software AG

Universal Messaging Installation Guide

Version 9.9

October 2015

Table of Contents

Installation Overview

You can install Universal Messaging in two different ways:

- As a self-contained, standalone installation. Refer to the section "Performing the Standalone Installation" on page 7 for details.

- As a product within the webMethods product suite. Refer to the section "Installation using the Software AG Installer" on page 29 for details.

1   Performing the Standalone Installation

Universal Messaging is made up of a server component and a client API used by applications to communicate with the server. The server, known as the Realm Server, maintains all the information regarding resources and P2P services, and the security around these. The Realm Server supports direct connections using normal TCP/IP sockets and Secure Socket Layer (SSL) while also supporting HTTP directly with no additional processes or web server infrastructure required. By directly supporting HTTP the process of configuring a Realm Server for the Internet is greatly simplified.

Universal Messaging is highly portable - written in 100% pure Java. It therefore requires a copy of the Java Development Kit (JDK) or Java Runtime Environment (JRE) to be installed. The recommended release of the required JDK and JRE is 1.8.

In order to make the installation of Universal Messaging a pain free experience we provide our users with a set of options. Installers can run in either GUI or console mode in order to enable it to be run on nearly all systems without the requirement of having some sort of graphical display to perform the task. In addition to that, installers can include a Java Runtime Environment (JRE) or not which significantly affects the size of your download but ensures compatibility and normal operation of the product.

The installers can be downloaded directly from the web site.

This installation guide contains information on the following sections:

Trial and Production Licences

Note for webMethods UM Users

Please use the ninstancemanager tool (see "Universal Messaging Instance Manager" on page 41) to manage your UM installations.

Pre-Installation

Pre-Installation

As mentioned in the installation overview section, when choosing the download package you have the option to download one with a JRE or one without a JRE. If you have chosen a Win32, Linux or Solaris installer that does not include a JRE or the Generic Unix installer you need to ensure that a compatible Java JDK or JRE installation exists prior to running the installer. The recommended release of the required JDK and JRE is 1.8.

The installer assumes that JRE or JDK is configured correctly and can be executed by opening a console/shell and typing the following command:

```
[root@host~]# java -version
java version "1.8.0_31"
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)
[root@host ~]#
```

Performing the Installation

The installation steps heavily depend on your chosen installer.

Win32

After downloading, double-click on install.exe. The installer will attempt to locate a suitable JRE/JDK installation on your machine but if you do not have one, be sure to download the package that includes one.

Linux/Solaris

After downloading open a shell and, cd to the directory where you downloaded the installer. At the prompt type:

```
sh ./install.bin
```

The installer will attempt to locate a suitable JRE/JDK installation on your machine but if you do not have one, be sure to download the package that includes one. Otherwise you may need to download one from your Java VM supplier's web site or contact your OS manufacturer. The installer can operate in GUI or console modes depending on your system configuration. The default mode of operation is console but you can perform a GUI installation by typing

```
sh ./install.bin -i gui .
```

> **Note:**   Note: the VM based installations use a 32-bit version of the JVM. Users wishing to install a 64-bit version of the product are recommended to use the no-VM installers.

Mac OSX

After downloading, double-click on nirvana_osx_novm.zip to uncompress it. This produces a single installer executable called install.app. Double-click again to start the installation using the OSX java configured in your Mac.

Generic Unix

After downloading open a shell and, cd to the directory where you downloaded the installer. At the prompt type:

```
sh ./install.bin
```

The installer will attempt to locate a suitable JRE/JDK installation on your machine. You can download one from your Java VM supplier's web site or contact your OS manufacturer. The installer can operate in GUI or console modes depending on your system configuration. The default mode of operation is console but you can perform a GUI installation by typing

```
sh ./install.bin -i gui
```

Index

Introduction

The first screen that is displayed when the installer is run is the introduction screen which describes the process that is about to start and advises to quit all other running

programs before continuing. An example of how the screens look like is illustrated below for Win32 (GUI mode) and Unix (Console mode).
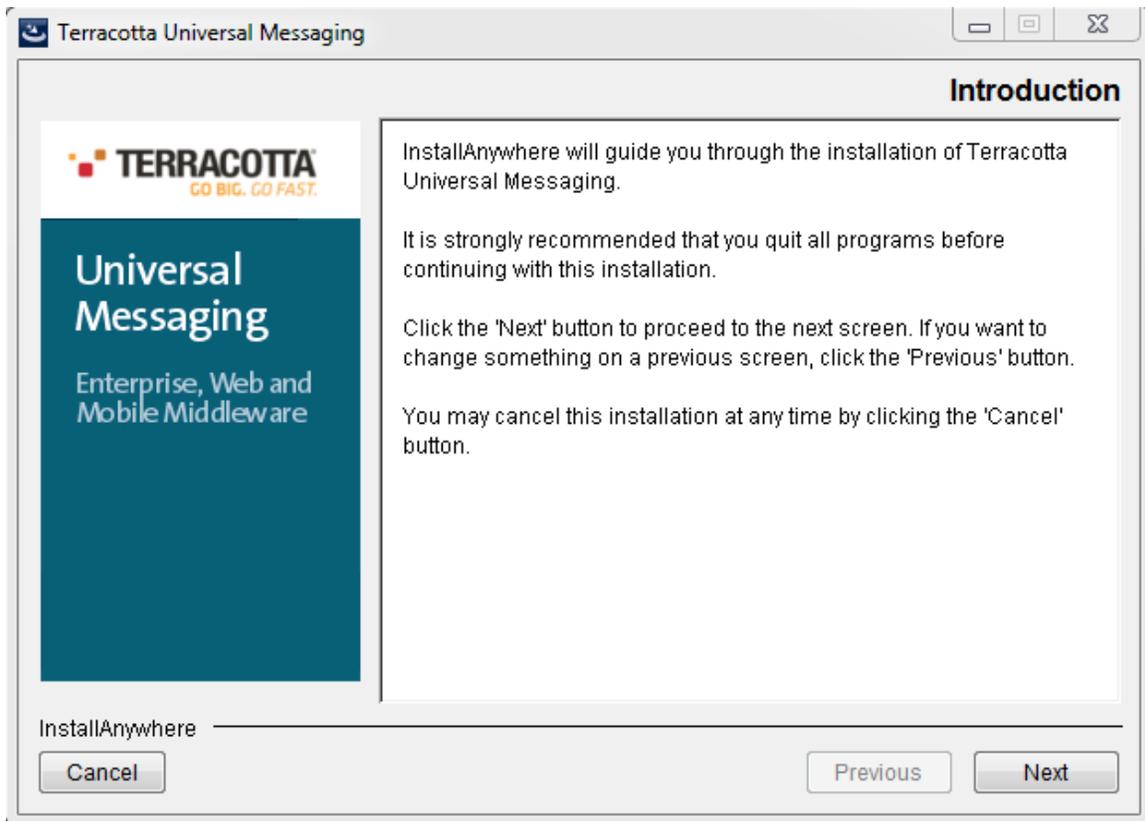

```
Preparing to install...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...
Launching installer...
================================================================================
Choose Locale...
----------------
    1- Deutsch
  ->2- English
    3- Español
    4- Français
    5- Italiano
CHOOSE LOCALE BY NUMBER:
================================================================================
Terracotta Universal Messaging                    (created with InstallAnywhere)
--------------------------------------------------------------------------------
Preparing CONSOLE Mode Installation...
================================================================================
Introduction
------------
InstallAnywhere will guide you through the installation of Terracotta Universal
Messaging.
It is strongly recommended that you quit all programs before continuing with
this installation.
Respond to each prompt to proceed to the next step in the installation.  If you
want to change something on a previous step, type 'back'.
You may cancel this installation at any time by typing 'quit'.
PRESS <ENTER> TO CONTINUE:
```

License Acceptance

The second screen displayed by the installer asks you to accept the Software AG License agreement before continuing with the installation. Accepting the license agreement implies you agree with all terms of use and the installation can continue normally. Rejecting the license agreement will end the installation process. An example of the license agreement as it appears in GUI and console modes is illustrated below.
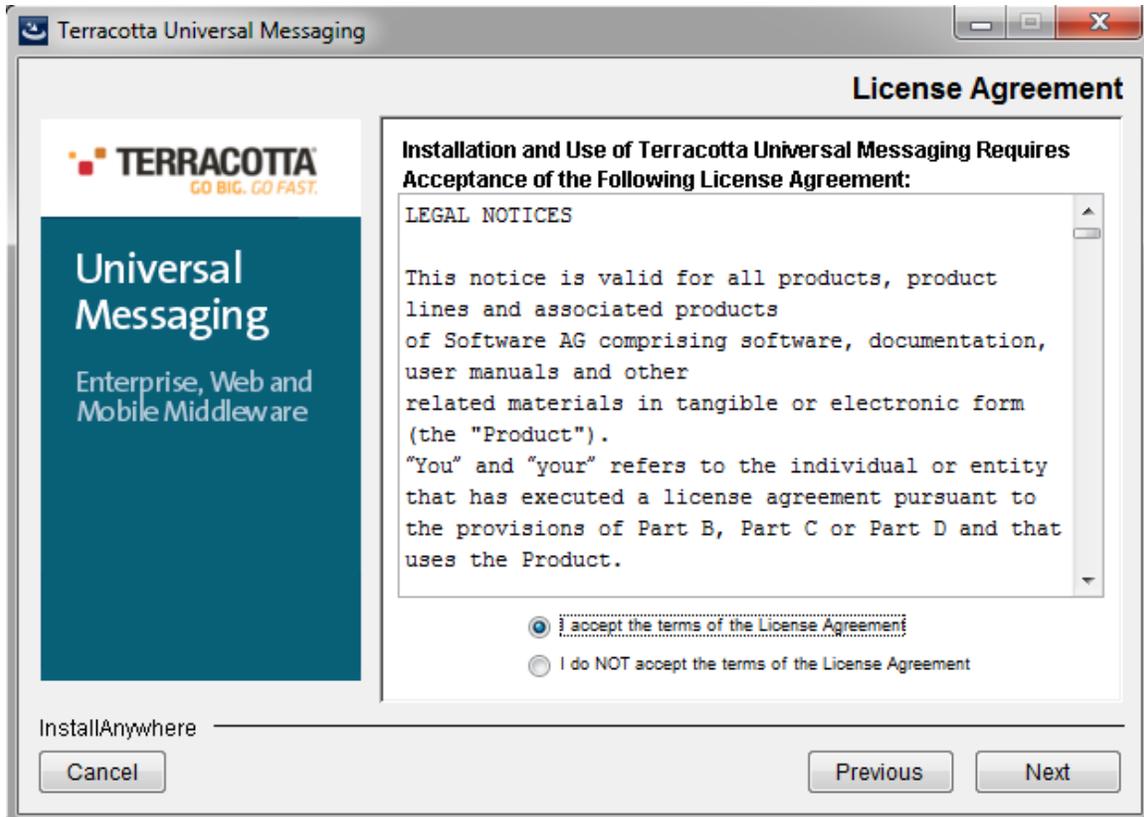

```
================================================================================
License Agreement
-----------------
Installation and use of Terracotta Universal Messaging requires acceptance of
the following License Agreement:
LEGAL NOTICES
This notice is valid for all products, product lines and associated products
of Software AG comprising software, documentation, user manuals and other
related materials in tangible or electronic form (the "Product").
"You" and "your" refers to the individual or entity that has executed a license
agreement pursuant to the provisions of Part B, Part C or Part D and that uses
the Product.
IMPORTANT: PLEASE READ BEFORE INSTALLING THE PRODUCT
PART A: GENERAL TERMS
You are not allowed to install or use the Product without a corresponding
license agreement.
(Full License Text Omitted)
to a third party the outcomes or results of any such exercise, or any
information derived from the outcomes or results of such exercise, without the
additional express written consent of Software AG.
END OF LEGAL NOTICES
```

```
DO YOU ACCEPT THE TERMS OF THIS LICENSE AGREEMENT? (Y/N):
```

