute Universal Spool List (USLIST) utility to find specific component IDs. All Workload Automation database records will be removed for component **123456789**.

### Command Line Options

The command line options used in this example are:

Command Options	Description
-component	Component identifier for which records will be removed.

### Components

[Universal Spool Remove](#)

## Remove Component Records - Change Database Directory

- [Remove Component Records - Change Broker Database Directory](#)
  - [Command Line Options](#)
  - [Components](#)

### Remove Component Records - Change Broker Database Directory

The following figure illustrate how to execute Universal Spool Remove (USLRM) and specify a Universal Broker database directory other than the default.

<b>UNIX</b>	<pre>cd /opt/universal/bin uslrm -component 123456789 -brokerspooldir "c:\program files\universal\spool2"</pre>
<b>Windows</b>	<pre>cd c:\program files\universal\uspool\bin uslrm -component 123456789 -brokerspooldir "c:\program files\universal\spool2"</pre>

All Workload Automation database records will be removed.

The **-brokerspooldir** option specifies the directory location in which the Universal Broker Component database is located. If the directory has spaces, it must be enclosed within double ( " ) quotation marks.

### Command Line Options

The command line options used in this example are:

Command Options	Description
-component	Component identifier for which records will be removed.
-brokerspooldir	Directory location in which the Universal Broker Component database ( <b>bcomponent.db</b> ) is located.

### Components

Universal Spool Remove

## **Workload Automation 5 - Security**

Error formatting macro: redirect: java.lang.NullPointerException

## Security Overview

### Workload Automation 5 Security

Workload Automation 5 Security is comprised of the following elements:

- [Component security](#)
- [Universal Access Control Lists](#)
- [X.509 Certificates](#)

## Security of Workload Automation 5 Components

Error formatting macro: redirect: java.lang.NullPointerException

## Security of Workload Automation 5 Components - Overview

### Security of Workload Automation 5 Components

Each component of Workload Automation 5 is designed to be a secure system.

As the level of security rises, so does the administrative complexity of the system. Workload Automation 5 balances the two, minimizing administrative complexity without sacrificing security.

These pages identify the security features inherent in the design for each Workload Automation 5 component:

- Universal Broker
- Universal Command Manager
- Universal Command Server
- Universal Data Mover Manager
- Universal Data Mover Server
- Universal Event Monitor Manager
- Universal Event Monitor Server
- Universal Control Manager
- Universal Control Server
- Universal Event Log Dump
- Universal Spool List
- Universal Spool Remove



## Security - Universal Broker

- Overview
- File Permissions
- Configuration Files
- Universal Access Control List
- Universal Broker User Account
  - Removing \*ALLOBJ Authority from UNVUBR510 User Profile
  - Removing \*SPLCTL Authority from **UNVUBR510** User Profile
  - Removing \*ALLOBJ and \*SPLCTL Authorities from **UNVUBR510** User Profile

### Overview

Universal Broker is designed to be a secure system. As the level of security rises, so does the administrative complexity of the system. Universal Broker has balanced the two to avoid the administrative complexity with a minimum sacrifice to security.

Universal Broker security concerns are:

1. Access to Universal Broker files, directories, and configuration options.
2. Account with which Universal Broker executes.
3. Privacy and integrity of transmitted network data.

### File Permissions

At a minimum, only trusted user accounts should have write access to the Universal Broker installation data sets. This most likely means only the administrators should have write access. For maximum security, only trusted accounts should have read access to these data sets.

<b>IBM i</b>	<p>At a minimum, limit non-trusted user accounts to object authority of use to the Universal Broker product library, <b>UNVPRD510</b>; the product temporary library, <b>UNVTMP510</b>; the command reference library, <b>UNVCMDREF</b>; the universal spool library, <b>UNVSPL510</b>; and all objects within these libraries.</p> <p>For maximum security, only trusted accounts (administrators and the <b>UNVUBR510</b> profile) should have management, existence, alter, add, update, or delete authority to these objects. As a reminder, the system value <b>QCRTAUT</b> controls public access authority to created objects unless overridden by specific commands.</p> <p>User profiles that use Stonebranch products require *CHANGE authority to the <b>UNVTMP510</b> library.</p>
<b>HP NonStop</b>	<p>All files that the Broker creates or updates are located in either <b>\$SYSTEM.UNVLOG</b> or <b>\$SYSTEM.UNVTRACE</b>. The Broker does not need write access to its installation subvolume.</p>
<b>UNIX</b>	<p>All files that the Broker creates or updates are located in the <b>/var/opt/universal</b> directory. This means that the Broker does not need write access to its installation directory or subdirectories.</p> <p>Universal Broker requires full control (read, write, remove, and add) of the <b>/var/opt/universal</b> directory and its subdirectories. The Broker creates spool files, trace files, and log files in this directory. Only the account used to execute the Broker requires full access to this directory.</p> <p>The Broker configuration options can be changed to use directories other than <b>/var/opt/universal</b>. If this is the case, the same permissions must be set up for these specified directories.</p>
<b>Windows</b>	<p>Universal Broker requires write access to its primary install directory (that is, <b>.\Universal\UBroker</b>), which serves as its default trace file location.</p> <p>The Broker requires full control over the database directory (that is, <b>.\Universal\spool</b>), along with all subdirectories and files under that location.</p> <p>When the Broker executes as a console application with its message destination set to <b>logfile</b>, the Broker requires full control over the <b>.\Universal\Broker\log</b> directory and all <b>.log</b> files within it. The Universal Broker Windows service always writes its message to the Windows event log, which means that it requires no write access to a log directory or any other of its installation subdirectories and files.</p>
<b>z/OS</b>	<p>Universal Broker requires update access to its database files, which exist as HFS- or zFS-allocated datasets mounted on the z/OS Unix System Services (z/OS USS) file system. The Broker accesses HFS-allocated datasets using the <b>UNVDB</b> and <b>UNVSPOOL</b> ddnames in its STC JCL. The Broker accesses zFS-allocated datasets via its <b>UNIX_DB_DATA_SET</b> and <b>UNIX_SPOOL_DATA_SET</b> configuration options.</p>

### Configuration Files

Only trusted user accounts should have write, create, or delete access to the Broker configuration files or any of the directories in the configuration file directory search list.



#### Windows

Although you can edit Workload Automation 5 configuration files with any text editor (for example, Notepad), we recommend using the [Universal Configuration Manager Control Panel](#) application set configuration options.

The Universal Configuration Manager provides a graphical interface and context-sensitive help, and helps protect the integrity of the configuration file by validating all changes to configuration option values (see [Universal Configuration Manager](#)). It also directs the Broker to refresh its cache of Indesca component configuration settings, making it unnecessary to issue a separate configuration REFRESH request via the Universal Control utility.

## Universal Access Control List

Universal Broker uses the Universal Access Control List (UACL) as an extra layer of security. The UACL contains entries (that is, rules) that permit or deny access to the Universal Broker (see [Security - Universal Access Control List \(UACL\)](#) for details).

Universal Broker reads the UACL entries when the program is started. If the UACL file is changed, the new entries can be activated either by:

- Stopping and starting Universal Broker.
- Sending Universal Broker a Universal Control configuration refresh request, which instructs Universal Broker to reread all of its configuration files, including the UACL file (see [Configuration Refresh](#)).



#### Windows

Although you may edit the UACL file with any text editor (for example, Notepad), we recommend that you maintain UACL entries using the Universal Configuration Manager Control Panel application (see [Universal Configuration Manager](#)). The Universal Configuration Manager sends a configuration refresh request to the Universal Broker. Updated values take effect immediately, making it unnecessary to recycle the Broker to apply UACL changes.

## Universal Broker User Account

Each Workload Automation 5 component that the Universal Broker spawns inherits the Broker's account credentials. Occasionally, components must perform privileged operations, such as establishing a process execution environment using a local user account's credentials.

On some platforms, this means that the Broker must execute with an account whose inherited credentials allow the spawned components to perform these operations. On other platforms, the Broker may be executed with a lesser-privileged user, provided that the components are configured in way that permits them to elevate their privileges when necessary.

The section contains platform-specific requirements to consider when setting the Broker's user account.

### IBM i

Universal Broker for IBM i runs with the **UNVUBR510** user profile, which is created at product installation time. Any component started by Universal Broker inherits this user profile.

By default, the **UNVUBR510** user profile has \*ALLOBJ, \*JOBCTL, and \*SPLCTL authority. Unless the user profile is modified as described in the following section, \*ALLOBJ authority is required for a component to switch its user profiles based on the request it is servicing. \*JOBCTL authority is required for internal control and should not be removed. The **UNVUBR510** user profile requires \*SPLCTL authority to provide Universal Submit Job job logs in specific, limited situations. (See the [Workload Automation Utilities 5.1.0 Reference Guide](#) for information on Universal Submit Job.)

Any other product or user should not use the **UNVUBR510** user profile. By default, users cannot access the system with the **UNVUBR510** profile.

#### *Removing \*ALLOBJ Authority from UNVUBR510 User Profile*

Given the extensive authority allowed by \*ALLOBJ special authority, it is desirable to avoid its use when possible. As of PTF OUC0126 for V1R2M1, it is possible to remove \*ALLOBJ special authority from the **UNVUBR510** user profile. However, by removing \*ALLOBJ from the **UNVUBR510** user profile, the administrative complexity is increased.

The following steps are required to use Universal Command with \*ALLOBJ special authority removed from the **UNVUBR510** user profile.

1. If the following objects do not have \*USE Public Authority, the **UNVUBR510** user profile must be given \*USE authority:

- QSYS/QSYGETPH
- QSYS/QWTSETP
- QSYS/QWCRJBST
- QSYS/QUSRMBRD

This can be accomplished with the following command:

```
====> EDTOBJAUT OBJ(QSYS/object_name) OBJTYPE(*PGM)
```

From the resulting screen, use F6 to add user **UBROKER** and give it \*USE authority.

2. **UNVUBR510** user profile must be given \*USE authority to the user profile objects of all user profiles that will be using the universal command server on the IBM i.

This can be accomplished with the following command:

```
====> EDTOBJAUT OBJ(QSYS/user_profile_name) OBJTYPE(*USRPRF)
```

From the resulting screen, use F6 to add user **UBROKER** and give it \*USE authority.

3. Use the following command to remove the **UNVUBR510** user profile \*ALLOBJ authority:

```
====> CHGUSRPRF USRPRF(UNVUBR510) SPCAUT(*JOBCTL *SPLCTL)
```

#### **Removing \*SPLCTL Authority from UNVUBR510 User Profile**

Use the following command to remove the **UNVUBR510** user profile \*SPLCTL authority:

```
====> CHGUSRPRF USRPRF(UNVUBR510) SPCAUT(*JOBCTL *ALLOBJ)
```

#### **Removing \*ALLOBJ and \*SPLCTL Authorities from UNVUBR510 User Profile**

Use the following command to remove all special authority from the **UNVUBR510** user profile:



```
====> CHGUSRPRF USRPRF(UNVUBR510) SPCAUT(*JOBCTL)
```

(Please refer to the previous two sections for additional information.)

#### **HP NonStop**

Universal Broker itself does not require **super.super** privileges. For example, Universal Command (UCMD) Server may require **super.super** authority. Since the component inherits its user ID from the Broker, either the Broker must be running as **super.super** or the UCMD Server program must be owned by **super.super** and **ProgID** must be set for the server program file.

If the Broker is started as a daemon at system startup time, it is started with a user ID of **super.super**. The Broker and all its components will then have sufficient authority.

<b>UNIX</b>	<p>Although Universal Broker itself does not require super-user privileges, some Indesca server components (for example, UCMD Server and UEM Server) may require super user authority to switch execution context to another user account, initialize group membership, or perform other privileged operations.</p> <p>Since the component inherits its user ID from Universal Broker, one of the following is required:</p> <ul style="list-style-type: none"> <li>• Universal Broker must execute as <b>root</b>.</li> <li>• <b>root</b> must own the Indesca Server application file (for example, <b>ucmsrv</b> or <b>uemsrv</b>), and the Indesca Server application file must have its "set user ID on execution" bit (<b>setuid on exec</b>) set (for example, <b>chmod u+s ucmsrv</b>).</li> </ul> <p>By default, the Universal Broker is owned and started with a user ID of <b>ubroker</b>. <b>root</b> will own the server components that need super user authority and these components will have their "set user ID on execution" bit set.</p> <div data-bbox="402 478 1446 684" style="background-color: #ffffcc; padding: 10px;"> <p> <b>Note</b> Indesca server components typically only invoke the privileged operations mentioned above when that component is configured to run with security enabled (that is, its security configuration option is set to a value other than <b>none</b>). When security is disabled in an Indesca server component's configuration, that component may not attempt to invoke any privileged operations, but relies completely upon the security context it inherits from the Broker.</p> </div>
<b>Windows</b>	<p>The Universal Broker Windows service can be configured to execute with the Local System account or with a specially-configured Administrative account (see <a href="#">Windows Service</a>).</p>
<b>z/OS</b>	<p>The Universal Broker started task may execute with any OMVS user ID provided that account has read access to the <b>BPX.DAEMON</b>, <b>BPX.SUPERUSER</b>, and <b>BPX.JOBNAME</b> resources in the FACILITY class.</p> <p>The Broker user account is typically configured at install time (see <a href="#">zOS Configuration - Started Tasks</a>).</p> <div data-bbox="324 940 1446 1073" style="background-color: #ffffcc; padding: 10px;"> <p> <b>Note</b> Starting with Universal Broker 5.1.0.1, the Broker user ID no longer requires READ access to the BPX.SUPERUSER resource. See <a href="#">zOS Configuration - Started Tasks</a> for more information.</p> </div>

## Security - Universal Command Manager

- File Permissions
- RACF Protection
- Configuration Files

### File Permissions

Only trusted user accounts should have write permission to the Universal Command (UCMD) Manager installation directory, its subdirectories, and all files within those directories. This most likely means only an administration group should have write access.

<b>IBM i</b>	Only trusted user accounts and the <b>UNVUBR510</b> user profile should have write permission to the UCMD Server product library ( <b>UNVPRD510</b> ) and all files within that library.  Eligible users of UCMD require read access to the message catalogs in the <b>UNVNLS</b> file.
<b>HP NonStop</b>	Eligible users of UCMD require read access to the message catalogs in the <b>\$SYSTEM.UNVNLS</b> subvolume.
<b>UNIX</b>	Eligible users of UCMD require read access to the message catalogs (*.umc files) in the <b>nls</b> subdirectory of the Workload Automation 5 installation directory ( <b>/opt/universal</b> by default).
<b>Windows</b>	Eligible users of UCMD require read access to the message catalogs (*.umc files) in the <b>nls</b> subdirectory of the <b>.Universal</b> installation directory. These file permissions are set automatically during the installation.
<b>z/OS</b>	Only trusted user accounts should have write access to the UCMD Manager installation files. Eligible users of UCMD require read access to the national language support library ( <b>SUNVNLS</b> ), the configuration file ( <b>UNVCONF</b> ), and the load library ( <b>SUNVLOAD</b> ).

### RACF Protection

The UCMD Manager for z/OS verifies a users access to a RACF general resource profile. The resource profile controls a user's access to execute a command on a remote host with a specific remote user identity.

See the [z/OS Installation - Configuration of Security](#) for complete details on installing and administering UCMD Manager RACF profiles.

### Configuration Files

Only trusted user accounts should have write access to the UCMD Manager configuration files.



#### Windows

Although you may edit configuration files with any text editor (for example, Notepad), we recommend that you manage configuration options using the [Universal Configuration Manager](#) Control Panel application. Only user accounts in the Administrator group may execute the Universal Configuration Manager.

## Security - Universal Command Server

- File Permissions
- Configuration Files
- Universal Command Server User ID
- User Authentication
- Universal Command Server User Profile
- Universal Command Server User ID

### File Permissions

Only trusted user accounts should have write permission to the Universal Command (UCMD) Server installation directory, subdirectories, and all files within them.

<b>IBM i</b>	<p>Only administrator accounts should have write permission to the UCMD Server product library, <b>UNVPRD510</b>; the command reference library, <b>UNVCMDREF</b>; the universal spool library, <b>UNVSPL510</b> and all objects within these libraries. For maximum security, only trusted accounts (administrators and the <b>UNVUBR510</b> profile) should have management, existence, alter, add, update, and delete authority to these objects. As a reminder, the system value <b>QCRTAUT</b> controls public access authority to created objects unless overridden by specific commands.</p> <p>Other than users authorized to use Workload Automation components, the same applies to the product temporary library, <b>UNVTMP510</b>.</p>
<b>HP NonStop</b>	Only trusted user accounts should have write permission to the UCMD Server installation subvolume and all files within them.
<b>Windows</b>	<p>Only trusted user accounts should have write permission to the UCMD Server installation directory and subdirectories, and all of the files within them. This most likely means only the administrator group should have write access. Eligible users of Universal Command require read access to the message catalogs (*.umc files) in the <b>nls</b> subdirectory of the Workload Automation installation directory.</p> <p>All eligible users of UCMD require permission to create directories in the UCMD Server working directory, if security is activated. A directory named after the user ID requesting the command is created for each user. The directory is created while impersonating the user; hence, it's created using the user's security account.</p> <p>Home directories are created with permissions giving the user full control of both the directory and the files within the directory.</p>
<b>z/OS</b>	Only trusted user accounts should have write permission to the UCMD Server installation data sets. No general user access is required.

### Configuration Files

Only trusted user accounts should have write access to the UCMD Server configuration files.

<b>HP NonStop</b>	Only trusted user accounts should have write permission to the UCMD Server configuration files, and add and delete access to the subvolume in which they reside.
<b>Windows</b>	Although you may edit configuration files with any text editor (for example, Notepad), we recommend that you manage configuration options using the <a href="#">Universal Configuration Manager</a> Control Panel application. Only user accounts in the Administrator group may execute the Universal Configuration Manager.


### Universal Command Server User ID

<b>HP NonStop</b>	UCMD Server requires read access to its installation subvolume. If user security is activated, the UCMD Server requires super.super access to create processes that execute with another user's identity. The UCMD Server security identity is inherited from the Universal Broker. If the Universal Broker is running with a non-super.super user ID, the UCMD Server program must have the <b>ProgID</b> bit set and <b>super.super</b> as owner. See <a href="#">Universal Message Translator</a> for details.
<b>UNIX</b>	UCMD Server requires read access to its installation directory and its working directory (defined in the component definition). If user security is activated, the Server requires root access to create processes that execute with another user's identity. The Server security identity is inherited from the Broker. If the Broker is running with a non-root user ID, then the Server program must have the set user ID on execution permission set and root as owner. See <a href="#">Universal Message Translator</a> for details.

<b>z/OS</b>	UCMD Server for z/OS requires read access to its installation data sets and its HFS working directory (defined in the component definition).
-------------	--

## User Authentication

User authentication is the process of verifying that a user is a known and valid user. The process used by UCMD Server requires the user to provide an operating system-specific user name / ID and a password. The UCMD Server passes the name / ID and password to the operating system for verification; this is referred to as logging on the user.

<b>Windows</b>	Windows provides two primary types of log on processes: batch and interactive. A user must be given the right to log on as a batch job for them to do a batch log on. All users can do an interactive log on. See the <a href="#">LOGON_METHOD</a> option for more details.
<b>UNIX</b>	Universal Command can use three different types of user authentication methods: <ol style="list-style-type: none"> <li>1. Default authentication uses the UNIX traditional password comparison method.</li> <li>2. PAM authentication uses the PAM API to authenticate users and, optionally, process session modules. This option is available only for certain UNIX platforms.</li> <li>3. HP-UX Trusted Security uses HP-UX Trust Security APIs to authenticate users. This is available only on Hewlett Packard HP-UX and Tru64 platforms.</li> </ol>
<b>HP-UX 11.00 and later}</b>	By default, supplemental group memberships are recorded in the <code>/etc/group</code> file. However, if an <code>/etc/loggingroup</code> file exists, it governs all supplemental group memberships and effectively overrides the entries in <code>/etc/group</code> . <div style="background-color: #ffffcc; padding: 10px; margin: 10px 0;"> <p> <b>Note</b>  <code>/etc/loggingroup</code> is not required to record supplemental group membership. If <code>/etc/loggingroup</code> does not exist, <code>/etc/group</code> is sufficient to record the groups in which a user belongs.</p> </div> <p>If any Workload Automation component fails to access system resources that are secured based on supplemental group membership, make sure that the authenticated user has an entry in <code>/etc/loggingroup</code>, if that file exists. Otherwise, the default entry in <code>/etc/group</code> should be sufficient.</p> <p>For more information about <code>/etc/loggingroup</code>, please see the HP-UX system documentation.</p>
<b>IBM i</b>	If the user name and password are successfully validated by the operating system, the Initiator program ( <code>UCMSINIT</code> ) changes the current user profile to the user profile of the user ID.

## Universal Command Server User Profile

If user security is activated, the UCMD Server for IBM i requires, by default, `*ALLOBJ` authority to change user profiles. Unless modifications are made (as described in [Removing \\*ALLOBJ Authority from UNVUBR510 User Profile](#) in the IBM i section of [Universal Broker User Account in Security - Universal Broker](#)), the Server user profile, which is inherited from the Broker, requires `*ALLOBJ` authority.

## Universal Command Server User ID

UCMD Server for HP NonStop requires read access to its installation subvolume.

If user security is activated, the UCMD Server requires `super.super` access to create processes that execute with another user's identity. The UCMD Server security identity is inherited from the Universal Broker. If the Universal Broker is running with a non-`super.super` user ID, the UCMD Server program must have the `ProgID` bit set and `super.super` as owner. See [Universal Message Translator](#) for details.

## Security - Universal Data Mover Manager

- [Universal Data Mover Manager Security](#)
- [File Permissions](#)
- [Configuration Files](#)

### Universal Data Mover Manager Security


Universal Data Mover is designed to be a secure system. As the level of security rises, so does the administrative complexity of the system. Universal Data Mover has balanced the two to avoid the administrative complexity with a minimum sacrifice to security.

Universal Data Mover security concerns are:

1. Access to Universal Data Mover files and directories
2. Access to Universal Data Mover configuration files
3. Universal Data Mover user account
4. Privacy and integrity of transmitted network data
5. User authentication

### File Permissions

Only trusted user accounts should have permission to write to the Universal Data Mover installation directory and subdirectories, and all files within those directories.

<b>IBM i</b>	<p><b>Object Permissions</b></p> <p>Only administrator accounts should have write permission to the following Workload Automation libraries (and all objects within these libraries):</p> <ul style="list-style-type: none"> <li>• Installation library, UNVPRD510 (by default)</li> <li>• Product temporary library, UNVTMP510</li> <li>• Universal spool library, UNVSPL510</li> </ul> <p>For maximum security, only trusted accounts (administrators and the UNVUBR510 profile) should have management, existence, alter, add, update, and delete authority to these objects.</p> <div style="background-color: #ffffcc; padding: 10px; margin-top: 10px;"> <p> <b>Note</b> System value QCRTAUT controls public access authority to created objects unless overridden by specific commands.</p> </div>
<b>z/OS</b>	<p><b>Data Set Permissions</b></p> <p>Only trusted user accounts should have write access to the Universal Data Mover installation files. Eligible users of Universal Data Mover require read access to the national language support library SUNVNLS, the configuration file UNVCONF, and the load library SUNVLOAD.</p>

### Configuration Files

Only trusted user accounts should have write access to the Universal Data Mover Manager configuration files.



#### Windows

Although you may edit configuration files with any text editor (for example, Notepad), we recommend that you manage configuration options using the Universal Configuration Manager Control Panel application. Only user accounts in the Administrator group can execute the [Universal Configuration Manager](#).



## Security - Universal Data Mover Server

- [Universal Data Mover Manager Security](#)
- [File Permissions](#)
- [Configuration Files](#)
- [Universal Data Mover Server User ID](#)
- [Universal Data Mover Server User Profile](#)
- [User Authentication](#)

### Universal Data Mover Manager Security

Universal Data Mover Server is designed to be a secure system. As the level of security rises, so does the administrative complexity of the system. Universal Data Mover Server has balanced the two to avoid the administrative complexity with a minimum sacrifice to security.

Universal Data Mover Server security concerns are:

- Access to product data sets
- Access to Workload Automation configuration files
- Universal Broker user account
- Privacy and integrity of transmitted network data
- User authentication

### File Permissions

Only trusted user accounts should have write permission to the Universal Data Mover Server installation directory and subdirectories, and all of the files within them.

<b>IBM i</b>	<p><b>Object Permissions</b></p> <p>Only administrator accounts should have write permission to the following Workload Automation libraries (and all objects within these libraries):</p> <ul style="list-style-type: none"> <li>• Installation library, UNVPRD510 (by default)</li> <li>• Product temporary library, UNVTMP510</li> <li>• Universal spool library, UNVSPL510</li> </ul> <p>For maximum security, only trusted accounts (administrators and the UNVUBR510 user profile) should have management, existence, alter, add, update or delete authority to these objects. As a reminder, the system value QCRTAUT controls public access authority to created objects unless overridden by specific commands.</p>
<b>z/OS</b>	<p><b>Data Set Permissions</b></p> <p>Only trusted user accounts should have write permission to the Universal Data Mover Server installation data sets. No general user access is required.</p>

### Configuration Files

Only trusted user accounts should have write access to the Universal Data Mover Server configuration files.



#### Windows

Although you may edit configuration files with any text editor (for example, Notepad), we recommend that you manage configuration options using the [Universal Configuration Manager](#) Control Panel application. Only user accounts in the Administrator group can execute the Universal Configuration Manager.

### Universal Data Mover Server User ID

Universal Data Mover Server requires read access to its installation directory and its working directory (defined in the component definition).

<b>UNIX</b>	If user security is activated, the Server requires root access to create processes that execute with another user's identity. The Server security identity is inherited from the Broker. If the Broker is running with a non-root user ID, then the Server program must have the set user ID on execution permission set and root as owner.
<b>z/OS</b>	Universal Data Mover Server requires read access to its installation data sets and its HFS working directory (defined in the component definition).

### Universal Data Mover Server User Profile


If user security is activated, the UDM Server for IBM i requires, by default, \*ALLOBJ authority to switch user profiles. This \*ALLOBJ authority requirement may be removed. The UDM Server initially inherits authority from the UNVUBR510 user profile. Following the switch to the user profile, the UDM Server runs under the authority of the user initiating the data transfer.

The UNVUBR510 user profile requires \*SPLCTL authority in order to provide Universal Submit Job with job logs in specific limited situations. The \*SPLCTL authority requirement can be removed. Removing \*SPLCTL from the UNVUBR510 user profile may prevent the job log processing in limited situations.

(See [Universal Broker User Account](#) for information on removing the \*ALLOBJ and \*SPLCTL authorities.)

### User Authentication

User authentication is the process of verifying that a user is known and valid to the system. The process used by UDM Server requires the user to provide a user name / ID and a password. The UDM Server passes the name / ID and password to the operating system for verification; this is referred to as logging on the user.

<b>IBM i</b>	For IBM i, user authentication is optional. However, if security is enabled, a user name / ID and password are required in order to verify the user's credentials. With security enabled, you transfer files using a specific user's security context.
<b>UNIX</b>	<p>For UNIX, user authentication is optional. However, if security is enabled, a user name / ID and password are required in order to verify the user's credentials. With security enabled, you transfer files using a specific user's security context.</p> <p>Universal Data Mover can use three different types of user authentication methods:</p> <ol style="list-style-type: none"> <li>1. Default authentication uses the UNIX traditional password comparison method.</li> <li>2. PAM authentication uses the PAM API to authenticate users. The PAM modules, which authenticate and account, are called. This option is available only for certain UNIX platforms.</li> <li>3. HP-UX Trusted Security uses HP-UX Trust Security APIs to authenticate users. This is available only on Hewlett Packard HP-UX platforms.</li> </ol>
<b>HP-UX 11.00 and later</b>	<p>By default, supplemental group memberships are recorded in the <b>/etc/group</b> file. However, if an <b>/etc/logingroup</b> file exists, it governs all supplemental group memberships and effectively overrides the entries in <b>/etc/group</b>.</p> <div style="background-color: #ffffcc; padding: 10px; margin: 10px 0;"> <p> <b>Note</b>  <b>/etc/logingroup</b> is not required to record supplemental group membership. If <b>/etc/logingroup</b> does not exist, <b>/etc/group</b> is sufficient to record the groups in which a user belongs.</p> </div> <p>If any Workload Automation component fails to access system resources that are secured based on supplemental group membership, make sure that the authenticated user has an entry in <b>/etc/logingroup</b>, if that file exists. Otherwise, the default entry in <b>/etc/group</b> should be sufficient.</p> <p>For more information about <b>/etc/logingroup</b>, please see the HP-UX system documentation.</p>
<b>Windows</b>	For Windows, user authentication is optional. However, if security is enabled, a user name / ID and password are required in order to verify the user's credentials. (With security enabled, you transfer files using a specific user's security context.)