Installation Folder

The third screen displayed by the installer asks you to choose the directory where you want the product installed. You can type the name of the directory or click on Choose to browse for a folder. If you type a directory name but wish to restore the default installation folder, you can do so by clicking on the Restore Default Folder button. The following figures illustrate an example of this screen for GUI and console modes.
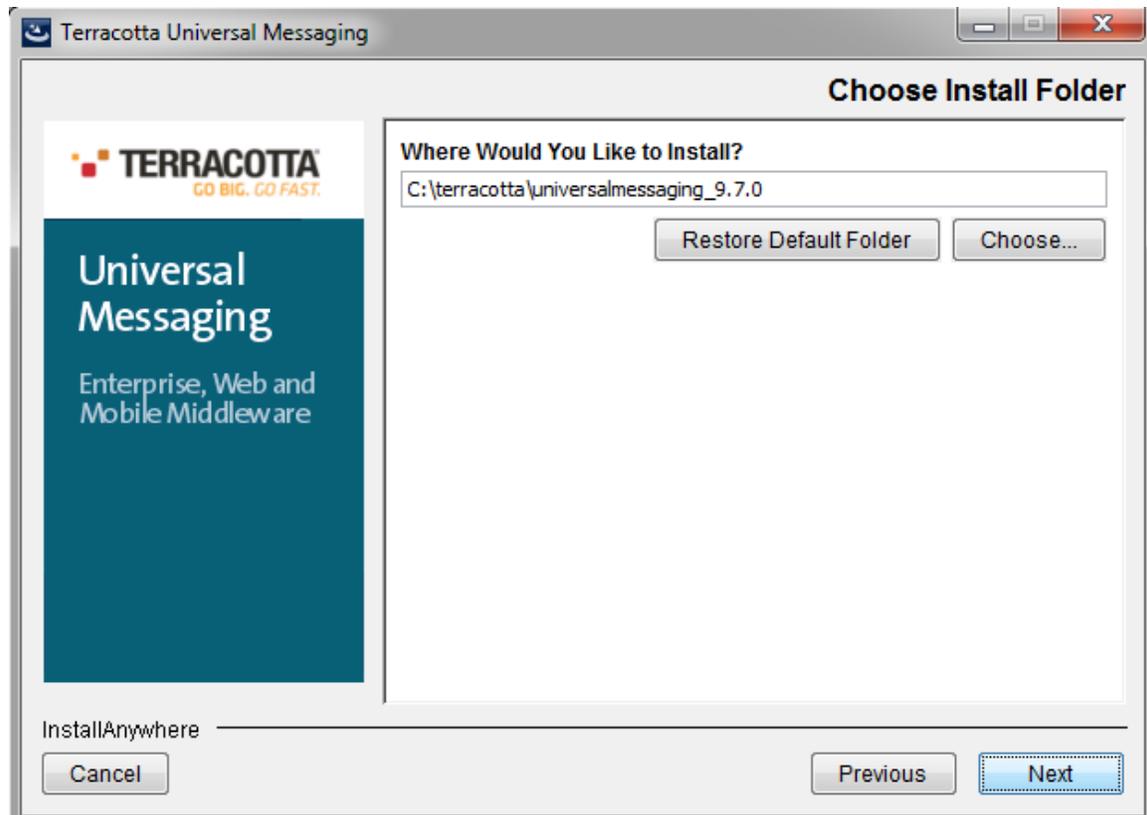

```
===============================================================================
Choose Install Folder
---------------------
Where would you like to install?
  Default Install Folder: /home/username/terracotta/universalmessaging_<version>
ENTER AN ABSOLUTE PATH, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
```

Installation Set

The fourth screen displayed by the installer asks you to specify if you wish to perform a typical installation or a customized one. A customized installation allows you to choose which product features should be installed as well as allows the customization of the Java virtual machine to use irrespective of the one chosen by the installer. The following figures illustrate an example of this screen for GUI and console modes.
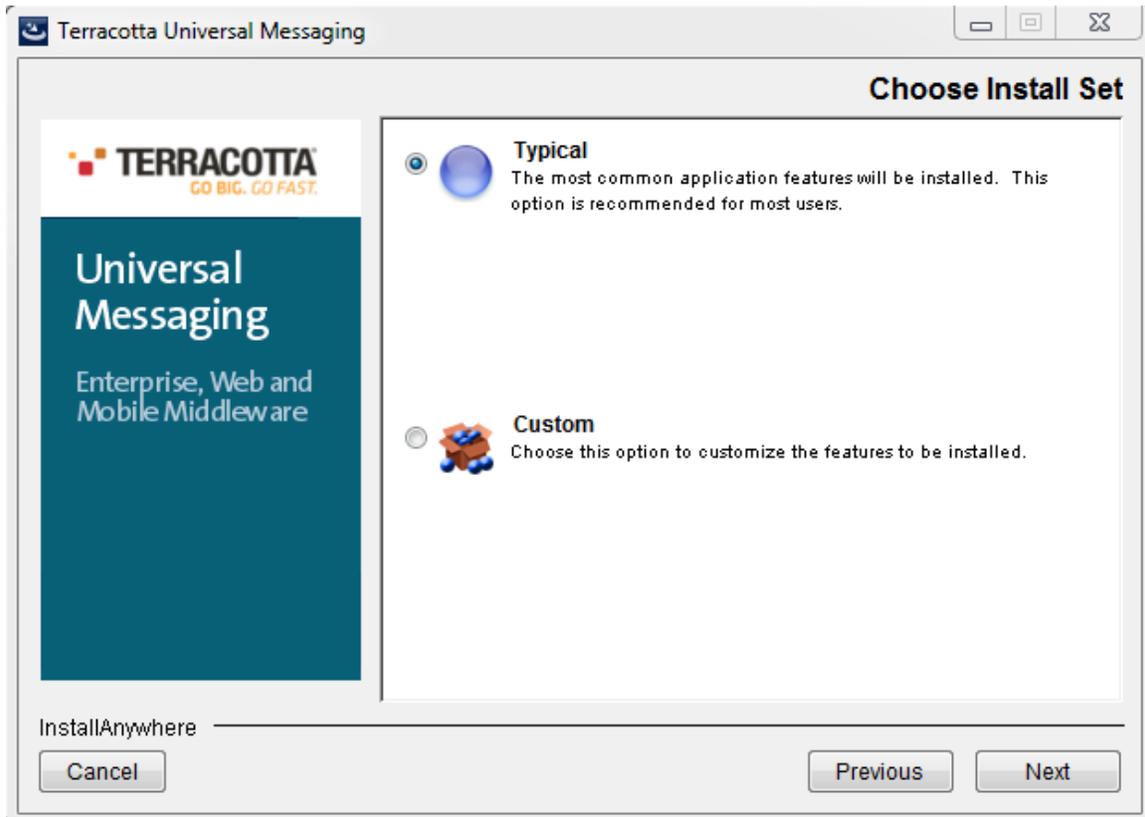
```
================================================================================
Choose Install Set
------------------
Please choose the Install Set to be installed by this installer.
->1- Typical
2- Customize...
ENTER THE NUMBER FOR THE INSTALL SET, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
```