## Security - Universal Event Monitor Manager

- [File Permissions](#)
- [Data Privacy](#)
- [RACF Protection](#)
- [Configuration Files](#)

### File Permissions

Only trusted user accounts, which are most likely those that are members of the Administrators group, should be granted write access to the UEM Manager installation directory and subdirectories, and the files within them.

<b>UNIX</b>	Authorized users of UEM require read access to the message catalogs (*.umc files) in the <b>nls</b> subdirectory of the primary Workload Automation installation directory.
<b>Windows</b>	Authorized users of UEM require read access to the message catalogs (*.umc files), which reside in the <b>.Universal\nls</b> directory. If UEM Manager is installed on an NTFS partition, these file permissions are set automatically during the installation.
<b>z/OS</b>	Eligible users of UEM require read access to the national language support library <b>SUNVNLS</b> , the configuration file <b>UNVCONF</b> , and the load library <b>SUNVLOAD</b> .

### Data Privacy

Data transmitted from a UEM Manager across a network connection to the Universal Broker and demand-driven UEM Server is protected using features present in all Stonebranch Inc. Workload Automation components.

For more information on the steps taken to protect transferred data, see [Network Data Transmission](#).

### RACF Protection

The UEM Manager for z/OS verifies a user's access to a RACF general resource profile. The resource profile controls a user's ability to monitor an event on a remote host with a specific remote user identity.

See the [Configuration of Security](#) for complete details on installing and administering UEM Manager RACF profiles.

### Configuration Files

Only trusted user accounts should have write access to the Universal Event Monitor Manager configuration files.



#### Windows

Although you may edit configuration files with any text editor (for example, Notepad), we recommend that you manage configuration options using the [Universal Configuration Manager](#) Control Panel application. Only user accounts in the Administrator group can execute the Universal Configuration Manager.

## Security - Universal Event Monitor Server

- [Data Privacy](#)
- [File Permissions](#)
- [Configuration Files](#)
- [User Authentication](#)

### Data Privacy

Data transmitted to a UEM Server across a network connection is protected using features present in all Stonebranch Inc. Workload Automation components.

For more information on the steps taken to protected transferred data, see [Network Data Transmission](#).

### File Permissions

Only trusted user accounts should have write access to the UEM Server installation directory and subdirectories, and the files within them. Authorized users of UEM require read access to the message catalogs (\*.umc files), which reside in the **.Juniversal/nls** directory.



#### Windows

If UEM Server is installed on an NTFS partition, these file permissions are automatically set during installation.

The component definitions for demand-driven and event-driven UEM Servers include the location of a **WORKING\_DIRECTORY**. By default, this is **.Juniversal\UEMHome**.

When the **USER\_SECURITY** option is enabled, and before a demand-driven UEM Server begins monitoring an event or an event-driven UEM Server executes an event handler process, the UEM Server will create a subdirectory (if it does not already exist) for the authenticated user under this working directory.

The name of the directory matches the ID of the user account specified from the UEM Manager command line or stored in the event handler record. If a Windows domain account is used, the name of the directory is **userid.domain**, where **userid** is the user ID and **domain** is the domain name. After the directory is created, the specified user account is given ownership of it and granted full control over it.

### Configuration Files


Only trusted user accounts should have write access to the Universal Event Monitor Server configuration files.



#### Windows

Although you may edit configuration files with any text editor (for example, Notepad), we recommend that you manage configuration options using the [Universal Configuration Manager](#) Control Panel application. Only user accounts in the Administrator group can execute the Universal Configuration Manager.

### User Authentication

<p><b>UNIX</b></p>	<p>When the <code>USER_SECURITY</code> option is enabled, a demand-driven UEM Server requires the ID of a valid local user account before it will begin monitoring the event. A password also may be required, depending on the rules set up in <code>ACCESS_ACL</code>.</p> <p>Likewise, an event-driven UEM Server requires this information to be stored in an event handler record before it can execute a process on behalf of that handler. All handler processes started by UEM Server when the <code>USER_SECURITY</code> option is enabled are executed in the security context of this user account.</p> <p>UEM Server for UNIX supports three different types of user authentication methods:</p> <ol style="list-style-type: none"> <li>1. Default authentication uses the UNIX traditional password comparison method.</li> <li>2. PAM authentication uses the PAM API to authenticate users. This option is only available for certain UNIX platforms.</li> <li>3. HP-UX Trusted Security uses HP-UX Trust Security APIs to authenticate users. This is only available on Hewlett Packard HP-UX platforms.</li> </ol>
<p><b>HP-UX 11.00 and later</b></p>	<p>By default, supplemental group memberships are recorded in the <code>/etc/group</code> file. However, if an <code>/etc/logingroup</code> file exists, it governs all supplemental group memberships and effectively overrides the entries in <code>/etc/group</code>.</p> <div data-bbox="418 646 1446 774" style="background-color: #ffffcc; padding: 10px; margin: 10px 0;"> <p> <b>Note</b>  <code>/etc/logingroup</code> is not required to record supplemental group membership. If <code>/etc/logingroup</code> does not exist, <code>/etc/group</code> is sufficient to record the groups in which a user belongs.</p> </div> <p>If any Workload Automation component fails to access system resources that are secured based on supplemental group membership, make sure that the authenticated user has an entry in <code>/etc/logingroup</code>, if that file exists. Otherwise, the default entry in <code>/etc/group</code> should be sufficient.</p> <p>For more information about <code>/etc/logingroup</code>, please see the HP-UX system documentation.</p>
<p><b>Windows</b></p>	<p>When the <code>USER_SECURITY</code> option is enabled, a demand-driven UEM Server requires the ID and password of a valid local user account before it will begin monitoring the event. Likewise, an event-driven UEM Server requires this information to be stored in an event handler record before it can execute a process on behalf of that handler. All handler processes started by UEM Server when the <code>USER_SECURITY</code> option is enabled are executed in the security context of this user account.</p> <p>To allow Windows to verify the user account information, a UEM Server will attempt to log that user on to the system via a call to a Windows system function.</p> <p>Windows provides two types of logon methods: interactive and batch. Unless they have been explicitly denied the ability to do so, most user accounts can be validated with the interactive logon method. Conversely, a user account typically must be granted an additional privilege before they can be authenticated using the batch logon method. This privilege is shown in Windows as “Log on as a batch job.”</p> <p>For information on configuring UEM Server to use this logon method, see the UEM Server <code>LOGON_METHOD</code> option.</p>

## Security - Universal Control Manager

- File Permissions
- Configuration Files
- RACF Protection

### File Permissions

Only trusted user accounts should have write permission to the Universal Control Manager installation directory and subdirectories, and all of the files within them. This most likely means that only the administrator group should have write access.

Eligible users of Universal Control require read access to the message catalogs (\*.umc files) in the NLS directory.

<b>HP NonStop</b>	Eligible users of Universal Control require read access to the message catalogs in the <b>\$SYSTEM.UNVNLS</b> subvolume.
<b>Windows</b>	Eligible users of Universal Control require read access to the message catalogs (*.umc files) in the <b>nls</b> subdirectory of the Workload Automation installation directory. If Universal Control Manager is installed on an <b>NTFS</b> partition, these file permissions are set automatically during the installation.
<b>z/OS</b>	Data set permissions: Eligible users of Universal Control require read access to: <ul style="list-style-type: none"> <li>• National language support library <b>SUNVNLS</b></li> <li>• Load library <b>SUNVLOAD</b></li> </ul>

### Configuration Files

Only trusted user accounts should have write access to the Universal Control Manager configuration files.



#### Windows

Although you may edit configuration files with any text editor (for example, Notepad), we recommend that you manage configuration options using the Universal Configuration Manager Control Panel application. Only user accounts in the Administrator group may execute the [Universal Configuration Manager](#).

### RACF Protection

The Universal Control Manager for z/OS verifies a user's access to a RACF general resource profile. The resource profile controls a user's access to execute a control request on a remote host.

See the [Configuration of Security](#) for complete details on installing and administering Universal Control Manager RACF profiles.

## Security - Universal Control Server

- File Permissions
- Configuration Files
- Universal Control Server User ID
- User Authentication

### File Permissions

Only trusted user accounts should have write permission to the Universal Control Server installation directory and subdirectories, and all of the files within them.

<b>HP NonStop</b>	Object permissions: Only trusted user accounts should have management, existence, alter, add, update or delete authority to the Universal Control Server installation libraries and objects.
<b>Windows</b>	<p>Eligible users of UCTL require read access to the message catalogs (*.umc files) in the <b>nls</b> subdirectory of the Workload Automation installation directory.</p> <p>If security is activated, all eligible users of UCTL require permission to create directories in the UCTL Server working directory. A directory named after the user ID requesting the command is created for each user. The directory is created while impersonating the user; hence, it is created using the user's security account.</p> <p>Home directories are created with permissions giving the user full control of both the directory and the files within them.</p>

### Configuration Files

Only trusted user accounts should have write access to the Universal Control Server configuration files.



#### Windows

Although you may edit configuration files with any text editor (for example, Notepad), we recommend that you manage configuration options using the Universal Configuration Manager Control Panel application. Only user accounts in the Administrator group may execute the [Universal Configuration Manager](#).

### Universal Control Server User ID

Universal Control Server requires read access to its installation directory and its working directory (defined in the component definition). The Universal Control Server security identity is inherited from the Universal Broker.

<b>IBM i</b>	The associated user profile ( <b>UNVUBR510</b> ) provides *ALLOBJ authority.
<b>z/OS</b>	UCTL Server requires read access to its installation data sets and its HFS working directory (defined in the component definition).

### User Authentication

User authentication is the process of verifying that a user is a known and valid user. The process used by Universal Control Server requires the user to provide a user name / ID and a password. The Universal Control Server passes the name / ID and password to the operating system for verification; this is referred to as logging on the user.



#### Windows

Windows provides two primary types of log on processes: batch and interactive.

A user must be given the right to log on as a batch job in order for the user to do a batch log on. All users can do an interactive log on. (See the Universal Control Server [LOGON\\_METHOD](#) option for more details.)

## Security - Universal Event Log Dump

- [Security Access](#)
- [Event Log Access](#)
- [Universal Configuration Manager](#)

### Security Access

No special security access is required to run Universal Event Log Dump (UELD). However, accessing the event logs and setting configuration options may require some special security considerations.

### Event Log Access

The system and application event logs may be read by all user accounts. The security log can only be accessed by accounts with Administrator privileges. Administrator privileges are also required to clear any of the event logs.

### Universal Configuration Manager

Only trusted user accounts should have write access to the Universal Event Log Dump configuration files.



#### Windows

Although you may edit configuration files with any text editor (for example, Notepad), we recommend that you manage configuration options using the Universal Configuration Manager Control Panel application. Only user accounts in the Administrator group may execute the [Universal Configuration Manager](#).



## Security - Universal Spool List

### Account Access

The account used to execute the Universal Spool List utility must have read access to the database files listed in [Workload Automation 5 - Databases](#).

## Security - Universal Spool Remove

### Account Access

The user account used to run the Universal Spool Remove utility must have read/write access to the database files listed in [Workload Automation 5 - Databases](#).

## Security - Universal Access Control List (UACL)

Error formatting macro: redirect: java.lang.NullPointerException

## Universal Access Control List - Overview

- [Overview](#)
- [UACL Configuration](#)

### Overview

Many Workload Automation 5 components utilize the Universal Access Control List (UACL) feature as an extra layer of security to the services they offer. UACLs are used for a variety of reasons but generally are used to determine if a client request is allowed or denied permission to the service and to set security attributes for the client request.

Each Universal Broker has an associated UACL configuration file that contains all the [UACL entries](#) for that system. The UACL entries can be used to enforce a security policy specific to the system on which its deployed.

The following Workload Automation 5 components use the UACL feature:

Component	Description
Universal Automation Center Agent	UACLs are used to control whether user credentials are required for task execution and to control whether or not user authentication is required. See <a href="#">Universal Automation Center Agent UACL</a> for complete details.
Universal Broker	UACLs are used to permit or deny TCP/IP client connections. See <a href="#">Universal Broker UACL</a> for complete details.
Universal Command Server	UACLs are used to permit or deny Universal Command Manager access and to control whether or not the Manager request requires user authentication. See <a href="#">Universal Command UACL</a> for complete details.
Universal Control Server	UACLs are used to permit or deny Universal Control Manager access and to control whether or not the Manager request requires user authentication. See <a href="#">Universal Control UACL</a> for complete details.
Universal Data Mover Server	UACLs are used to permit or deny Universal Data Manager access and to control whether or not the Manager request requires user authentication. See <a href="#">Universal Data Mover UACL</a> for complete details.
Universal Event Monitor Server	UACLs are used to permit or deny Universal Event Monitor Manager access and to control user authentication for event handlers. See <a href="#">Universal Event Monitor UACL</a> for complete details.

### UACL Configuration

UACL entries are maintained in a configuration file. The UACL configuration file is required for the Universal Broker to start even if there are no UACL entries defined in it.

The UACL configuration file syntax is the same as all other Workload Automation configuration files except for one difference: multiple UACL entries of the same name may be defined. The order in which the UACL entries are listed in the configuration file determines the order in which they are searched. See [Configuration File Syntax](#) for details on configuration file syntax.

The following table describes the location of the UACL configuration file and how it is accessed for each platform.

Platform	Description
z/OS	All UACL entries are defined in member <b>ACLCFG00</b> in library <b>UNVCONF</b> . The Universal Broker started task allocates the UACL configuration file to ddname <b>UNVACL</b> .
UNIX	All UACL entries are defined in the <b>uacl.conf</b> configuration file. This file is installed in <b>/etc/universal</b> by default. The UACL file is searched for in the same manner as all other product configuration files.
Windows	All UACL entries are defined in the <b>uacl.conf</b> configuration file. The location of this file depends on the version of Windows. It is recommended to use the Windows <a href="#">Universal Configuration Manager</a> to view and update UACL entries.
IBM i	All UACL entries are defined in member <b>UACL</b> of file <b>UNVCONF</b> .
HP NonStop	All UACL entries are defined in file <b>uacfcfg</b> . The UACL file is located within the same subvolume as all other product configuration files.

## Universal Access Control List - UACL Entries

Error formatting macro: redirect: java.lang.NullPointerException

## UACL Entries - Overview

- [UACL Entries](#)
  - [UACL Entries Example](#)
  - [Rule Syntax](#)
- [Searching UACL Entries](#)
  - [Generics](#)

### UACL Entries

UACL entries are specified in the UACL configuration file. All Workload Automation 5 UACL configuration files are simple text files. UACL entries are defined one per line in the following format:

```
TYPE RULE
```

- **TYPE** identifies the UACL entry. Each Workload Automation component using UACL entries have a defined set of entry types it supports. For example, the Universal Broker component uses UACL entries of type **ubroker\_access**.
- **RULE** defines the UACL entry matching criteria, the client access, and potentially some additional security attributes. The client's identity and the client's request are used to match UACL rules

UACL entries of the same type are listed in the order in which they should be searched. Since all UACL entries are in the same UACL configuration file, it is highly recommended to keep entries grouped together by their type for easy maintenance.

There is no limit to the number of UACL entries that can be specified in the UACL configuration file.

### UACL Entries Example

An example of UACL entries in a UACL configuration file is listed below.

```
ucmd_access      10.20.30.,TS1004,tsup1004,allow,noauth
ucmd_access      10.20.30.,TS1004,*,allow,auth
ucmd_access      10.20.30.,*,*,deny,auth
ucmd_access      ALL,*,root,deny,auth

ucmd_cert_access joe,tsup1004,allow,noauth
ucmd_cert_access joe,*,allow,auth
ucmd_cert_access operations,*,deny,auth
ucmd_cert_access *,root,deny,auth
```

### Rule Syntax

The UACL entry rule consists of a comma-separated list of values. If there is a space or tab character in the list of values, the entire list must be enclosed in quotation (") characters. What values are required and the meaning of each value is specific to the UACL entry type and are defined in the Workload Automation component documentation for its UACL entries.

As an example, the following **ucmd\_request** UACL entry contains a space in the "DSPLIB QGPL" rule value, so the entire rule is enclosed in quotation characters.

```
ucmd_request "prod.host.name,remoteuser,localuser,cmd,DSPLIB QGPL,allow,auth"
```

### Searching UACL Entries

UACL entries are searched in the order they are listed in the UACL configuration file. The search criteria is based on the client identity and the client request. Once a matching UACL entry has been found, the search stops and the matching entry is used.

The client identity is defined as a combination of the client TCP/IP IP address, client TCP/IP host name, client user ID, and client digital certificate. See [UACL Entries - Client Identification](#) for details on client identification.

The client request is defined based on the UACL entry type. There is typically an UACL entry type representing the different types of client requests. Each rule value has fields that correspond to the client request values. See [UACL Entries - Request Identification](#) for details on request identification.

## Generics

Generics can be used in the UACL rule to match client requests. Generics allow you to specify a string pattern in the rule to match a client request value. The string pattern is a convenient way of specifying one or more values.

The following pattern control characters can be used:

Control Character	Description
*	Match 0 or more characters.
?	Match one character.
/	Escape character to escape matching control characters so they are used as literal characters and to specify control codes.

In addition to the pattern control characters, pattern control codes can be specified to control how the pattern matching is performed. Pattern control codes are specified in the pattern string by prefixing them with the escape character, which is the slash character ( / ).

The following pattern control characters can be used:

Control Codes	Description
c	Perform a case insensitive compare.
C	Perform a case sensitive compare (the default).
s	Normalize spaces by reducing multiple spaces to one.
S	Don't normalize spaces (the default).

Some example string patterns using generics are listed below.

Pattern	Description
*le	matches "apple", "le", and "red apple".
/*le	matches "*le" only.
ap?le	matches "apple", "ap le", but not "apple".
/c*le	matches "apple", "APPLE", and "appLe".
a/c*le	matches "apple", "aPpLe", but not "APPLE".
/s*le	matches "apple", "red apple", and "red apple".

## UACL Entries - Client Identification

- Client Identification Methods
  - X.509 Certificate Authentication
    - Certificate Map Matching Criteria
    - Certificate Identifier Field
  - Client IP Address Identification
    - Client IP Address - Matching Criteria

### Client Identification Methods

Rule matching is based on the client identity and the client request.

There are two client identification methods:

1. X.509 Certificate Authentication
2. Client IP Address Identification

### X.509 Certificate Authentication

X.509 certificates identify an entity. An entity can be a program, person, or host computer. When an X.509 certificate is authenticated, it authenticates that the entity is who it claims to be.

X.509 certificates are utilized in UACL entries by first mapping a client certificate to a UACL certificate identifier. The certificate identifier then is used in the UACL entries. A certificate identifier provides for:

1. Concise representation of certificates in UACL entries. There are a large number of certificate fields that may be used and many of the fields have lengthy, tedious naming formats. A certificate map only needs to be defined once and then the concise certificate identifier can be used in the UACL entries.
2. Mapping of one or more certificates to a single certificate identity. A group of entities that share a common security access level may be represented by one certificate identity reducing the number of UACL entries to maintain.

UACL certificate map entries are searched sequentially (that is, top to bottom) matching the client certificate to each entry until a match is found. The certificate map defines a set of X.509 certificate fields that may be used as matching criteria.

### Certificate Map Matching Criteria

The following table defines the certificate map matching criteria.

Criteria	Description
SUBJECT	<p>Matches the X.509 subject field. The subject field is formatted as an X.501 Distinguished Name (DN). A DN is a hierarchical list of attributes referred to as Relative Distinguished Names (RDNs).</p> <p>RDNs are separated with a comma ( , ) by default. If a different separator is required (perhaps one of the RDN values uses a comma), start the DN with the different separator character.</p> <p>Valid separators are slash ( / ), comma ( , ), semicolon ( ; ), and period ( . ).</p> <p>Many RDN values can be used in a DN. Some of the most common values are:</p> <ul style="list-style-type: none"> <li>• C (Country name)</li> <li>• CN (Common name)</li> <li>• L (Locality)</li> <li>• O (Organization)</li> <li>• OU (Organizational Unit)</li> <li>• ST (State)</li> </ul> <p>The RDN attributes must be listed in the same order as they are defined in the certificate to be considered matched.</p> <p>A partial DN can be specified. All certificates that have a subject name that matches up to the last RDN are considered a match. This permits a group of certificates to be matched.</p> <p>The RDN attribute values can include pattern matching characters. An asterisk ( * ) matches 0 or more characters and a question mark ( ? ) matches one character.</p> <p>Some example of SUBJECT values are:</p>



	<ul style="list-style-type: none"> <li>• <b>subject="C=US,ST=Georgia,O=Acme,CN=Road Runner"</b></li> <li>• <b>subject="C=US,ST=Georgia,O=Acme,CN=Road *"</b></li> <li>• <b>subject="C=US,ST=Georgia,O=Acme,CN=Road ?unner"</b></li> </ul> <p>Whether an RDN value is case sensitive or not depends on the format in which the value is stored. The certificate creator has some control over which format is used. All formats except for printableString are case sensitive.</p>
EMAIL	<p>Matches the X.509 <b>emailAddress</b> attribute of the <b>subject</b> field and <b>rfc822Name</b> of the <b>subjectAltName</b> extension value. Both fields format the email address as an RFC 822 <b>addr-spec</b> in the form of <b>identifier@domain</b>.</p> <p>The attribute values may include pattern matching characters. An asterisk ( * ) matches 0 or more characters and a question mark ( ? ) matches one character.</p> <p>Some example EMAIL values are:</p> <ul style="list-style-type: none"> <li>• <b>email=user1@acme.com</b></li> <li>• <b>email=*@acme.com</b></li> <li>• <b>email=user?@acme.com</b></li> </ul> <p>RFC 822 names are not case sensitive.</p>
HOSTNAME	<p>Matches the following X.509 fields in the order listed:</p> <ol style="list-style-type: none"> <li>1. <b>dNSName</b> of the <b>subjectAltName</b> extension value.</li> <li>2. <b>commonName (CN) RDN attribute of the *subject</b> field's DN value.</li> </ol> <p>Some example HOSTNAME values are:</p> <ul style="list-style-type: none"> <li>• <b>hostname=bigfish.acme.com</b></li> <li>• <b>hostname=*.acme.com</b></li> </ul> <p>The values are not case sensitive.</p>
IP ADDRESS	<p>Matches the X.509 <b>IPAddress</b> field of the <b>subjectAltName</b> extension value.</p> <p>An example IPADDRESS value is:</p> <ul style="list-style-type: none"> <li>• <b>ipaddress=10.20.30.40</b></li> </ul>
SERIAL NUMBER	<p>Matches the X.509 <b>serialNumber</b> value.</p> <p>The value can be specified in a hexadecimal format by prefixing the value with <b>0x</b> or <b>0X</b>, otherwise, the value is considered a decimal format. For example, the value <b>0x016A392E7F</b> would be considered a hexadecimal format.</p> <p>An example SERIALNUMBER value is:</p> <ul style="list-style-type: none"> <li>• <b>serialnumber=0x7a2d52cbae</b></li> </ul>

#### Certificate Identifier Field

If a certificate map rule is found that matches the client certificate, the rule's identifier is assigned to the client's request. The certificate identifier is then used in matching certificate-based UACL entries.

The following table defines the certificate identifier field as used in UACL entries.

Field	Description
CERTID	<p>Matches the certificate identifier defined by the certificate map entry. The CERTID value has the following syntax:</p> <ul style="list-style-type: none"> <li>• An asterisk ( * ) matches 0 or more characters and a question mark ( ? ) matches one character. For example, <b>AB*M</b> matches <b>ABCDM</b> and <b>ABM</b>. <b>AB?M</b> matches <b>ABCM</b>, but not <b>ABCDM</b>.</li> <li>• The comparison is case insensitive.</li> </ul>

- Pattern matching characters, such as the asterisk and question mark, are included in the text to be matched by prefixing them with a forward slash ( / ) character. For example, **A/\*B** matches **A\*B**. **A//B** matches **A/B**.


### Client IP Address Identification

TCP/IP provides a method to obtain a client's IP address. The IP address typically identifies the host computer on which the client is executing. However, there are exceptions to this. Networks can be configured with Network Address Translation (NAT) systems between the client and the Broker that hides the client's IP address. In addition to the client IP address, Workload Automation clients provide a user account name with which they are executing that is used to further refine the client's identity.

UACL entries are searched matching the client's IP address and user account to each entry until a match is found.

### Client IP Address - Matching Criteria

The following table defines possible matching criteria for IP address and user account client identification.

Field	Description
HOST	<p>Matches the TCP/IP address of the remote user.</p> <p>The HOST value has the following syntax:</p> <ul style="list-style-type: none"> <li>• Dotted numeric form of an IP address. For example, <b>10.20.30.40</b>.</li> <li>• Dotted numeric prefix of the IP addresses. For example, <b>10.20.30</b>. matches all IP addresses starting with <b>10.20.30</b>. The last dot (.) is required.</li> <li>• A net/mask expression. For example, <b>131.155.72.0/255.255.254.0</b> matches IP address range <b>131.155.72.0</b> through <b>131.155.73.255</b>. The mask and the host value are AND'ed together. The result must match net.</li> </ul> <div style="background-color: #ffffcc; padding: 5px; margin: 10px 0;"> <p> <b>Note</b> Contact your network administrator for calculation of the correct net / mask expression.</p> </div> <ul style="list-style-type: none"> <li>• Host name for an IP address. For example, <b>sysa.abc.com</b>.</li> <li>• Host name suffix for a range of IP addresses. For example, <b>.abc.com</b> matches all host names ending with <b>abc.com</b>, such as, <b>sysa.abc.com</b>. The first dot (.) is required.</li> <li>• A value of <b>ALL</b> matches all IP addresses. The value must be uppercase.</li> </ul>
REMOTE_USER	<p>Matches the user name with which the remote user is executing as on the remote system.</p> <p>The REMOTE_USER value has the following syntax:</p> <ul style="list-style-type: none"> <li>• An asterisk (*) matches 0 or more characters and a question mark ( ? ) matches one character. For example, <b>AB*M</b> matches <b>ABCDM</b> and <b>ABM</b>. <b>AB?M</b> matches <b>ABCM</b> but not <b>ABCDM</b>.</li> <li>• Control code /c switches off case-sensitivity and /C switches on case?ensitivity matching. The default is on. For example, <b>/cABC</b> matches <b>abc</b>. <b>/ca/Cbc</b> matches <b>Abc</b> but not <b>ABC</b>.</li> <li>• Pattern matching characters, such as the asterisk and question mark, are included in the text to be matched by prefixing them with a forward slash ( / ) character. For example, <b>A/*B</b> matches <b>A*B</b>. <b>A//B</b> matches <b>A/B</b>.</li> </ul>


## UACL Entries - Request Identification

### Request Identification

In addition to the client identity being used to search for UACL entries, the client's request may be part of the matching criteria. The exact request fields used are dependent on the component's UACL entry type.

### Request Fields

The following table lists a complete set of possible request fields. See each component's UACL entry definitions for further details.