Installation Set Customization

This screen will only be displayed if you previously chose a custom installation set in order to choose the product features that will be installed. Selecting an individual product feature displays additional information to facilitate your choice. The following figures illustrate an example of this screen for GUI and console modes.
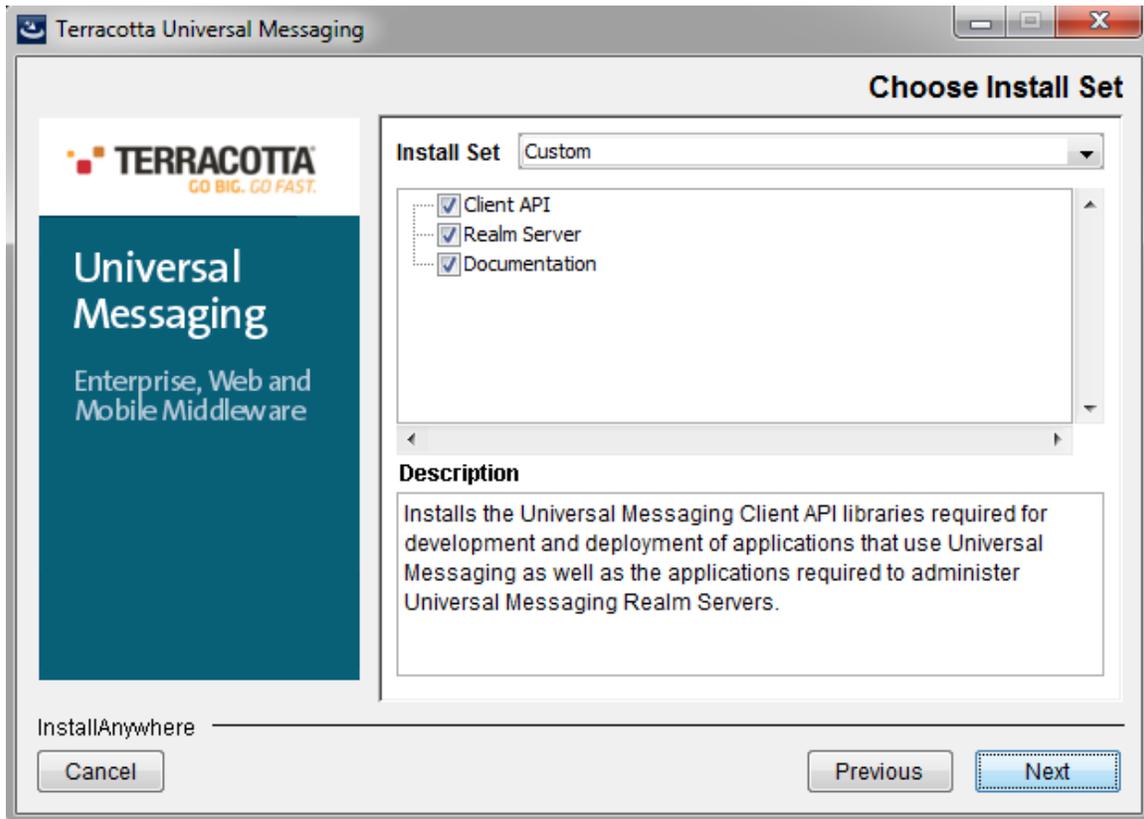
```
================================================================================
Choose Product Features
-----------------------
ENTER A COMMA_SEPARATED LIST OF NUMBERS REPRESENTING THE FEATURES YOU WOULD
LIKE TO SELECT, OR DESELECT. TO VIEW A FEATURE'S DESCRIPTION, ENTER
'?<NUMBER>'. PRESS <RETURN> WHEN YOU ARE DONE:
1- [X] Client API
2- [X] Realm Server
3- [X] Documentation
Please choose the Features to be installed by this installer.
:
```

Realm Bootstrap Configuration

The Universal Messaging Realm server comes with a very powerful remote administration API. All the administration tools provided make use of that API only which means that all functionality is available for use in your applications or enterprise management tools of your choice.

As the administration API operates using any of the 4 available Universal Messaging transport protocols the installer needs to perform bootstrap configuration on the Realm server being installed so that further administration using the tools or API can be easily performed as soon as the installation completes. The following table illustrates a list of possible bootstrap configuration parameters, the default values provided by the installer and a short description.

Bootstrap Parameter	Default Value	Description
Realm Name	Universal Messaging	This is a unique identifier for your Universal Messaging realm. It is very important that this name is unique when using your realm in federated namespaces or cluster configurations.
NHP Interface binding	0.0.0.0	The Universal Messaging realm you are installing will have a bootstrap network interface installed to allow sample applications and administration tools to connect to it. This interface will use the NHP protocol (see protocol overview below). This parameter specifies if the interface should bind to all known IP addresses or a specific one only.
NHP Interface port	9000	The port that the bootstrap interface will listen on.
Data directory	<install folder>/server/ <realm name>/data	This is the folder that all Universal Messaging realm data files, logs and configuration files will be stored when the server starts. This parameter is NOT customizable during the installation.
Log level	4	This is the log level specifying the verbosity of the Universal Messaging realm log file ranging from 0 (very verbose) to 7 (quiet). This parameter is NOT customizable during the installation.

Bootstrap Parameter	Default Value	Description
Initial Memory	256MB	This is the initial amount of memory the Universal Messaging realm will use. This parameter is NOT customizable during the installation.
Max Memory	512MB	This is the maximum amount of memory the Universal Messaging realm can use if needed. This parameter is NOT customizable during the installation

During the installation process, the first 4 bootstrap parameters can be customized. The remaining can be customized by modifying the nserver.lax configuration file. The following figures illustrate an example of the bootstrap customization screens for GUI and console modes.
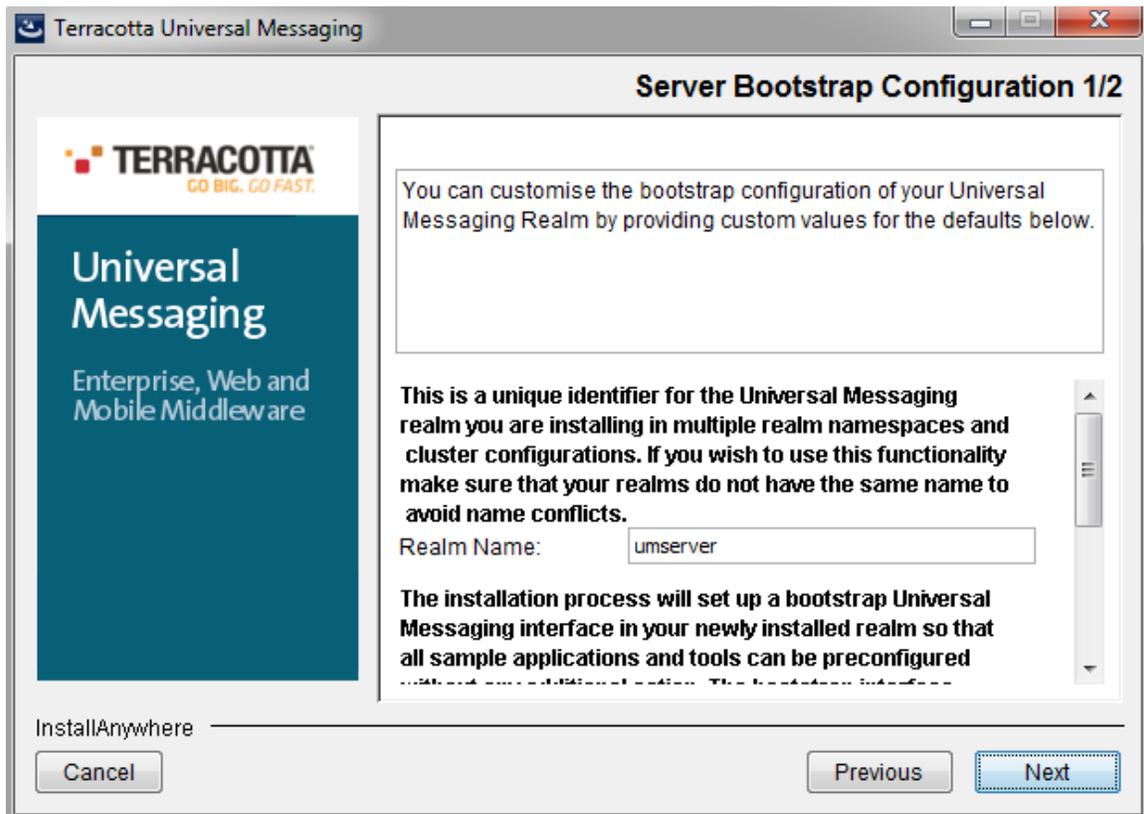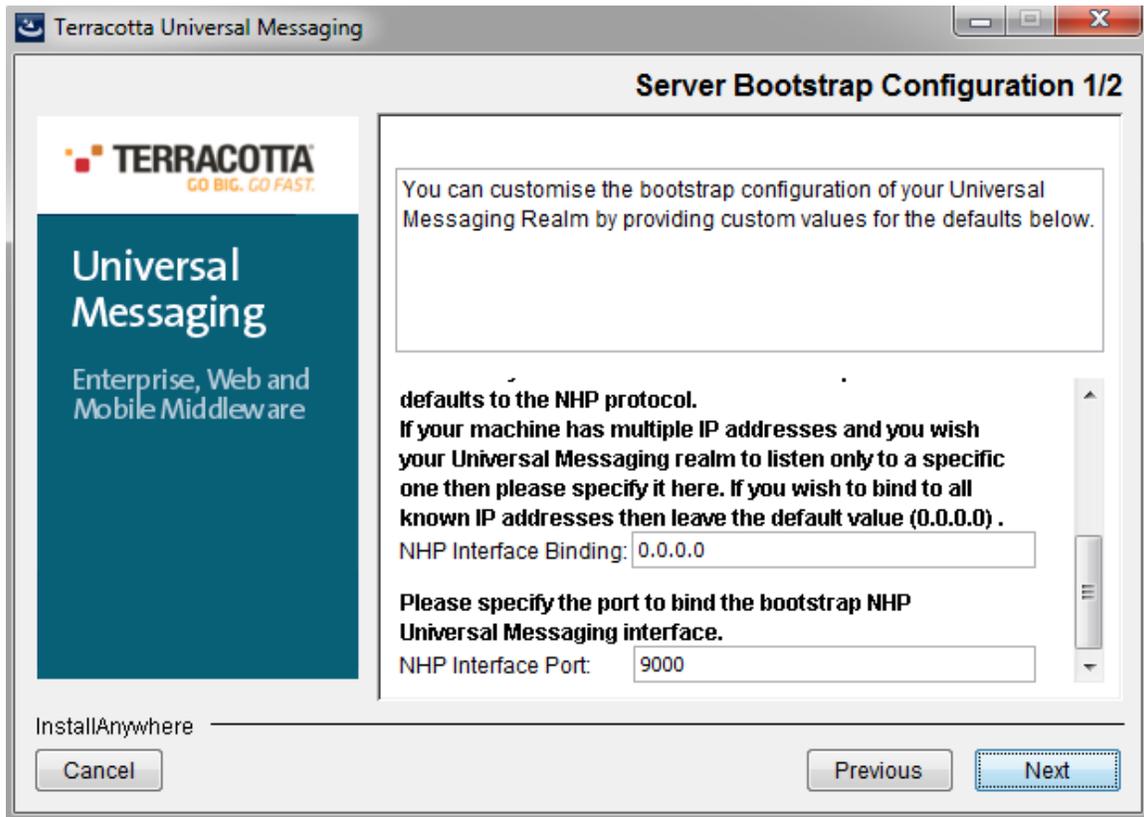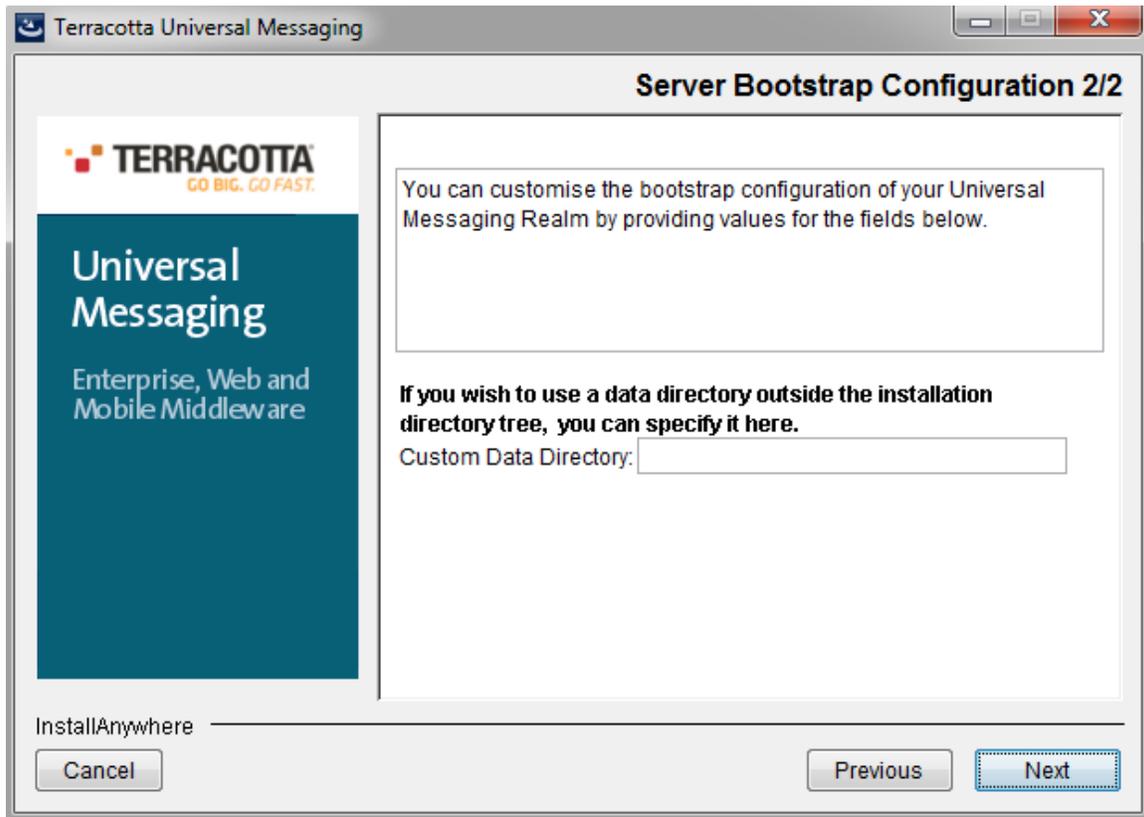
```
===============================================================================
Server Bootstrap Configuration - Realm Name
---------------------------------------------
This is a unique identifier for the Universal Messaging realm you are
installing in multiple realm namespaces and cluster configurations.
If you wish to use this functionality make sure that your realms do not
have the same name to avoid name conflicts.
Realm Name: (DEFAULT: umserver):
===============================================================================
Server Bootstrap Configuration - NHP Interface Binding
------------------------------------------------------
The installation process will set up a bootstrap Universal Messaging
interface in your newly installed realm so that all sample applications
and tools can be preconfigured without any additional action. The
bootstrap interface defaults to the NSP protocol.
If your machine has multiple IP addresses and you wish your Universal
Messaging realm to listen only to a specific one then please specify it
here. If you wish to bind to all known IP addresses then leave the
default value (0.0.0.0) .
NHP Interface Binding: (DEFAULT: 0.0.0.0):
===============================================================================
Server Bootstrap Configuration - NHP Interface Port
---------------------------------------------------
Please specify the port to bind the bootstrap NHP Universal Messaging interface.
NHP Interface Port: (DEFAULT: 9000):
===============================================================================
Server Bootstrap Configuration - Custom Data Directory
------------------------------------------------------
If you wish to use a data directory outside the installation directory tree,
you can specify it here.
Custom Data Directory: (DEFAULT: ):
```