Field	Description
LOCAL_USER	<p>Matches the local user name with which the remote user is requesting to execute as on the local host. LOCAL_USER value has the following syntax:</p> <ul style="list-style-type: none"> <li>An asterisk ( * ) matches 0 or more characters and a question mark ( ? ) matches one character. For example, <b>AB*M</b> matches <b>ABCDM</b> and <b>ABM</b>. <b>AB?M</b> matches <b>ABCM</b> but not <b>ABCDM</b>.</li> <li>Control code <b>/c</b> switches off case-sensitivity and <b>/C</b> switches on case?ensitivity matching. The default is on. For example, <b>/cABC</b> matches <b>abc</b>. <b>/ca/Cbc</b> matches <b>Abc</b> but not <b>ABC</b>.</li> <li>Pattern matching characters, such as the asterisk and question mark, are included in the text to be matched by prefixing them with a forward slash ( / ) character. For example, <b>A/*B</b> matches <b>A*B</b>. <b>A//B</b> matches <b>A/B</b>.</li> <li>Variable name <b>\$RMTUSER</b> can be included in the value. The variable name itself is not case sensitive. <b>\$RMTUSER</b> and <b>\$rmtuser</b> are the same. The <b>\$RMTUSER</b> variable value is the user name with which the remote user is executing. It is the same value used in matching the REMOTE_USER field.</li> </ul> <p>A space character delimits the variable name, or it can be enclosed in parentheses (for example, <b>\$(RMTUSER)</b>), in which case it is delimited by the right parenthesis. This is useful if it is immediately followed by text.</p> <p>For example, if the remote user name is <b>TOM</b>, a LOCAL_USER value of <b>\$RMTUSER</b> will match if the local user name requested is also <b>TOM</b>. A LOCAL_USER value of <b>\$(RMTUSER)01</b> will match if the local user name requested is <b>TOM01</b>.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #cfe2f3;"> <p> <b>Windows</b> The LOCAL_USER value is not case sensitive, since Windows user account names are not case sensitive.</p> </div>
REQUEST_TYPE	<p>Matches the type of request a Universal Command Manager is requesting. The REQUEST_TYPE value has the following syntax:</p> <ul style="list-style-type: none"> <li>An asterisk ( * ) matches 0 or more characters and a question mark ( ? ) matches one character. For example, <b>AB*M</b> matches <b>ABCDM</b> and <b>ABM</b>. <b>AB?M</b> matches <b>ABCM</b> but not <b>ABCDM</b>.</li> <li>The comparison is case insensitive.</li> <li>Pattern matching characters, such as the asterisk and question mark, are included in the text to be matched by prefixing them with a forward slash ( / ) character. For example, <b>A/*B</b> matches <b>A*B</b>. <b>A//B</b> matches <b>A/B</b>.</li> </ul>
REQUEST_NAME	<p>The REQUEST_NAME field matches the name of a Universal Command Manager is request. The REQUEST_NAME value has the following syntax:</p> <ul style="list-style-type: none"> <li>An asterisk ( * ) matches 0 or more characters and a question mark ( ? ) matches one character. For example, <b>AB*M</b> matches <b>ABCDM</b> and <b>ABM</b>. <b>AB?M</b> matches <b>ABCM</b> but not <b>ABCDM</b>.</li> <li>Case sensitivity depends on the REQUEST_TYPE and the operating system on which the Universal Command Server is executing. See the Server's Security section for the operating system in question.</li> <li>Control code <b>/c</b> switches off case-sensitivity and <b>/C</b> switches on case?ensitivity matching. The default is on. For example, <b>/cABC</b> matches <b>abc</b>. <b>/ca/Cbc</b> matches <b>Abc</b> but not <b>ABC</b>.</li> <li>Control code <b>/s</b> normalizes spaces and <b>/S</b> does not normalize spaces. Space normalization removes preceding and trailing spaces as well as reduce consecutive multiple spaces to a single space. The default is no space normalization. For example, <b>/sa b c</b> matches <b>a b c</b>. <b>/Sa b c</b> matches <b>a b c</b> but not <b>a bc</b>.</li> <li>Pattern matching characters, such as the asterisk and question mark, are included in the text to be matched by prefixing them with a forward slash ( / ) character. For example, <b>A/*B</b> matches <b>A*B</b>. <b>A//B</b> matches <b>A/B</b>.</li> </ul>



## UACL Entries - Certificate- and Non Certificate-Based Entries

- [Certificate-Based and Non Certificate-Based UACL Entries](#)
  - [Certificate-Based Entries Search](#)
  - [Non Certificate-Based Entries Search](#)

### ***Certificate-Based and Non Certificate-Based UACL Entries***

Workload Automation components that support X.509 certificates define their UACL entries in two varieties:

1. Certificate-based entries
2. Non certificate-based entries

The two entry types are distinguished by their name. For example, **ucmd\_cert\_access** is the certificate-based form of the entry and **ucmd\_access** is a non certificate-based entry. All entries follow the same format.

Either the certificate-based UACL entries or the non certificate-based UACL entries are searched, but not both.

#### **Certificate-Based Entries Search**

Certificate-based UACL entries are searched under the following conditions:

- Client provides an X.509 certificate that matches a certificate map entry.

#### **Non Certificate-Based Entries Search**

Non certificate-based UACL entries are searched under the following conditions:

- Client provides an X.509 certificate and no certificate map entry matches.
- Client does not provide an X.509 certificate.

## **UACL Entries for Workload Automation 5 Components**

### ***UACL Entries for Workload Automation 5 Components***

The following UACL entries are available for Workload Automation 5 components:

- Universal Broker UACL Entries
- Universal Automation Center Agent UACL Entries
- Universal Command UACL Entries
- Universal Data Mover UACL Entries
- Universal Event Monitor UACL Entries
- Universal Control UACL Entries

## Security - Universal Access Control List (UACL) Examples

### UACL Examples

- UACL Examples - Universal Broker for z/OS
- UACL Examples - Universal Command Server for z/OS
- UACL Examples - Universal Data Mover Server for z/OS
- UACL Examples - Universal Control Server for z/OS
- UACL Examples - Universal Broker for Windows
- UACL Examples - Universal Command Server for Windows
- UACL Examples - Universal Data Mover Server for Windows
- UACL Examples - Universal Control Server for Windows
- UACL Examples - Universal Event Monitor Server for Windows
- UACL Examples - Universal Broker for UNIX
- UACL Examples - Universal Command Server for UNIX
- UACL Examples - Universal Data Mover Server for UNIX
- UACL Examples - Universal Control Server for UNIX
- UACL Examples - Universal Event Monitor Server for UNIX
- UACL Examples - Universal Broker for IBM i
- UACL Examples - Universal Command Server for IBM i
- UACL Examples - Universal Data Mover Server for IBM i
- UACL Examples - Universal Control Server for IBM i
- UACL Examples - Universal Broker for HP NonStop
- UACL Examples - Universal Command Server for HP NonStop
- UACL Examples - Universal Control Server for HP NonStop]

## UACL Examples - Universal Broker for zOS

### Universal Broker for z/OS

The following set of rules authorize the Universal Enterprise Controller at address 10.20.30, with update access to the product configuration files and setting of the configuration managed mode of the Broker, and denies all other connections.

```
remote_config_access    10.20.30.,*,allow,allow
remote_config_access    ALL,*,deny,deny
```

The following set of rules permit connections for the subnet 10.20.30 and denies all other connections.

```
ubroker_access         10.20.30.,allow
ubroker_access         ALL,deny
```

The following set of rules permit connections from host 10.20.30.40 and 10.20.30.50 and denies all other connections.

```
ubroker_access         10.20.30.40,allow
ubroker_access         10.20.30.50,allow
ubroker_access         ALL,deny
```

The following set of rules map X.509 certificates to certificate identifiers.

```
cert_map               id=joe,subject="/C=US/ST=Georgia/O=Acme, Inc./
                       OU=Sales/CN=Joe Black"
```

## Components

Universal Broker for zOS



## UACL Examples - Universal Command Server for zOS

### Universal Command Server for z/OS

The following set of rules permit services for the subnet 10.20.30 and denies all other connections unless an X.509 certificate is presented that maps to certificate ID operations.

```
ucmd_access      10.20.30.*,*,*,allow,auth
ucmd_access      ALL,*,*,deny,auth

ucmd_cert_access operations,*,allow,auth
ucmd_cert_access *,*,deny,auth
```

When no certificate is presented that maps to a certificate ID, the following set of rules effectively permit connections from any host, but has limited access from host 10.20.30.40 to user **TS1004** on that host.

- No host can execute commands as local user **SUPERID**.
- User **TS1004** on host 10.20.30.40 can execute commands as local user **TSUP1004** without providing the password.
- Users **TS1004** from host 10.20.30.40 can execute commands as any local user by providing the local user password.

When a certificate is presented that maps to a certificate ID, certificate ID joe can request local user ID **TSUP1004** without a password.

- Certificate ID **joe** is allowed to execute commands with any other local user ID with a password.
- Certificate ID operations cannot run anything.
- All other certificate IDs can execute commands with any user ID except for **SUPERID** with a password.

```
ucmd_access      10.20.30.40,TS1004,tsup1004,allow,noauth
ucmd_access      10.20.30.40,TS1004,*,allow,auth
ucmd_access      10.20.30.40,*,*,deny,auth
ucmd_access      ALL,*,superid,deny,auth

ucmd_cert_access joe,tsup1004,allow,noauth
ucmd_cert_access joe,*,allow,auth
ucmd_cert_access operations,*,deny,auth
ucmd_cert_access *,superid,deny,auth
```

### Components

Universal Command Server for zOS

## UACL Examples - Universal Data Mover Server for zOS

### Universal Data Mover Server for z/OS

The following set of rules permit services for the subnet 10.20.30 and denies all other connections.

```
udm_access 10.20.30.,*,*,allow,auth
udm_access ALL,*,*,deny,auth
```

The following set of rules effectively permit connections from any host, but has limited access from host 10.20.30.40 to user TS1004 on that host.

- No host can execute commands as local user root.
- User TS1004 on host 10.20.30.40 can execute commands as local user tsup1004 without providing the password.
- Users TS1004 from host 10.20.30.40 can execute commands as any local user by providing the local user password.

```
udm_access 10.20.30.40,TS1004,tsup1004,allow,noauth
udm_access 10.20.30.40,TS1004,*,allow,auth
udm_access 10.20.30.40,*,*,deny,auth
udm_access ALL,*,root,deny,auth
```

### Components

Universal Data Mover Server for zOS

## UACL Examples - Universal Control Server for zOS

### Universal Control Server for z/OS

The following set of rules permit services for the subnet 10.20.30 and denies all other connections unless an X.509 certificate is presented that maps to certificate ID operations.

```
uctl_access      10.20.30.,*,*,allow,auth
uctl_access      ALL,*,*,deny,auth

uctl_cert_access operations,*,allow,auth
uctl_cert_access *,*,deny,auth
```

When no certificate is presented that maps to a certificate ID, the following set of rules effectively permit connections from any host, but has limited access from host 10.20.30.40 to user TS1004 on that host.

- No host can execute commands as local user root.
- User TS1004 on host 10.20.30.40 can execute commands as local user tsup1004 without providing the password.
- Users TS1004 from host 10.20.30.40 can execute commands as any local user by providing the local user password.

When a certificate is presented that maps to a certificate ID, certificate ID joe can request local user id TSUP1004 without a password.

- Certificate ID joe is allowed to execute commands with any other local user ID with a password.
- Certificate ID operations cannot run anything.
- All other certificate IDs can execute commands with any user ID except for SUPERID with a password.

```
uctl_access      10.20.30.40,TS1004,tsup1004,allow,noauth
uctl_access      10.20.30.40,TS1004,*,allow,auth
uctl_access      10.20.30.40,*,*,deny,auth
uctl_access      ALL,*,root,deny,auth

uctl_cert_access joe,tsup1004,allow,noauth
uctl_cert_access joe,*,allow,auth
uctl_cert_access operations,*,deny,auth
uctl_cert_access *,root,deny,auth
```

### Components

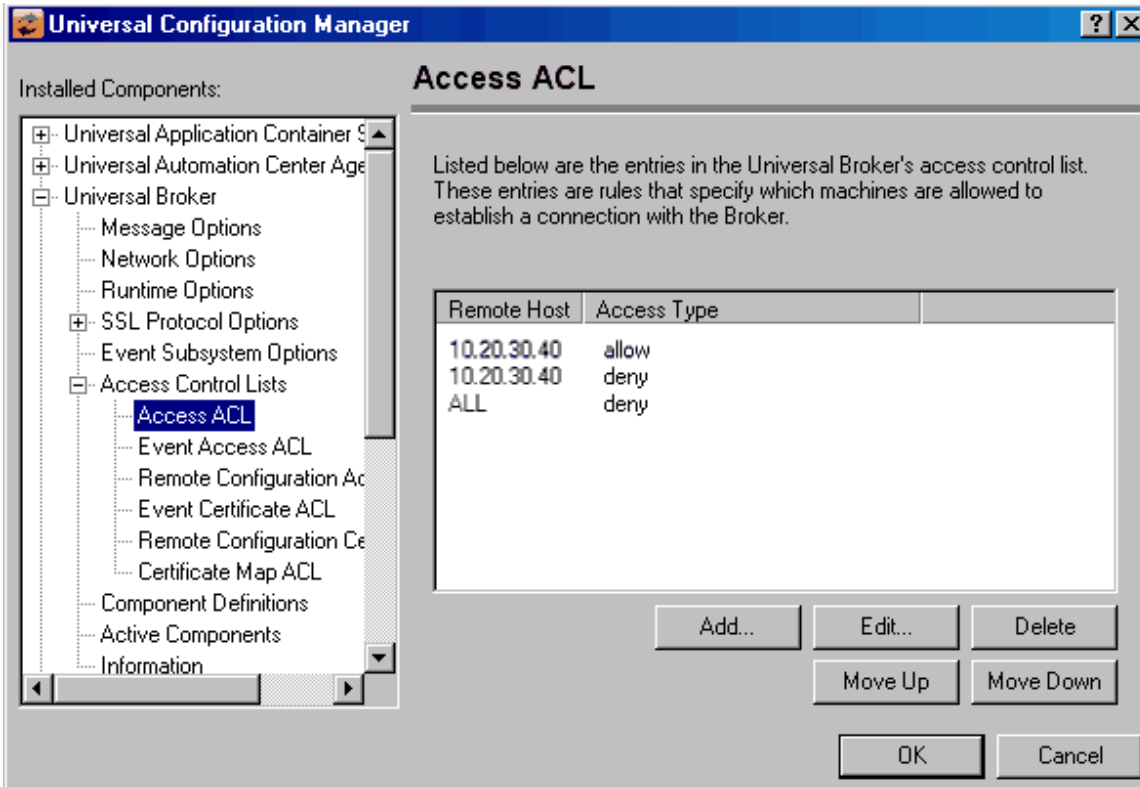
Universal Control

## UACL Examples - Universal Broker for Windows

### Universal Broker for Windows

Although UACL files can be edited with any text editor (for example, Notepad), the [Universal Configuration Manager](#) application, accessible via the Control Panel, is the recommended way to update UACL entries. From there, ACL entries can be added, changed, deleted or sorted (rules are applied in the order in which they are listed).

The following figure illustrates an example.



### Components

Universal Broker for Windows

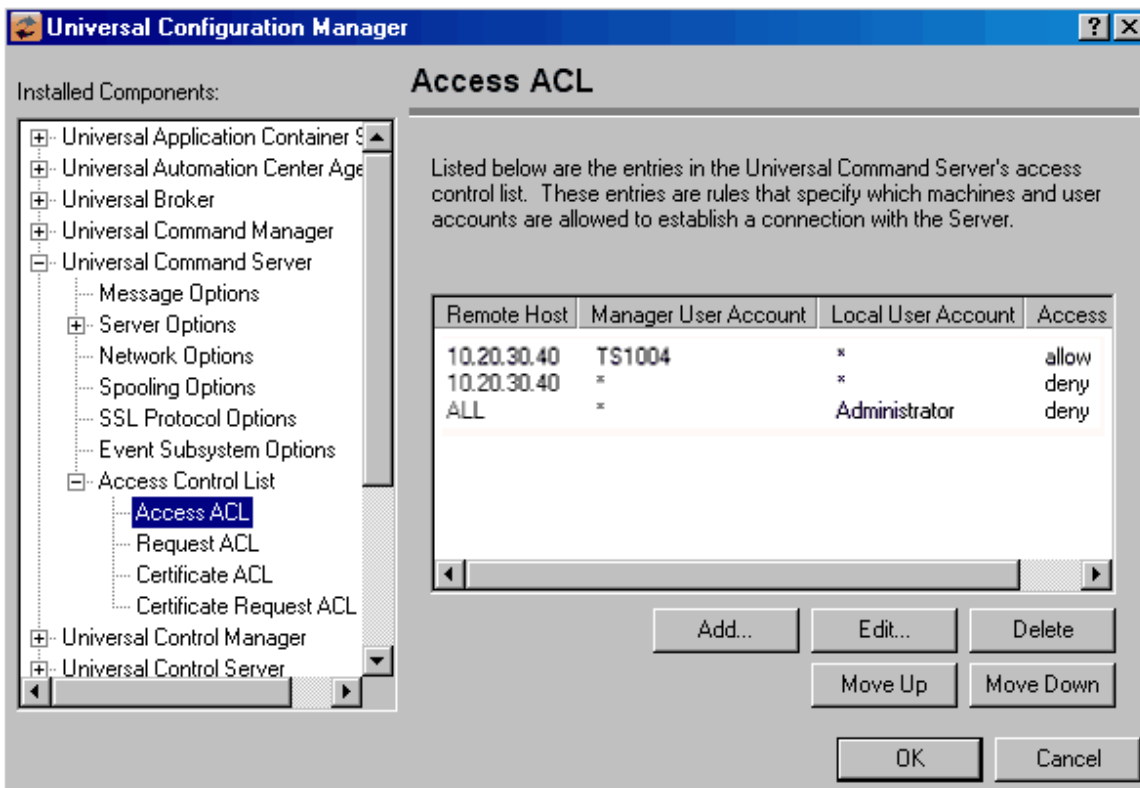
## UACL Examples - Universal Command Server for Windows

### Universal Command Server for Windows

Although UACL files can be edited with any text editor (for example, Notepad), the [Universal Configuration Manager](#) application, accessible via the Control Panel, is the recommended way to update UACL entries. From there, ACL entries can be added, changed, deleted or sorted (rules are applied in the order in which they are listed).

The following figure illustrates an example. The set of ACL entries only allows connections from host 10.20.30.40 if the user on that host is TS1004. All other remote users will be blocked.

- TS1004 may run processes on the local system using any user account, provided the correct password is supplied.
- No processes may be run with Universal Command using the Administrator account on the local system, regardless of where the request originated.



### Components

[Universal Command Server for Windows](#)

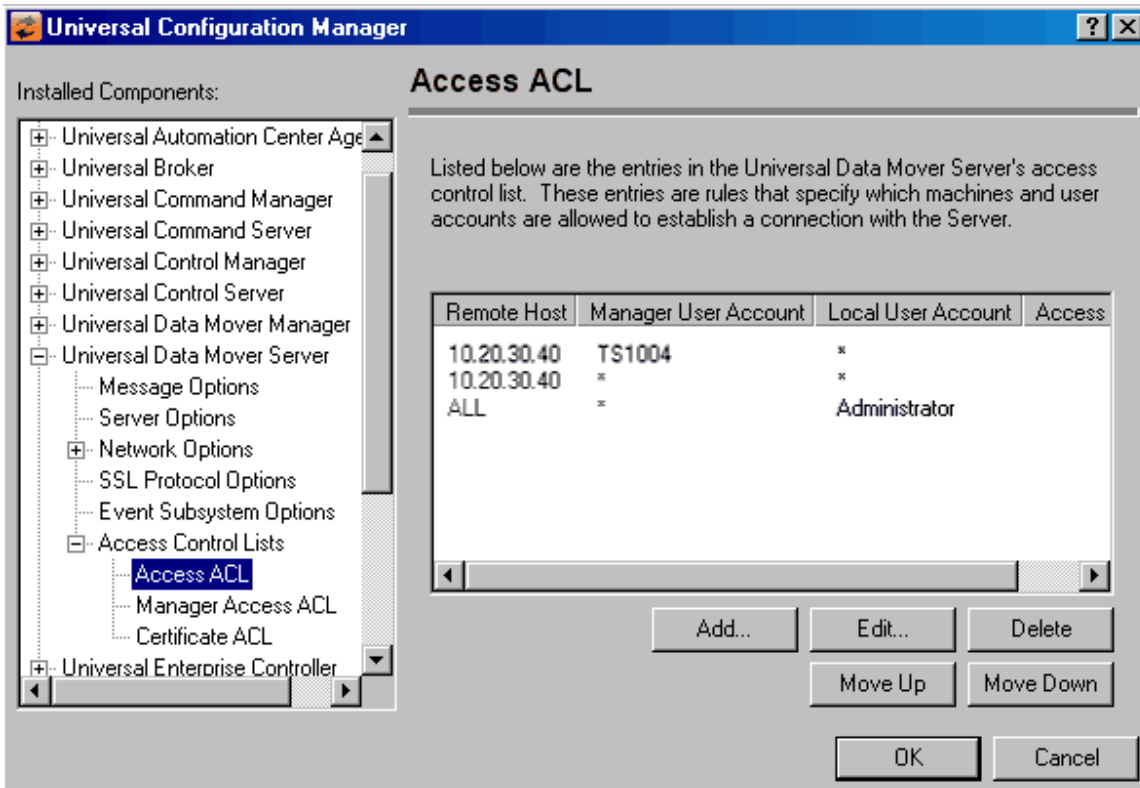
## UACL Examples - Universal Data Mover Server for Windows

### Universal Data Mover Server for Windows

Although UACL files can be edited with any text editor (for example, Notepad), the [Universal Configuration Manager](#) application, accessible via the Control Panel, is the recommended way to update UACL entries. From there, ACL entries can be added, changed, deleted or sorted (rules are applied in the order in which they are listed).

The following figure illustrates an example. The set of ACL entries only allows connections from host 10.20.30.40 if the user on that host is TS1004. All other remote users will be blocked.

- TS1004 may run processes on the local system using any user account, provided the correct password is supplied.
- No processes may be run with Universal Data Mover using the Administrator account on the local system, regardless of where the request originated.



### Components

Universal Data Mover Server for Windows

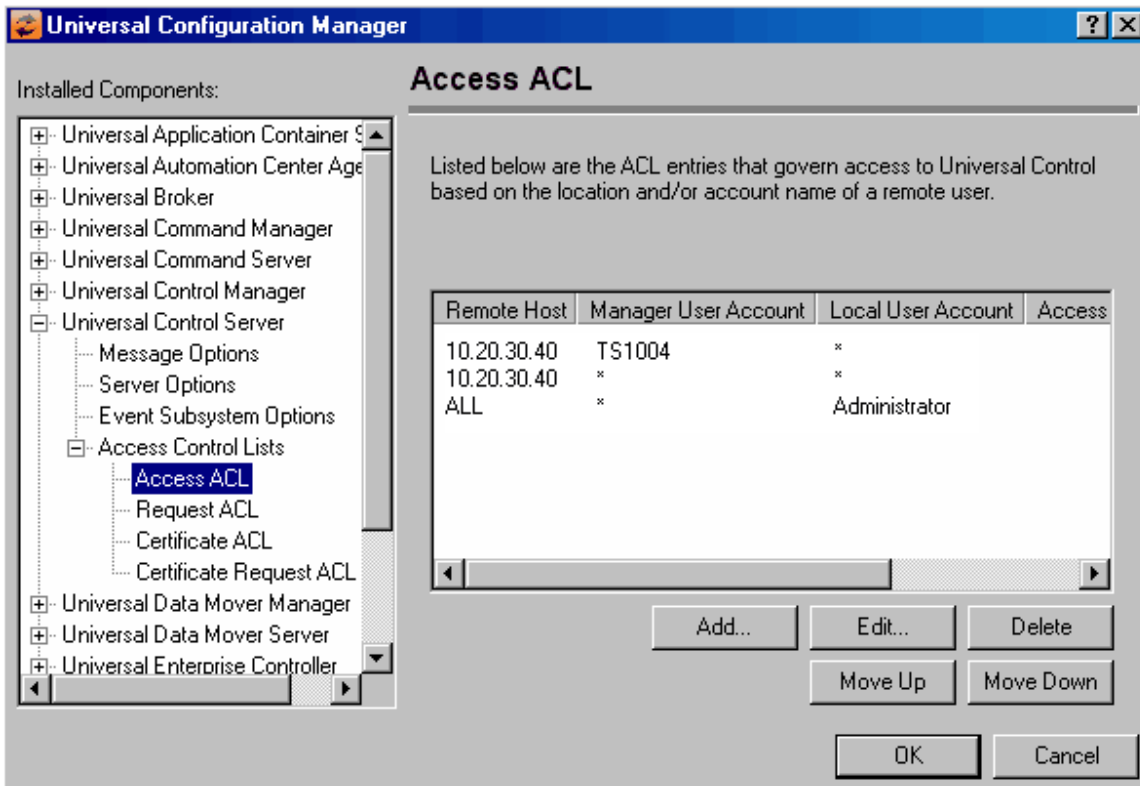
## UACL Examples - Universal Control Server for Windows

### Universal Control Server for Windows

Although UACL files can be edited with any text editor (for example, Notepad), the [Universal Configuration Manager](#) application, accessible via the Control Panel, is the recommended way to update UACL entries. From there, ACL entries can be added, changed, deleted or sorted (rules are applied in the order in which they are listed).

The following figure illustrates an example. The set of ACL entries only allows connections from host 10.20.30.40 if the user on that host is TS1004. All other remote users will be blocked.

- TS1004 may run processes on the local system using any user account, provided the correct password is supplied.
- No processes may be run with Universal Command using the Administrator account on the local system, regardless of where the request originated.



### Components

Universal Control

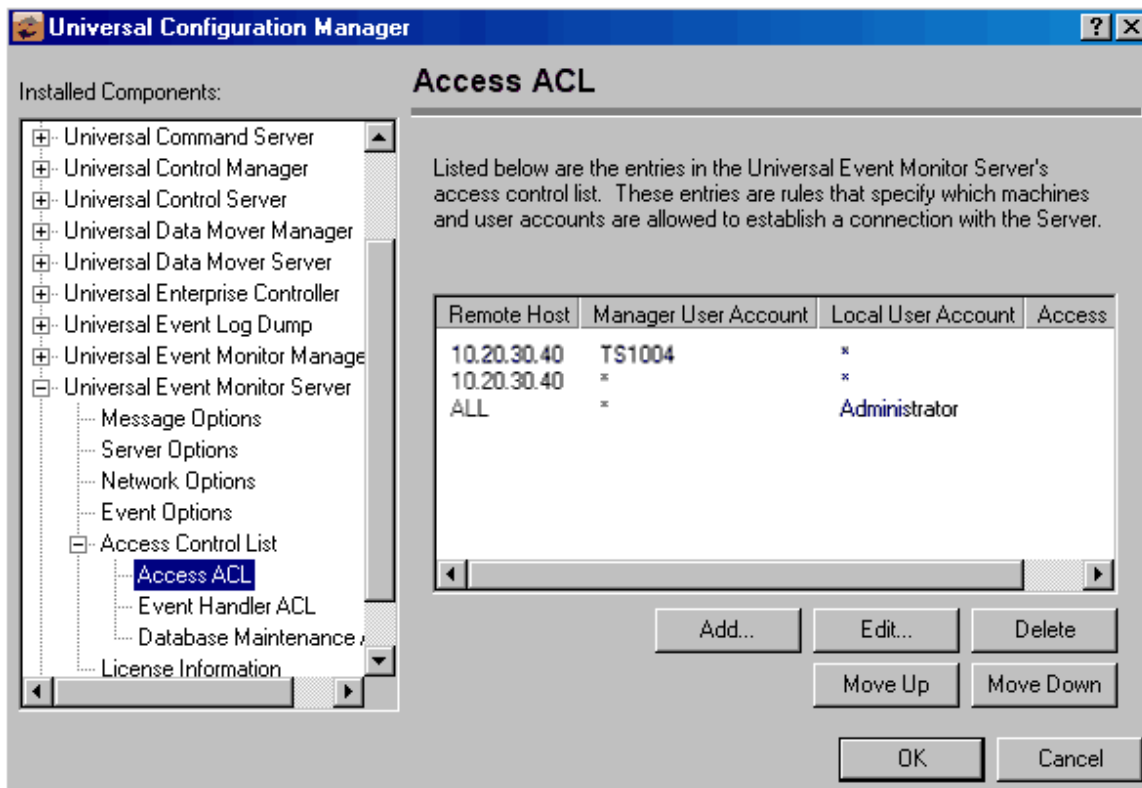
## UACL Examples - Universal Event Monitor Server for Windows

### Universal Event Monitor Server for Windows

Although UACL files can be edited with any text editor (for example, Notepad), the [Universal Configuration Manager](#) application, accessible via the Control Panel, is the recommended way to update UACL entries. From there, ACL entries can be added, changed, deleted, or sorted (rules are applied in the order in which they are listed).