Protocol Overview

Universal Messaging is capable of using multiple transport protocols depending on the deployment and security requirements in place. Although the bootstrap interface defaults to the Universal Messaging HTTP Protocol (NHP) more interfaces can be added using the Universal Messaging Enterprise manager or an application using the Universal Messaging Admin API.

Universal Messaging Socket Protocol

Out of the box the installer configures the Realm to have a Universal Messaging Socket Protocol (nsp) adapter. As the name suggests NSP uses TCP/IP sockets directly and is useful for servers and clients on a secure wan/lan configuration where encryption of data and firewall traversal are not required.

Universal Messaging HTTP Protocol

In order to facilitate communications in environments where security policies enforce network restrictions, the Realm can use the Universal Messaging HTTP (Protocol) (nhp). This driver supports HTTP and provides a mechanism for clients and other servers to communicate with the server through proxy servers, firewalls and essentially the Internet. It would not be prudent to run this protocol directly on the Internet without SSL to provide a level of security. It can be useful however, in traversing internal firewalls to get to a DMZ hosted Universal Messaging Realm server thus allowing trusted clients and servers access to the external facing Realm.

Universal Messaging Secure Socket Protocol

The Universal Messaging Secure Socket Protocol (nsps) is identical to the Universal Messaging Socket Protocol but using SSL sockets rather than plain TCP/IP ones. It can be configured to require client side authentication or not and it requires a Java VM with Secure Socket Layer installed. For more information on that please see the Secure Socket Layer Installation section below.

Universal Messaging Secure HTTP Protocol

The Universal Messaging Secure HTTP Protocol (nhps) is identical to the Universal Messaging HTTP Protocol but it uses SSL sockets rather than plain TCP/IP ones. It can be configured to require client side authentication or not and it requires a Java VM with Secure Socket Layer installed. For more information on that please see the Secure Socket Layer Installation section below.

Secure Socket Layer Installation

In order for the server to support the NSPS and NHPS protocols the Java Secure Socket Extension (JSSE) package is required. Please note that JSSE must be installed in the JVM before the server can utilise it (note that JSSE it is already built into recent JVM's, and as such does not require any further action). Should you need to install JSSE however, the installation can be found at http://Java.sun.com/products/jsse/INSTALL.html. With

recent JVMs, Universal Messaging Realm server can also be configured to use any JSSE compliant SSL provider.

Java Virtual Machine Customizations

This screen will only be displayed if you previously chose a custom installation set. It allows you to specify which Java Virtual machine will be used by the product once the installation is complete, irrespective of the virtual machine chosen by the installer to perform the installation. The following figures illustrate an example of the screen for GUI and console modes:
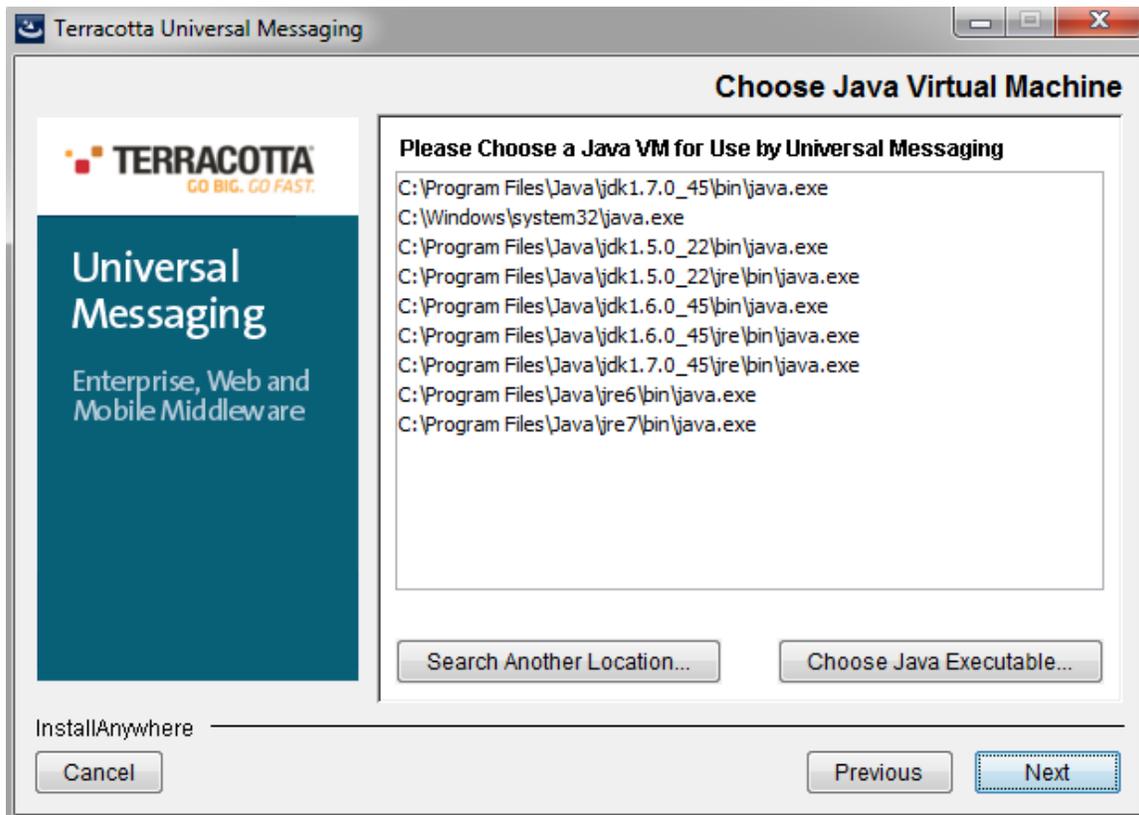

```
================================================================================
Choose Java Virtual Machine
---------------------------
Please choose a Java VM for use by Universal Messaging
  ->1- /usr/bin/java
    2- /usr/local/java/jdk<jdk_version>/bin/java
    3- Choose a Java VM already installed on this system
ENTER THE NUMBER FOR THE JAVA VM, OR PRESS <ENTER> TO ACCEPT THE
      CURRENT SELECTION:
```