The following figure illustrates an example. The set of ACL entries only allows connections from host 10.20.30.40 if the user on that host is TS1004. All other remote users will be blocked.

- TS1004 may run processes on the local system using any user account, provided the correct password is supplied.
- No processes may be run with Universal Event Monitor using the Administrator account on the local system, regardless of where the request originated.



### Components

Universal Event Monitor



## UACL Examples - Universal Broker for UNIX

### Universal Broker for UNIX

The following set of rules is required to allow I-Management Console to access Universal Broker.

```
remote_config_access    10.20.30.*,allow,allow
remote_config_access    ALL,*,deny,deny
```

The following set of rules permit connections for the subnet 10.20.30 and denies all other connections.

```
ubroker_access         10.20.30.,allow
ubroker_access         ALL,deny
```

The following set of rules permit connections from host 10.20.30.40 and 10.20.30.50 and denies all other connections.

```
ubroker_access         10.20.30.40,allow
ubroker_access         10.20.30.50,allow
ubroker_access         ALL,deny
```

The following set of rules map X.509 certificates to certificate identifiers.

```
cert_map               id=joe,subject="/C=US/ST=Georgia/O=Acme, Inc./"
```

### Components

Universal Broker for UNIX

## UACL Examples - Universal Command Server for UNIX

### Universal Command Server for UNIX

The following set of rules permit services for the subnet 10.20.30 and denies all other connections unless an X.509 certificate is presented that maps to certificate ID operations.

```
ucmd_access      10.20.30.*,*,*,allow,auth
ucmd_access      ALL,*,*,deny,auth

ucmd_cert_access operations,*,allow,auth
ucmd_cert_access *,*,deny,auth
```

When no certificate is presented that maps to a certificate ID, the following set of rules effectively permit connections from any host but has limited access from host 10.20.30.40 to user TS1004 on that host.

- No host can execute commands as local user root.
- User TS1004 on host 10.20.30.40 can execute commands as local user tsup1004 without providing the password.
- Users TS1004 from host 10.20.30.40 can execute commands as any local user by providing the local user password.

When a certificate is presented that maps to a certificate ID, certificate ID joe can request local user ID tsup1004 without a password.

- Certificate ID **joe** is allowed to execute commands with any other local user ID with a password.
- Certificate ID **operations** cannot run anything.
- All other certificate IDs can execute commands with any user ID except for root with a password.

```
ucmd_access      10.20.30.40,TS1004,tsup1004,allow,noauth
ucmd_access      10.20.30.40,TS1004,*,allow,auth
ucmd_access      10.20.30.40,*,*,deny,auth
ucmd_access      ALL,*,root,deny,auth

ucmd_cert_access joe,tsup1004,allow,noauth
ucmd_cert_access joe,*,allow,auth
ucmd_cert_access operations,*,deny,auth
ucmd_cert_access *,root,deny,auth
```

## Components

Universal Command Server for UNIX

## UACL Examples - Universal Data Mover Server for UNIX

### Universal Data Mover Server for UNIX

The following set of rules permit services for the subnet 10.20.30 and denies all other connections.

```
udm_access 10.20.30.,*,*,allow,auth
udm_access ALL,*,*,deny,auth
```

The following set of rules effectively permit connections from any host, but has limited access from host 10.20.30.40 to user TS1004 on that host.

- No host can execute commands as local user root.
- User TS1004 on host 10.20.30.40 can execute commands as local user tsup1004 without providing the password.
- Users TS1004 from host 10.20.30.40 can execute commands as any local user by providing the local user password.

```
udm_access 10.20.30.40,TS1004,tsup1004,allow,noauth
udm_access 10.20.30.40,TS1004,*,allow,auth
udm_access 10.20.30.40,*,*,deny,auth
udm_access ALL,*,root,deny,auth
```

### Components

Universal Data Mover Server for UNIX

## UACL Examples - Universal Control Server for UNIX

### Universal Control Server for UNIX

The following set of rules permit services for the subnet 10.20.30 and denies all other connections unless an X.509 certificate is presented that maps to certificate ID operations.

```
uctl_access      10.20.30.,*,*,allow,auth
uctl_access      ALL,*,*,deny,auth

uctl_cert_access operations,*,allow,auth
uctl_cert_access *,*,deny,auth
```

When no certificate is presented that maps to a certificate ID, the following set of rules effectively permits connections from any host, but has limited access from host 10.20.30.40 to user **TS1004** on that host.

- No host can execute commands as local user **root**.
- User **TS1004** on host 10.20.30.40 can execute commands as local user **tsup1004** without providing the password.
- User **TS1004** from host 10.20.30.40 can execute commands as any local user by providing the local user password.

When a certificate is presented that maps to a certificate ID, certificate ID **joe** can request local user id `t*sup1004*` without a password.

- Certificate ID **joe** is allowed to execute commands with any other local user ID with a password.
- Certificate ID operations cannot run anything.
- All other certificate IDs can execute commands with any user ID except for root with a password.

```
uctl_access      10.20.30.40,TS1004,tsup1004,allow,noauth
uctl_access      10.20.30.40,TS1004,*,allow,auth
uctl_access      10.20.30.40,*,*,deny,auth
uctl_access      ALL,*,root,deny,auth

uctl_cert_access joe,tsup1004,allow,noauth
uctl_cert_access joe,*,allow,auth
uctl_cert_access operations,*,deny,auth
uctl_cert_access *,root,deny,auth
```

## Components

### Universal Control

## UACL Examples - Universal Event Monitor Server for UNIX

### Universal Event Monitor Server for UNIX

The following set of rules permit services for the subnet 10.20.30 and denies all other connections unless an X.509 certificate is presented that maps to certificate ID operations.

```
uem_access 10.20.30.*,*,allow,auth
uem_access ALL,*,*,deny,auth
```

The following set of rules effectively permit connections from any host but has limited access from host 10.20.30.40 to user TS1004 on that host.

- No host can monitor events as local user root.
- User TS1004 on host 10.20.30.40 can monitor events as local user tsup1004 without providing the password.
- Users TS1004 from host 10.20.30.40 can execute commands as any local user by providing the local user password.

```
uem_access 10.20.30.40,TS1004,tsup1004,allow,noauth
uem_access 10.20.30.40,TS1004,*,allow,auth
uem_access 10.20.30.40,*,*,deny,auth
uem_access ALL,*,root,deny,auth
```

### Components

Universal Event Monitor Server for UNIX

## UACL Examples - Universal Broker for IBM i

### Universal Broker for IBM i

The following set of rules permit connections for the subnet 10.20.30 and denies all other connections.

```
ubroker_access    10.20.30.,*,allow,allow
ubroker_access    ALL,*,deny,deny
```

The following set of rules permit connections from host 10.20.30.40 and 10.20.30.50 and denies all other connections.

```
ubroker_access    10.20.30.40,allow
ubroker_access    10.20.30.50,allow
ubroker_access    ALL,deny
```

The following set of rules map X.509 certificates to certificate identifiers.

```
cert_map          id=joe,subject="/C=US/ST=Georgia/O=Acme, Inc./
                  OU=Sales/CN=Joe Black"
```

### Components

Universal Broker for IBM i

## UACL Examples - Universal Command Server for IBM i

### Universal Command Server for IBM i

The following set of rules permit services for the subnet 10.20.30 and denies all other connections unless an X.509 certificate is presented that maps to certificate ID operations.

```
ucmd_access      10.20.30.*,*,*,allow,auth
ucmd_access      ALL,*,*,deny,auth

ucmd_cert_access operations,*,allow,auth
ucmd_cert_access *,*,deny,auth
```

When no certificate is presented that maps to a certificate ID, the following set of rules effectively permit connections from any host, but has limited access from host 10.20.30.40 to user **TS1004** on that host.

- No host can execute commands as local user **root**.
- User **TS1004** on host 10.20.30.40 can execute commands as local user **tsup1004** without providing the password.
- Users **TS1004** from host 10.20.30.40 can execute commands as any local user by providing the local user password.

When a certificate is presented that maps to a certificate ID, certificate ID joe can request local user ID **tsup1004** without a password.

- Certificate ID **joe** is allowed to execute commands with any other local user ID with a password.
- Certificate ID operations cannot run anything.
- All other certificate IDs can execute commands with any user ID except for root with a password.

```
ucmd_access      10.20.30.40,TS1004,tsup1004,allow,noauth
ucmd_access      10.20.30.40,TS1004,*,allow,auth
ucmd_access      10.20.30.40,*,*,deny,auth
ucmd_access      ALL,*,root,deny,auth

ucmd_cert_access joe,tsup1004,allow,noauth
ucmd_cert_access joe,*,allow,auth
ucmd_cert_access operations,*,deny,auth
ucmd_cert_access *,root,deny,auth
```

## Components

Universal Command Server for IBM i

## UACL Examples - Universal Data Mover Server for IBM i

### Universal Data Mover Server for IBM i

The following set of rules permit services for the subnet 10.20.30 and denies all other connections.

```
udm_access    10.20.30.,*,*,allow,auth
udm_access    ALL,*,*,deny,auth
```

The following set of rules effectively permit connections from any host, but has limited access from host 10.20.30.40 to user TS1004 on that host.

- No host can execute commands as local user root.
- User TS1004 on host 10.20.30.40 can execute commands as local user tsup1004 without providing the password.
- Users TS1004 from host 10.20.30.40 can execute commands as any local user by providing the local user password.

```
udm_access    10.20.30.40,TS1004,tsup1004,allow,noauth
udm_access    10.20.30.40,TS1004,*,allow,auth
udm_access    10.20.30.40,*,*,deny,auth
udm_access    ALL,*,root,deny,auth
```

### Components

Universal Data Mover Server for IBM i



## UACL Examples - Universal Control Server for IBM i

### Universal Control Server for IBM i

The following set of rules permit services for the subnet 10.20.30 and denies all other connections unless an X.509 certificate is presented that maps to certificate ID operations.

```
uctl_access      10.20.30.*,*,allow,auth
uctl_access      ALL,*,*,deny,auth

uctl_cert_access operations,*,allow,auth
uctl_cert_access *,*,deny,auth
```

When no certificate is presented that maps to a certificate ID, the following set of rules effectively permit connections from any host but has limited access from host 10.20.30.40 to user **TS1004** on that host.

- No host can execute commands as local user root.
- User **TS1004** on host 10.20.30.40 can execute commands as local user **tsup1004** without providing the password.
- Users **TS1004** from host 10.20.30.40 can execute commands as any local user by providing the local user password.

When a certificate is presented that maps to a certificate ID, certificate ID **joe** can request local user ID **tsup1004** without a password.

- Certificate ID **joe** is allowed to execute commands with any other local user ID with a password.
- Certificate ID **operations** cannot run anything.
- All other certificate IDs can execute commands with any user ID except for root with a password.

```
uctl_access      10.20.30.40,TS1004,tsup1004,allow,noauth
uctl_access      10.20.30.40,TS1004,*,allow,auth
uctl_access      10.20.30.40,*,*,deny,auth
uctl_access      ALL,*,root,deny,auth

uctl_cert_access joe,tsup1004,allow,noauth
uctl_cert_access joe,*,allow,auth
uctl_cert_access operations,*,deny,auth
uctl_cert_access *,root,deny,auth
```

## Components

### Universal Control

## UACL Examples - Universal Broker for HP NonStop

### Universal Broker for HP NonStop

The following set of rules permit connections for the subnet 10.20.30 and denies all other connections.

```
ubroker_access    10.20.30.,allow
ubroker_access    ALL,deny
```

The following set of rules permit connections from host 10.20.30.40 and 10.20.30.50 and denies all other connections.

```
ubroker_access    10.20.30.40,allow
ubroker_access    10.20.30.50,allow
ubroker_access    ALL,deny
```

### **Components**

Universal Broker for HP NonStop

## UACL Examples - Universal Command Server for HP NonStop

### Universal Command Server for HP NonStop

The following set of rules permit services for the subnet 10.20.30 and denies all other connections.

```
ucmd_access    10.20.30.,*,*,allow,auth
ucmd_access    ALL,*,*,deny,auth
```

The following set of rules effectively permit connections from any host, but has limited access from host 10.20.30.40 to user TS1004 on that host.

- No host can execute commands as local user root.
- User TS1004 on host 10.20.30.40 can execute commands as local user tsup1004 without providing the password.
- Users TS1004 from host 10.20.30.40 can execute commands as any local user by providing the local user password.

```
ucmd_access    10.20.30.40,TS1004,tsup1004,allow,noauth
ucmd_access    10.20.30.40,TS1004,*,allow,auth
ucmd_access    10.20.30.40,*,*,deny,auth
ucmd_access    ALL,*,root,deny,auth
```

### Components

[Universal Command Server for HP NonStop](#)

## UACL Examples - Universal Control Server for HP NonStop

### Universal Control Server for HP NonStop

The following set of rules permit services for the subnet 10.20.30 and denies all other connections.

```
uctl_access    10.20.30.,*,*,allow,auth
uctl_access    ALL,*,*,deny,auth
```

The following set of rules effectively permit connections from any host, but has limited access from host 10.20.30.40 to user TS1004 on that host.