Shortcut/Link Customization

This screen will only be displayed if you previously chose a custom installation set. It allows you to specify the location that will be used to install shortcuts/links. The following figures illustrate an example of this screen for GUI and console modes.
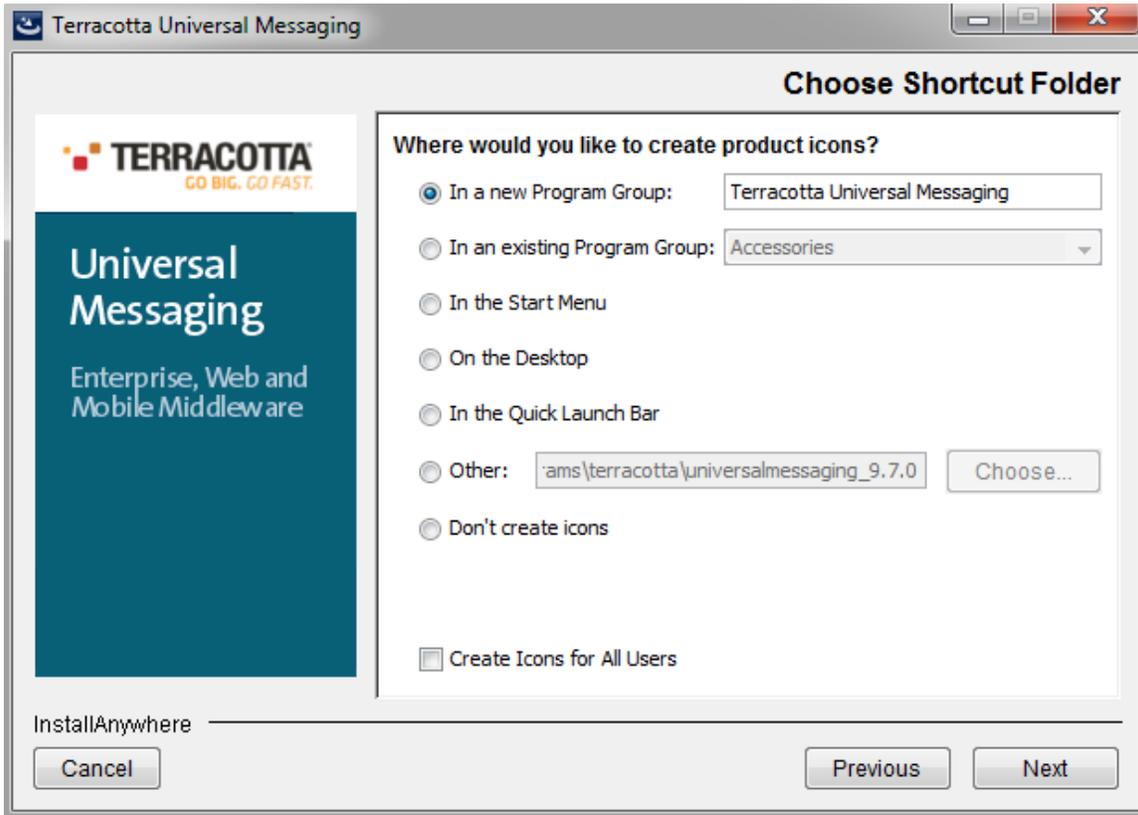
```
================================================================================
Choose Link Location
-------------------
Where would you like to create links?
->1- Default: /home/username/terracotta/universalmessaging_<version>/links
2- In your home folder
3- Choose another location...
4- Don't create links
ENTER THE NUMBER OF AN OPTION ABOVE, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
:
```

Pre-Installation Summary

The is the last screen displayed by the installer showing a pre-installation summary of the actions that will be performed if you choose to continue. The following figures illustrate an example of this screen for GUI and console modes.
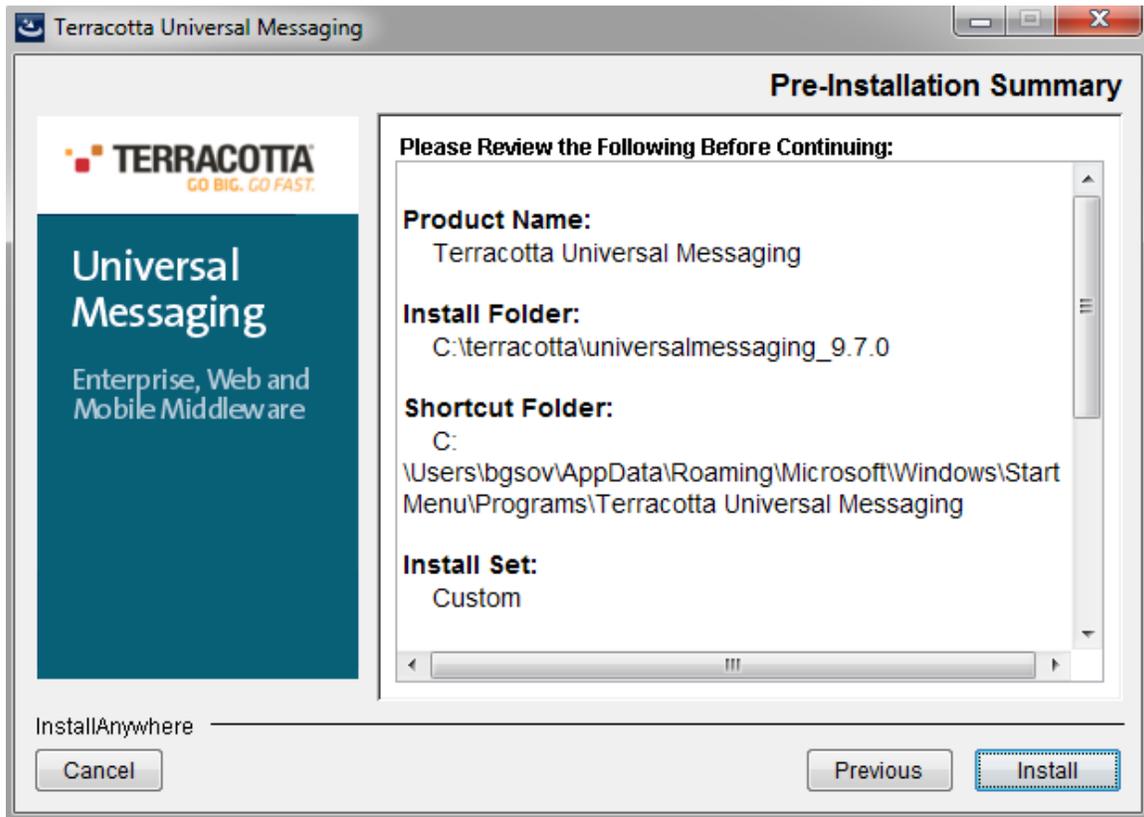
```
================================================================================
Pre-Installation Summary
------------------------
Please Review the Following Before Continuing:
Product Name:
    Terracotta Universal Messaging
Install Folder:
    /home/chefo/terracotta/universalmessaging_<version>
Link Folder:
    /home/chefo/terracotta/universalmessaging_<version>/links
Install Set:
    Custom
Product Features:
    Client API,
    Realm Server,
    Documentation
Java VM to be Used by Installed Product:
    /usr/local/java/jdk<jdk_version>
Disk Space Information (for Installation Target):
    Required:  276,032,018 Bytes
    Available: 13,249,114,112 Bytes
PRESS <ENTER> TO CONTINUE:
```

Running a Silent Installation

There may be occasions where you want to install Universal Messaging in silent mode, ie where you can script the install, and not require any user input. This section describes

how to run the installer in silent mode. There are currently some outstanding issues with the InstallAnywhere installer bundle that causes the silent install to not correctly work without making some slight alterations to the installer output. These steps are described in the following sections.

To begin, you need to have already installed Universal Messaging somewhere (you will need to run the installer with '-r installer.props' in order to first generate the response file required for a silent install). Once you have the response file (installer.props but it can be called anything you wish), there are some modifications that need to be made to this file in order to successfully run the silent install. So, assuming you have the installer.props file in the same location as the unix Universal Messaging install binary, you will need to edit the installer.props and make the following changes:

Custom Variables

The installer.props contains all the custom information provided during the install, such as installation directory, realm name etc. Due to an outstanding issue with the installer software, some of these variables should in fact be appended with '_1'. For example, CUSTOM_REALM_NAME, should in fact be CUSTOM_REALM_NAME_1. All CUSTOM variables require the additional _1 to be appended.

Escaping Characters

Another issue with the installer, is that the actual variables stored as CUSTOM install parameters, contain unnecessary escape characters. For example,

CUSTOM_REALM_NAME=\"nirvana\"

The escape characters need to be removed, so after completing the Custom Variable step and the escaped characters changes you should end up with something that resembles:

■ CUSTOM_REALM_NAME_1=nirvana

■ CUSTOM_NSP_BINDING_1=0.0.0.0

■ CUSTOM_NSP_PORT_1=9000

Remove Chosen List

The final configuration change that is required is to ensure that all components are installed during the silent install. To do this, simply comment out the following:

■ CHOSEN_FEATURE_LIST=Client ,Realm_SRV,Doc

■ CHOSEN_INSTALL_FEATURE_LIST=Client ,Realm_SRV,Doc

■ CHOSEN_INSTALL_SET=Typical

So it looks like:

■ #CHOSEN_FEATURE_LIST=Client ,Realm_SRV,Doc

■ #CHOSEN_INSTALL_FEATURE_LIST=Client ,Realm_SRV,Doc

■ #CHOSEN_INSTALL_SET=Typical

Performing A Silent Install

Once you have a custom installer properties file, this can be used as a template to create customisable installs for any server instance you wish to script. For example, the installer.props file that contains all of the changes described above can now be copied and modified to contain the boot strap information required for any nirvana realm.

Once you are happy that you have modified the properties file to contain all of the information required, you can now run the install in silent mode, by using the following command:

```
sh ./nirvana_linux_novm.bin -i silent -f installer.props
```

replacing the installer.props file with the name of your custom configuration file.

Command Prompts

Command Prompts

In order to make it easy for new users to test the rich functionality provided by Universal Messaging, the Universal Messaging installation includes compiled versions of all the sample applications as well as native launchers that wrap them and use environment variables to minimise the input required by the user. A command prompt is therefore a console/shell with some environment variables already set.

Client Command Prompt

The client command prompt is a console/shell with environment variables set by a client environment script. Examples of such environment variables include RNAME, PATH, CLASSPATH, Certificate stores etc. On Win32 platforms a client command prompt is opened using the shortcut in the start menu while on Unix platforms it is opened by executing the appropriate softlink in the links folder under the install directory:

```
$ cd ~/terracotta/universalmessaging_<version>/links/Client/umserver
$ ./Java\ Examples\ Command\ Prompt
```

where *<version>* is the product version number.

Alternatively, to run C++ samples:

```
$ ./C++\ Examples\ Command\ Prompt
```

You are now ready to run sample client applications and sample administration (AdminAPI) applications with the minimum possible input.

Server Command Prompt

The server command prompt is a console/shell with environment variables set by a server environment script. Examples of such environment variables include PATH, CLASSPATH, server certificate stores etc. On Win32 platforms a server command prompt is opened using the shortcut in the start menu while on Unix platforms it is opened by executing the appropriate softlink in the links folder under the install directory:

```
$ cd ~/terracotta/universalmessaging_<version>/links/Server/umserver
$ ./Command_Prompt
```

You are now ready to run server side administration commands such as start a realm server, stop a realm server etc.

Starting the Realm Server

Starting the Universal Messaging Realm server depends on the installation operating system.

For win32 operating systems, the installer installs a shortcut called *Start Universal Messaging Realm Server* . Furthermore, the server command prompt can be used to open a console window and the server can be started by typing `nserver`.

Alternatively you can register the Realm as an NT service and control operation through the service control panel.

For Linux/Solaris/Generic Unix operating systems, starting the Realm can be done using the softlink inside the `Server` directory:

```
$ cd ~/terracotta/universalmessaging_<version>/links/Server/umserver/
$ nohup ./Start\ Universal\ Messaging\ Realm\ Server &
```

where *<version>* is the product version number.

For the personal Java installation simply click the server link in the pjbin or the evbin, depending on the Personal Java JRE of your choice.

At this point the server should have started and is now ready for operation. To confirm this it is easy to check the Realm Server log file and ensure there are no errors being reported. Check the log file for a completion message such as the following:

```
[Mon Apr 07 10:59:22 BST 2014],Startup: Realm Server Startup sequence completed
```

As the Realm Server starts up it reports the current log level settings, currently set to 1000000 characters and a log level of 4. The log level was set during the installation, a level of 0 reports every action the server does and can roll log files every minute on a busy realm.

The Realm Server then reports the JRE environment and the currently installed security providers. These are important if running SSL and cause the majority of configuration problems. After this the server then reloads all the configuration parameters, channels and topics and then is ready to accept connections.

If there are any problems binding to a port or creating an SSL instance these exceptions are reported into the Realm Server log file with as much detail as the server can produce.

To test that the Realm Server is up and ready to accept connections a simple test is to request information about the realm itself. To do this you need to open a client command prompt, and then issue the command ngetrealms.

To open a client command prompt on a Win32 installation, click on the Command Prompt shortcut in your Start Menu.

To open a client command prompt on a Linux/Solaris/Generic Unix installation, open a console and source the Command Prompt script as follows:

```
$ cd ~/terracotta/universalmessaging_<version>/links/Client/umserver
$ ./Java\ Examples\ Command\ Prompt
```

where *<version>* is the product version number.

This will set up your environment so that you can communicate with your Realm server.

Stopping the Realm Server

Stopping the Universal Messaging Realm server depends on the installation operating system.

For win32 operating systems, the installer installs a shortcut called *Stop Universal Messaging Realm Server* . Furthermore, the server command prompt can be used to open a console window and the server can be stopped by typing `nstopserver`.

If however you have started the Realm from service control panel then stopping it should also be done from there.

For Linux/Solaris/Generic Unix operating systems, stopping the Realm can be done using the softlink inside the `Server` directory:

```
$ cd ~/terracotta/universalmessaging_<version>/links/Server/umserver
$ ./Stop\ Universal\ Messaging\ Realm\ Server
```

where *<version>* is the product version number.

At this point the server will try and close down all client links and resources and P2P services. This may take some time to perform a graceful shutdown due to the Operating System resources that the Realm Server uses. The RealmServer will automatically perform a complete shutdown within 10 seconds if the graceful shutdown has not yet completed at which the Realm Server will perform an immediate shutdown. The Realm Server will generate a thread dump which will be written to the log file prior to shutdown.

The Realm Server will then log the shutdown within the server log and can be confirmed as complete by the following entries

```
[Tue Apr 08 09:26:58 BST 2014],Shutdown: Realm Server shutdown sequence started
[Tue Apr 08 09:26:58 BST 2014],Shutdown: Removed all logger listeners
[Tue Apr 08 09:26:58 BST 2014],Shutdown: Disabling client requests
[Tue Apr 08 09:26:58 BST 2014],Shutdown: Stopping Realm status updates
[Tue Apr 08 09:26:58 BST 2014],Shutdown: Stopping Interface Manager
[Tue Apr 08 09:26:58 BST 2014],Shutdown: Closing all accept handlers
[Tue Apr 08 09:26:58 BST 2014],Shutdown: Closing all sessions with connected clients
[Tue Apr 08 09:26:58 BST 2014],Shutdown: Stopping Cluster management
[Tue Apr 08 09:26:58 BST 2014],Cluster> Cluster Agent restoring client request table
[Tue Apr 08 09:26:58 BST 2014],Cluster> Cluster Agent finished restoring client request table
[Tue Apr 08 09:26:58 BST 2014],--------- Log File Closed ---------
```

2 Installation using the Software AG Installer

As an alternative to installing Universal Messaging using the standalone installation described in the previous section, you can install Universal Messaging using the Software AG Installer. The availability of this alternative may depend on your product licensing options. The Software AG Installer is a generic tool for installing Software AG products.

Universal Messaging is typically licensed as part of a product bundle, so when you run the Software AG Installer, a dialog allows you to select the appropriate bundle. A subsequent dialog shows the individual products that you can install from the bundle, including Universal Messaging.

Overview of the Documentation for using the Software AG Installer

For the installation using the Software AG Installer, refer to the following documents:

■ **Using the Software AG Installer**

This document describes how to use the Software AG Installer tool. The usage of the Software AG Installer is the same for all products, so the documentation of the Software AG Installer does not refer explicitly to Universal Messaging.

To access the document *Using the Software AG Installer*, do the following:

1. Log in to the Software AG documentation web site at http://documentation.softwareag.com/, using the Empower login ID and password that you have received by email when you licensed the product.

2. Select the link for the Software AG Installer.

3. The selected page lists several versions of the installer documentation, each shown with a release date. Select the version of the installer documentation that corresponds to the release date of the Software AG Installer you are using. The release date of the Software AG Installer is generally included in the file name of the downloaded executable file. You can also find the release date of the Software AG Installer by clicking the "About" link when you run the Software AG Installer.

■ **Installing Software AG Products**

This document contains specific installation information about many Software AG products, including Universal Messaging.

The most recent version of the document *Installing Software AG Products* is available in the documentation web site using the following URL:

http://documentation.softwareag.com/webmethods/Installing_Software_AG_Products.htm

■ **Upgrading Software AG Products**

If you are upgrading from a previous product version, refer also to the document. This document contains information about how to upgrade an existing Software AG product version to a new version.

The most recent version of the document *Upgrading Software AG Products* is available in the documentation web site using the following URL:

http://documentation.softwareag.com/webmethods/
Upgrading_Software_AG_Products.htm

If you need to access versions of the documents *Installing Software AG Products* and *Upgrading Software AG Products* for previous product releases, proceed as follows:

1. From the starting page at http://documentation.softwareag.com/, follow the "webMethods" link.

2. Navigate to the webMethods Product Suite, then select the Suite version number that matches the Universal Messaging version number.

3. Select the link that deals with installation topics.

Specifying a different JDK/JRE Location

The installation of Universal Messaging, when performed using the Software AG Installer, includes the installation of a version of the JDK.

If you already have a JDK installed, you can change the Universal Messaging configuration to use your own installed JDK. Since Universal Messaging only requires the JRE component of the JDK, you can instead configure Universal Messaging to use your own installed JRE.

The JDK/JRE used by Universal Messaging is configured in the following configuration files. If you change the setup for the JDK/JRE in any of these files, you must make the same change in the other files also.

- *<Software_AG_directory>* /UniversalMessaging/server/*<instanceName>* /bin/ Server_Tools_Common.conf

- *<Software_AG_directory>* /UniversalMessaging/server/*<instanceName>* /bin/ Server_Common.conf

- *<Software_AG_directory>* /UniversalMessaging/java/*<instanceName>* /bin/ Admin_Tools_Common.conf

- *<Software_AG_directory>* /UniversalMessaging/java/*<instanceName>* /bin/ Samples_Common.conf

where *<Software_AG_directory>* is the default location for the installation of products using the Software AG Installer, and *<instanceName>* is the name of your Universal Messaging server. During the installation, a default server called "umserver" is created, so configuration files exist for this server.