- No host can execute commands as local user root.
- User TS1004 on host 10.20.30.40 can execute commands as local user tsup1004 without providing the password.
- Users TS1004 from host 10.20.30.40 can execute commands as any local user by providing the local user password.

```
uctl_access    10.20.30.40,TS1004,tsup1004,allow,noauth
uctl_access    10.20.30.40,TS1004,*,allow,auth
uctl_access    10.20.30.40,*,*,deny,auth
uctl_access    ALL,*,root,deny,auth
```

### Components

Universal Control

## Security - X.509 Certificates

Error formatting macro: redirect: java.lang.NullPointerException



## X.509 Certificates - Sample X.509 Certificate


- [Sample X.509 Certificate](#)
- [Certificate Fields](#)

### Sample X.509 Certificate

The following figure illustrates a sample X.509 version 3 certificate for Joe Buck at the Acme corporation.

```

Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      01:02:03:04:05:06:07:08
    Signature Algorithm: md5WithRSAEncryption
    Issuer: C=US, ST=Florida, O=Acme, Inc., OU=Security, CN=CA Authority/emailAddress=ca@acme.com
    Validity
      Not Before: Aug 20 12:59:55 2011 GMT
      Not After : Aug 20 12:59:55 2011 GMT
    Subject: C=US, ST=Florida, O=Acme, Inc., OU=Sales, CN=Joe Buck
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (1024 bit)
      Modulus (1024 bit):
        00:be:5e:6e:f8:2c:c7:8c:07:7e:f0:ab:a5:12:db:
        fc:5a:1e:27:ba:49:b0:2c:e1:cb:4b:05:f2:23:09:
        77:13:75:57:08:29:45:29:d0:db:8c:06:4b:c3:10:
        88:e1:ba:5e:6f:1e:c0:2e:42:82:2b:e4:fa:ba:bc:
        45:e9:98:f8:e9:00:84:60:53:a6:11:2e:18:39:6e:
        ad:76:3e:75:8d:1e:b1:b2:1e:07:97:7f:49:31:35:
        25:55:0a:28:11:20:a6:7d:85:76:f7:9f:c4:66:90:
        e6:2d:ce:73:45:66:be:56:aa:ee:93:ae:10:f9:ba:
        24:fe:38:d0:f0:23:d7:a1:3b
      Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Basic Constraints:
        CA:FALSE
      X509v3 Subject Alternative Name:
        email:joe.buck@acme.com
    Signature Algorithm: md5WithRSAEncryption
    a0:94:ca:f4:d5:4f:2d:da:a8:6d:e3:41:6e:51:83:57:b3:b5:
    31:95:32:b6:ca:7e:d1:4f:fb:01:82:db:23:a0:39:d8:69:71:
    31:9c:0a:3b:ce:f6:c6:e2:5c:af:23:f0:d7:ee:87:3e:8a:7b:
    40:03:39:64:a1:8c:29:7d:5b:99:93:fa:23:19:e1:e4:ac:4d:
    13:0f:de:ad:51:27:e3:4e:4b:9f:40:4c:05:fd:f2:82:09:3e:
    46:05:f0:ad:cc:f7:78:25:3e:11:f8:ca:b6:df:f7:37:57:9b:
    63:00:d0:b5:b5:18:ec:38:73:d2:85:a3:c7:24:21:47:ee:f2:
    8c:0d
  
```

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

### Certificate Fields

A certificate is composed of many fields.

The following table describes the main certificate fields.

Field or Section	Description
Version	X.509 certificates come in two versions: 1 and 3.

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



## X.509 Certificates - SSL Peer Authentication

- [Overview](#)
- [Certificate Verification](#)
- [Certificate Revocation](#)
- [Certificate Identification](#)
- [Certificate Support](#)
- [Sample Set-up for Universal Command Peer Authentication of Universal Broker](#)
  - [Certificate](#)

### Overview

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

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

Certificate authentication is performed in the following steps:

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

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

### Certificate Verification

The Workload Automation 5 component must be configured with a list of trusted CA certificates. When a peer certificate is received, the trusted CA certificates are used to verify that the peer certificate is issued by one of the trusted CA's.

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

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

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

### Certificate Revocation

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

The `CERTIFICATE_REVOCATION_LIST` configuration option specifies the PEM-formatted file that contains the CRL. This option is available in all Workload Automation 5 components that utilize certificates.

### Certificate Identification

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

A Workload Automation 5 Manager validates a Broker certificate by the Broker host name, IP address, or the certificate serial number. The `VERIFY_HOST_NAME` configuration option is used to specify the host name or IP address that is identified in the Broker certificate. Each certificate signed by a CA must have a unique serial number for that CA. The `VERIFY_SERIAL_NUMBER` option is used to specify the serial number in the Broker certificate.

If certificate identification fails, the session is immediately terminated.

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

### Certificate Support

Many certificate authority applications, also known as Public Key Infrastructure (PKI) applications, are available. Workload Automation 5 should

be able to utilize any certificate in a PEM-formatted file. PEM (Privacy Enhanced Mail) is a common text file format used for certificates, private keys, and CA lists.

Workload Automation 5 support X.509 version 1 and version 3 certificates.

Although implementing a fully featured PKI infrastructure is beyond the scope of Workload Automation 5 and this documentation, some assistance is provided using the [OpenSSL toolkit](#).

Workload Automation 5 on most of the supported platforms utilize the OpenSSL toolkit for its SSL and certificate implementation. OpenSSL is delivered on most UNIX distributions and Windows distributions and also is available on the [OpenSSL website](#).

Workload Automation 5 supports z/OS System SSL on the IBM z/OS operating system as well as OpenSSL. System SSL interfaces directly with the RACF security product for certificate access. All certificates, CA and user certificates, and private keys must be stored in the RACF database to use System SSL.

The Workload Automation 5 suite includes an X.509 certificate utility, [Universal Certificate](#), to create certificates for use in the Workload Automation 5 suite.

### Sample Set-up for Universal Command Peer Authentication of Universal Broker

<b>Step 1</b>	Create a Self-Signed CA Request:  <pre>ucert -create request -request_file ca_req.pem -private_key_file ca_pkey.pem -country US -state GA -locality Alpharetta -organization Stonebranch -common_name Stonebranch</pre>
<b>Step 2</b>	Create a CA Certificate:  <pre>ucert -create cert -request_file ca_req.pem -private_key_file ca_pkey.pem -cert_file ca_cert.pem -ca yes -not_after_date +3650</pre>
<b>Step 3</b>	Create a Server Certificate Request:  <pre>ucert -create request -request_file ubrl_req.pem -private_key_file ubrl_pkey.pem -country US -state GA -locality Alpharetta -organization Stonebranch -common-name "l64agent"</pre>
<b>Step 4</b>	Create a Server Certificate:  <pre>ucert -create cert -ca_cert_file ca_cert.pem -request_file ubrl_req.pem -private_key_file ca_pkey.pem -cert_file ubrl_cert.pem -not_after_date +3650</pre>
<b>Step 5</b>	The following files are generated in Steps 1 - 4:  <ul style="list-style-type: none"> <li>• CA PKEY = ca_pkey.pem</li> <li>• CA CERT = ca_cert.pem</li> <li>• Server PKEY = ubrl_pkey.pem</li> <li>• Server CERT = ubrl_cert.pem</li> </ul>
<b>Step 6</b>	Add Server CERT and PKEY to the target ubroker.conf:  <ul style="list-style-type: none"> <li>• certificate /home/test/ubrl_cert.pem</li> <li>• private_key /home/test/ubrl_pkey.pem</li> </ul>
<b>Step 7</b>	Copy ca_cert.pem to the source server.
<b>Step 8</b>	Run the following command from the source server to test:  <pre>/opt/universal/bin/ucmd -host l64agent -userid test -pwd xxx -cmd "pwd" -level info -verify_host_name yes -ca_certs /home/test/ca_cert.pem</pre>
<b>Step 9</b>	Use <a href="#">Universal Certificate</a> to print the certificate and verify the certificate serial number:  <pre>ucert -print cert -cert_file ubrl_cert.pem</pre> See <a href="#">Certificate</a> , below.
<b>Step 10</b>	Run following command from the source server to test:  <pre>/opt/universal/bin/ucmd -host l64agent -userid test -pwd xxx -cmd "pwd" -level info -verify_host_name yes -ca_certs /home/test/ca_cert.pem -verify_serial_number x028c91a7fb2f26649</pre>

#### Certificate

```

Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      28:c9:1a:7f:b2:f2:66:49
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=US, ST=GA, L=Alpharetta, O=Stonebranch, CN=Stonebranch
    Validity
      Not Before: Feb  8 21:08:12 2016 GMT
      Not After : Feb  6 02:08:12 2026 GMT
    Subject: C=US, ST=GA, L=Alpharetta, O=Stonebranch, CN=l64agent
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (1024 bit)
        Modulus (1024 bit):
          00:d9:30:22:5b:b4:62:5c:d9:26:4b:16:02:cc:22:
          65:b8:ed:89:2d:6e:94:f8:b4:51:2c:1b:b7:5b:63:
          74:ce:c5:05:a6:a9:52:47:f2:56:5e:58:cd:f8:c6:
          a9:1d:54:a6:52:9f:5c:95:4f:27:db:bd:6f:ba:cc:
          23:17:67:aa:3a:12:1b:21:97:32:ce:bf:22:c2:1c:
          2d:4b:a5:c4:99:18:38:96:48:06:9b:2b:98:df:74:
          e3:92:af:86:21:75:ed:77:86:63:af:a2:71:c4:0e:

          a8:ac:1d:dc:26:65:b0:ed:b0:06:50:4b:da:e4:01:

          7a:49:7e:9b:38:1d:c7:2d:57

        Exponent: 3 (0x3)
    X509v3 extensions:
      X509v3 Basic Constraints:
        CA:FALSE
      X509v3 Subject Key Identifier:
        CA:8D:DB:15:B8:A9:42:EC:51:A2:B7:C3:19:76:F7:15:35:1D:C8:9E
      X509v3 Authority Key Identifier:
        DirName:/C=US/ST=GA/L=Alpharetta/O=Stonebranch/CN=Stonebranch
        serial:79:19:7A:72:ED:D5:1F:7B
      X509v3 Key Usage:
        Digital Signature, Non Repudiation, Key Encipherment
    Signature Algorithm: sha1WithRSAEncryption
      b0:b3:0d:8c:06:fe:4a:b0:e8:46:fd:8f:d8:64:d1:5e:11:b3:
      68:43:34:28:08:4b:e0:62:39:c1:6c:06:76:f3:e5:9d:8c:4e:
      15:57:56:d7:bf:92:f3:cf:6a:c8:36:54:28:2d:f9:9f:ad:67:
      44:1a:2e:32:ad:8b:8a:a0:86:64:8d:73:a0:60:46:65:f0:62:
      1f:02:db:c7:7c:99:db:ad:5b:80:3e:e9:b2:88:19:23:15:e6:
      7a:1d:53:e3:51:60:2d:99:0c:20:08:5a:ae:0f:c8:d3:20:a4:
      31:91:8b:a7:c2:c8:7a:ab:6c:2d:18:7a:1e:95:4b:c0:3e:5f:
      f9:cf

```

## Security - Creating Certificates Examples

### Creating Certificates Examples

- [Creating a Certificate Authority \(CA\) Certificate](#)
- [Creating a Certificate](#)

## Creating a Certificate

### Creating a Certificate

There are two steps in creating a certificate:

1. First step is performed by the party that wants the certificate.
2. Second step is performed by the Certificate Authority (CA) that creates the certificate.

#### Step 1

Step one is creating the certificate request. The certificate request will then be sent to the CA that verifies the request and creates the certificate from the request. The command that creates the certificate request also creates a private key. The private key must be secured so that only the entity identified by the certificate request has read access.

The following Universal Certificate command creates:

- Certificate request, which it writes it to file **req.pem**
- Private key, which it writes it to file **pkey.pem**

```
ucert -create request -request_file req.pem -private_key_file pkey.pem -country US -state Maryland
-locality Baltimore
-organization "Acme, Inc." -common_name "Joe Buck"
```

#### Step 2

Step two is for the CA to create a certificate from the request and sign it with the CA's private key.

The following Universal Certificate command creates the certificate and writes it to file **cert.pem**.

```
ucert -create cert -request_file req.pem -cert_file cert.pem -private_key_file cakey.pem -ca_cert_file
cacert.pem
```

## Components

Universal Certificate

## Creating a Certificate Authority (CA) Certificate

### Creating a Certificate Authority Certificate

The first step in creating a certificate hierarchy is creating the root Certificate Authority (CA) certificate. The CA certificate is used to issue user certificates.

A certificate is created by creating a certificate request and then having the CA validate and sign the certificate. Since we are creating a root CA certificate, there is no CA to sign the certificate request, so instead a self-signed certificate is created and the CA flag is set.

The following Universal Certificate command creates:

- Certificate request, which it writes it to file **req.pem**
- Private key, which it writes it to file **cakey.pem**

```
ucert -create request -request_file req.pem -private_key_file cakey.pem -country US -state Maryland
-locality Baltimore
-organization "Acme, Inc." -common_name "Acme CA"
```

It is imperative that the private key file **cakey.pem** is secured so that no one other than the CA has read access. If unauthorized access is gained to the CA's private key, all certificates issued by the CA no longer can be trusted.

The following Universal Certificate command creates the CA certificate and writes it to file **cacert.pem**.

```
ucert -create cert -request_file req.pem -cert_file cacert.pem -private_key_file cakey.pem -ca yes
```

The CA certificate, **cacert.pem**, must be made available to any system that wants to consider the certificates issued by the CA as valid.

### Components

#### Universal Certificate

## Licenses and Copyrights

- [Overview](#)
- [Berkeley DB License](#)
- [OpenSSL License](#)
- [zlib License](#)

### Overview

This page provides the following license files for the third-party libraries used within Workload Automation:

- Berkeley DB License
- OpenSSL License
- zlib License

### 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.
*/

```

## OpenSSL License

The OpenSSL toolkit is used in binary form on z/OS, Windows, UNIX, and IBM i 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.]
*/
```

## **zlib License**

The zlib library is used in binary form on z/OS, Windows, UNIX, and IBM i 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