In all cases, the setting to be updated is "wrapper.java.command=". To make the change, proceed as follows:

1. Open the configuration file in a text editor.

2. Locate the line that begins with "wrapper.java.command=". This line points to the current location of the JDK/JRE, for example:

    ```
    wrapper.java.command=C:\SoftwareAG\jvm\jvm\bin\java.exe
    ```

3. Change the location to the location of your own installed JDK/JRE, for example:

    ```
    wrapper.java.command=C:\MyWorkArea\MySetup\jvm\jvm\bin\java.exe
    ```

4. Save and close the file.

5. Make the same change in the other configuration files.

6. Stop and restart Universal Messaging.

You can use the Instance Manager to create a new instance of the Universal Messaging server. Each time you create a new instance, the following templates are used to create configuration files for the new instance. Therefore, if you wish to use your own installed JDK/JRE for all future instances of the server, you should modify the setting of `wrapper.java.command` in these templates:

■ *<Software_AG_directory>*/UniversalMessaging/server/templates/ Server_Tools_Common.conf

■ *<Software_AG_directory>*/UniversalMessaging/server/templates/ Server_Common.conf

■ *<Software_AG_directory>*/UniversalMessaging/java/EMtemplates>/ Admin_Tools_Common.conf

■ *<Software_AG_directory>*/UniversalMessaging/java/templates/ Samples_Common.conf

Note: Even if you use your own JDK/JRE, you should NOT remove the JDK that is installed with Universal Messaging; this JDK is required to run the Software AG Uninstaller, regardless of any other JDK/JRE you use.

3 **Common Post-Installation Procedures for Standalone and Suite Installation**

The information in the following sections describes procedures that you can use after the installation has completed. The procedures apply to both the standalone installation (described in section "Performing the Standalone Installation" on page 7 and the Suite installation (described in section "Installation using the Software AG Installer" on page 29).

Server Memory Modes

Server Memory Modes

The performance and behaviour of the Realm Server is inseparably linked to the amount of maximum heap memory allocated to the Java VM hosting it. The Realm Server is capable of scaling depending on the hardware platform it is hosted on, and that is determined by the memory available to the Java VM. The Realm detects this and switches its mode of operation to *Small Memory Mode*, *Medium Memory Mode* or *Large Memory Mode*.

Small Memory Mode

Allocating 16MB or less of heap memory to the Java VM hosting the Universal Messaging Realm will make it operate in small memory mode. This is confirmed at start-up by a log entry like the following:

```
Audit,Setting Server mode to Small Memory Mode
```

The Universal Messaging Realm small memory mode should be used when running a Realm on mobile or embedded devices, or other machines with very limited memory resources available. Apart from limited memory available to store events in reliable channels, all thread pooled sub systems are changed to have only one thread. It is therefore recommended that persistent channels should always be used on such Realms. The performance will also be reduced by the fact that all caching is disabled in this mode.

All of the functionality provided by the innovative Universal Messaging server side Realms are available in the small memory mode and hence on handheld devices etc.

Medium Memory Mode

Allocating 16MB or more of heap memory to the Java VM hosting the Universal Messaging Realm will make it operate in medium memory mode. This is confirmed at start-up by a log entry like the following:

```
Audit,Setting Server mode to Medium Memory Mode
```

The Universal Messaging Realm medium memory mode should be used when running Realms on development or where memory is at a premium. All thread pooled sub systems will start up with our recommended default values for this mode. Tuning the Realm to higher values for those sub systems will increase the Realm's memory requirements and increase caching age values.

Large Memory Mode

Allocating 70MB or more of heap memory to the Java VM hosting the Universal Messaging Realm will make it operate in large memory mode. This is confirmed at start-up by a log entry like the following:

```
Audit,Setting Server mode to Large Memory Mode
```

The Universal Messaging Realm large memory mode should be used when running Realms on development, staging or production environments, or when using reliable channels. All thread pooled sub systems will start up with our recommended default values for this mode.. Tuning the Realm to higher values for those sub systems will increase the Realm's memory requirements and increase caching age values.

Client Deployment

Client Deployment

Depending on the functionality used by your Universal Messaging application, different jar files are required. This following table illustrates the deployment dependencies between the jar libraries installed by the Universal Messaging installer.

JAR File	Description	Dependency
nClient.jar	Provides Universal Messaging Client functionality (Pub/Sub and Queues)	None
nP2P.jar	Provides Universal Messaging Peer 2 Peer functionality	nClient.jar
nJMS.jar	Provides Universal Messaging Provider for JMS functionality	nClient.jar
nAdminAPI.jar	Provides Universal Messaging Administration & Monitoring functionality	nClient.jar, nP2P.jar
nAdminXMLAPI.jar	Provides Universal Messaging Configuration XML Import / Export functionality	nClient.jar, nP2P.jar, nAdminAPI.jar
nEnterpriseManager.jar	Contains the Enterprise Manager tool	nClient.jar, nP2P.jar, nAdminAPI.jar,

JAR File	Description	Dependency
		nAdminXMLAPI.jar (Optional)
nServer.jar	Contains the Universal Messaging Realm Server	None
Universal Messaging.tlb	Provides Universal Messaging Client Functionality for ActiveX applications	nClient.jar & Universal Messaging ActiveX Bridge

JMS Configuration

The Universal Messaging Realm server is designed to automatically support applications that use the provided Universal Messaging Provider for JMS. Such applications however need a valid JNDI context configuration in order to remain vendor neutral.

Universal Messaging features a JNDI service provider that can operate using any of the Universal Messaging transport protocols (nsp, nhp, nsps and nhps) as well as tools that allow configuration of the latter or any JNDI context is required from LDAP to NIS. The Universal Messaging JNDI provider uses a channel called /naming/defaultContext to store JMS references and the implementation class is com.pcbsys.nirvana.nSpace.NirvanaContextFactory.

JNDI configuration can be performed in 2 ways. The first is by using a command line application (with full source code provided) called JMSAdmin. For more information about how to use this application please check the appropriate developer guide section. The second is by using the realm JNDI configuration panel in the Universal Messaging Enterprise Manager.

On the client side, you would need to set the standard java.naming.factory.initial key to the aforementioned com.pcbsys.nirvana.nSpace.NirvanaContextFactory and pass the JNDI provider URL of the realm under the nirvana.provider.url key or the standard java.naming.provider.url key. Note that the JNDI API suggests two ways of defining these properties - either by setting them as system properties, or by passing them in a JDNI environment Hashtable argument. When the NirvanaContextFactory is searching for the provider URL, it will by default first check the system properties, and if not found, it will consult the environment Hashtable argument. To reverse the order of lookup, you can set a system property key nirvana.provider.urlpref.sysprops to the value "N".

How to access the Universal Messaging log file

The Universal Messaging log file can be accessed using the Universal Messaging Enterprise Manager GUI by clicking on the LogFile Tab. This provides access to

the Universal Messaging logfile itself as well as the log files associated with the Universal Messaging Realm Plugins. To switch between the available log files select the appropriate tab from the bottom of the logs panel.

Universal Messaging log files can also be accessed programmatically using the Universal Messaging Admin API.

If multiple Realms have been added to the namespace then the log file for each can be accessed by clicking on that realm in the namespace and then selecting the Monitoring tab which will by default display the log panel.
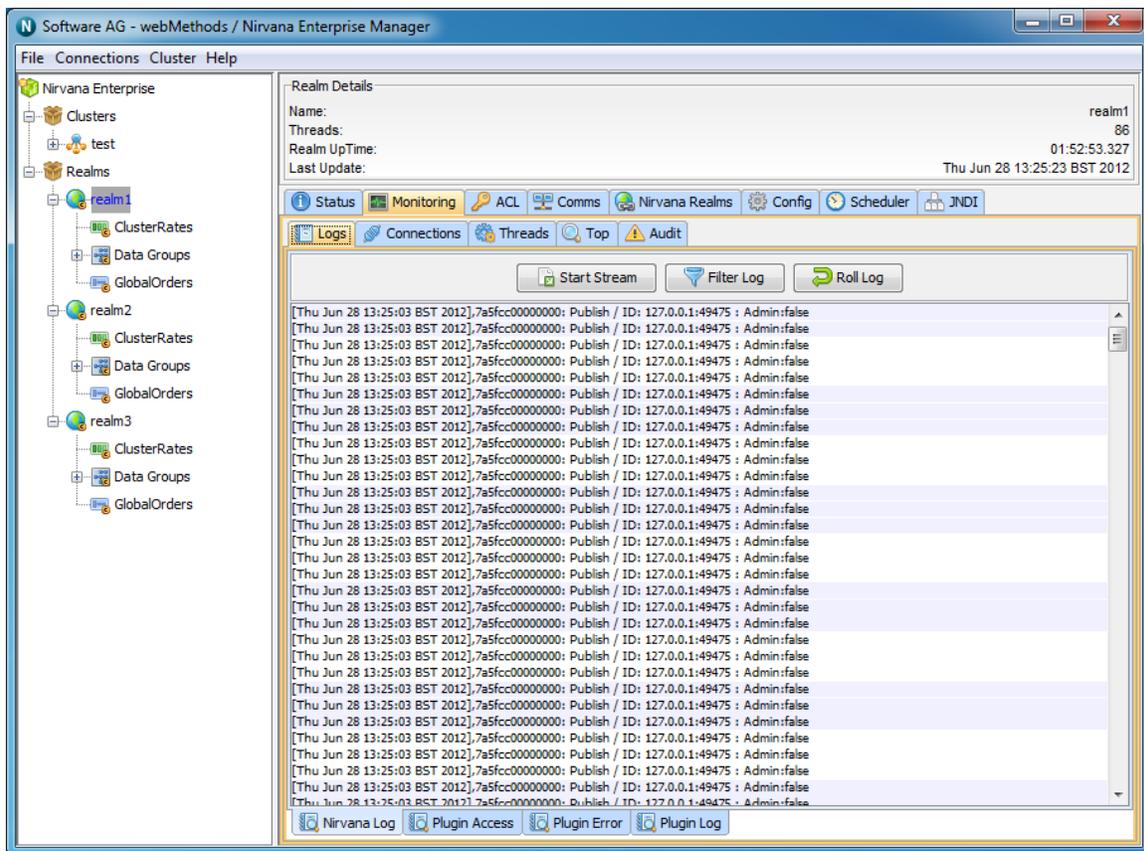
How to Administer a Remote Universal Messaging Realm

A typical development setup involves installing a Universal Messaging Realm in a remote development server as well as the developer's workstation. This guide will help you connect to the remote development Universal Messaging realm for administration purposes.

A Universal Messaging realm by default enforces a security model that allows only the username running the server process to connect to it with full privileges, and that can only be done when connecting from localhost (127.0.0.1). Therefore, in order to be able to connect remotely you need to add an ACL entry for the user and the IP address you will

connecting from. To do this you need to use the naddrealmacl command line tool on the development server as follows:

Windows OS Server Steps

1. Open a client command prompt from the Start Menu shortcut (see "Command Prompts" on page 25)

2. Type "naddrealmacl <user> <ip> full", where <user> is the OS username that the development workstation will use to connect and <ip> is the ip address of the development workstation. In this instance we specify that full access should be given to this user.

Linux / Solaris Server Steps

1. Open a console window (shell)

2. Type "cd <install dir>/client/<realm_name>/bin", where <install_dir> is your installation path and <realm_name> is the name you assigned to the realm during installation.

3. Type "export RNAME=nsp://localhost:9000", this sets an environment variable called RNAME that all samples and command line tools use in order to know how to connect to the server. In this instance we are using the Universal Messaging Socket Protocol on localhost and port 9000. If you have chosen a different port please adjust accordingly.

4. Type " ./naddrealmacl <user> <ip> full", where <user> is the OS username that the development workstation will use to connect and <ip> is the ip address of the workstation. In this instance we specify that full access should be given to this user.

Development Workstation Steps

1. Run your enterprise manager on the development workstation and click on the Connections menu, selecting Connect To Realm.

2. A dialog will pop up asking you to specify the RNAME to use. Similarly to what we did for the command line tool, we specify "nsp://<server ip>:9000", where <server ip> is the IP address where the server is running and 9000 is the port the server is listening on. Again if you have chosen a different port please adjust accordingly

3. Click ok and you should see your realm appear in the tree under the Realms folder.

Upgrading from a Trial to a Production Licence

Note: Universal Messaging ships with a trial licence, which allows the server to run for a maximum of 90 days from first run.

Trial Users

To purchase a production licence, please contact us.

Production Users

If you already have a production licence, and download a new build of the version for which you are licenced, then you should overwrite the shipped trial licence with your production licence to avoid being restricted to only 90 days' usage.

To do this, simply copy your production `licence.xml` over the trial `licence.xml` and restart your server.

The location of licence file is by default as follows:

■ If you are using the webMethods Universal Messaging distribution:

<Software_AG_directory> /UniversalMessaging/server/*<realm_server_name>*

where *<Software_AG_directory>* is the default location for the installation of webMethods products, and *<realm_server_name>* is the name of the realm server to which the licence applies.

■ If you installed the product using the Universal Messaging standalone (Terracotta) installer:

<install_directory> /server/licence/

If you encounter any problem with this process, please contact us for further support.

4   Universal Messaging Instance Manager

During the installation of webMethods Universal Messaging, you have the option of creating a default instance (called "umserver" by default) for all the components installed. If you need to create additional instances, this can be done using the ninstancemanager command line tool, which can be found under *<Software_AG_directory>* /UniversalMessaging/tools/InstanceManager/.

> **Note:**   The Instance Manager is available if you have installed Universal Messaging using the Software AG Installer; it is not available if you installed Universal Messaging using the standalone installation.

Components

The ninstancemanager tool can create instances of realm servers (RS), Enterprise Manager (EM) and template applications (TA). In order to create an instance of a component, this needs to have been installed first.

Usage Message

Executing the ninstancemanager tool without any arguments provides a usage message as follows:

```
ninstancemanager <Action> <Instance Name> <Component> <Host> <Port> [Data directory]
```

- <Action> can be either create, delete, query or deleteAll (followed directly by a component).

- <Instance Name> can be any instance name.

- <Component> is the component the action applies on, namely RS (for Realm Server), EM (for Enterprise Manager), TA (for Template Application) or ALL (for everything installed).

- <Host> is the hostname or IP that the template apps & Enterprise manager will point to, and the adapter the realm will bind to.

- <Port> is the TCP port that the template apps & Enterprise Manager will point to, and the adapter the realm will bind to.

- <Data directory> is the realm server working directory. This parameter is optional, and the default value is "UniversalMessaging/server/*<Instance Name>* ".

Example 1: To create a new instance called umserver2 and listening to all IPs on port 9001, you would run:

```
ninstancemanager create umserver2 all 0.0.0.0 9001
```

Example 2: To create a new EM instance called umserver2 and pointing to a realm on 192.168.1.100 port 9001 you would run:

```
ninstancemanager create umserver2 all 192.168.1.100 9001
```

Example 3: To delete all instances called umserver2 you would run:

```
ninstancemanager delete umserver2 all
```

Example 4: To delete an EM instance called umserver2 you would run:

```
ninstancemanager delete umserver2 em
```

Example 5: To query installed instances you would run:

```
ninstancemanager query
```

Example 6: To delete all realm server instances you would run:

```
ninstancemanager deleteAll rs
```

Querying Installed Instances

Running the ninstancemanager tool with the query action displays a list of currently installed instances. For example:

```
ninstancemanager query
```

will display an output similar to the following in a default installation (taking release 9.8.0 as an example):

```
webMethods Universal Messaging installation query
-------------------------------------------------
Realm Server Installed Version:
e2ei/11/NUM_9.8.0.0.13321/UniversalMessaging/NUMRealmServer
Instances: umserver
Enterprise Manager Installed Version:
e2ei/11/NUM_9.8.0.0.13321/UniversalMessaging/NUMEnterpriseManager
Instances: umserver
Template Applications Installed Version:
e2ei/11/NUM_9.8.0.0.13321/UniversalMessaging/NUMTemplateApplications
Instances: umserver
```

Creating Instances

Running the ninstancemanager tool with the create action allows you to create instances of all the installed components or a subset. In order to create an instance, you need to run the ninstancemanager as follows:

```
ninstancemanager create <Instance Name> <Component> <Host> <Port>
```

Where:

■ <Instance Name> is a logical name for the instance which needs to be unique for each installation.

■ <Component> is the component you wish to create an instance of. The possible values are ALL (for all components installed), RS (for a realm server instance), TA (for template applications instance) or EM (for enterprise manager instance).

Example: If we wanted to create an instance of all components installed called testinstance, bound to all IPs of the machine and listening on port 9002 you would enter:

```
ninstancemanager create testinstance all 0.0.0.0 9002
```

Output:

```
Created RS instance testinstance
```

```
Created TA instance testinstance
Created EM instance testinstance
```

You can then verify the instance's presence by issuing a query action:

```
ninstancemanager query
```

Output:

```
webMethods Universal Messaging installation query
-------------------------------------------------
Realm Server Installed Version:
e2ei/11/NUM_9.8.0.0.13321/UniversalMessaging/NUMRealmServer
Instances: testinstance , umserver
Enterprise Manager Installed Version:
e2ei/11/NUM_9.8.0.0.13321/UniversalMessaging/NUMEnterpriseManager
Instances: testinstance , umserver
Template Applications Installed Version:
e2ei/11/NUM_9.8.0.0.13321/UniversalMessaging/NUMTemplateApplications
Instances: testinstance , umserver
```

Deleting Instances

The ninstancemanager tool can be used to delete any instances created, including the default instance created using the installer. The components specified allow you to remove an instance for one component while keeping it for the others.

In order to delete an instance, you need to run the ninstancemanager as follows:

```
ninstancemanager delete <Instance Name> <Component>
```

Where:

■ <Instance Name> is a logical name for the instance which needs to be unique for each installation.

■ <Component> is the component you wish to create an instance of. The possible values are ALL (for all components installed), RS (for a realm server instance), TA (for template applications instance) or EM (for enterprise manager instance).

Example: If we wanted to delete a previously created instance of all components called testinstance, you would enter:

```
ninstancemanager delete testinstance all
```

Output:

```
RS instance testinstance has been deleted
TA instance testinstance has been deleted
EM instance testinstance has been deleted
```

You can then verify the instance's presence by issuing a query action:

```
ninstancemanager query
```

Output:

```
webMethods Universal Messaging installation query
-------------------------------------------------
Realm Server Installed Version:
e2ei/11/NUM_9.8.0.0.13321/UniversalMessaging/NUMRealmServer
Instances: umserver
Enterprise Manager Installed Version:
e2ei/11/NUM_9.8.0.0.13321/UniversalMessaging/NUMEnterpriseManager
```

```
Instances: umserver
Template Applications Installed Version:
e2ei/11/NUM_9.8.0.0.13321/UniversalMessaging/NUMTemplateApplications
Instances: umserver
```

Deleting All Instances of a Component

You can delete all instances of a component (RS, EM, TA or ALL) by using the "deleteAll" action and passing the component:

Example: If we wanted to delete all previously created instances of the component type "template application" (TA), you would enter:

```
ninstancemanager deleteAll ta
```

Output:

```
TA instance umserver has been deleted
```

You can then verify that the instance(s) have been deleted by issuing a query action:

```
ninstancemanager query
```

Output:

```
webMethods Universal Messaging installation query
-------------------------------------------------
Realm Server Installed Version:
e2ei/11/NUM_9.8.0.0.13321/UniversalMessaging/NUMRealmServer
Instances: umserver
Enterprise Manager Installed Version:
e2ei/11/NUM_9.8.0.0.13321/UniversalMessaging/NUMEnterpriseManager
Instances: umserver
Template Applications Installed Version:
e2ei/11/NUM_9.8.0.0.13321/UniversalMessaging/NUMTemplateApplications
Instances:
```