Windchill® Configuration Assistant Installation and Usage Guide

Windchill 9.0 & Windchill 9.1

Windchill PDMLink®
Windchill ProjectLink™
Pro/INTRALINK® 9.0 & Pro/INTRALINK® 9.1
Arbortext® Content Manager™

May 2010

Parametric Technology Corporation, 140 Kendrick Street, Needham, MA 02494 USA

Contents

Overview 1-1

What the Windchill Configuration Assistant Does 1-1

Audience 1-2

Windchill System Requirements 1-2

Supported Windchill Releases 1-3

Affected Windchill Property Files 1-3

Affected Apache and Tomcat Files 1-3

Installing the Windchill Configuration Assistant 2-1

Media Content 2-1

Installation Steps 2-2

Steps for Uninstalling the Assistant 2-3

Running the Windchill Configuration Assistant 3-1

General Procedure for Running the Assistant 3-1

Ant Command Syntax for Running the Assistant 3-2

Running Windchill Configuration Assistant in Unattended Mode 3-2

Running Windchill Configuration Assistant in Propagation Mode 3-3

Targets for WindchillConfigAssistant.xml 3-4

Running the Assistant in Interactive Mode 3-8

Setting Property Values on the Command Line 3-9

Troubleshooting xconfmanager Issues 3-9

Missing windchillconfigurator.xconf File 3-9

Missing ConfigurationRef Element 3-10

Setting Property Values Using the xconfmanager Utility 3-10

Windchill Configuration Assistant Log Files 3-11

Configuration Walkthrough 4-1

configureWindchill 4-1

Determining System Memory and CPUs 4-2

Calculating Memory Regions 4-3

Calculating Optimum Server Manager and Method Server Configuration 4-4

Setting Windchill Application Properties 4-5

Interactive Mode Configuration Options 4-7

configureTomcat 4-9

configureApache 4-10

Sample Ant Command Output A-1

1

Overview

This chapter provides an overview of the Windchill Configuration Assistant.

What the Windchill Configuration Assistant Does

The Windchill Configuration Assistant (WCA) examines system resource information for the server on which Windchill is installed and then calculates recommended values for Windchill properties to take maximum advantage of available memory and CPU resources. Additionally, if your Windchill environment includes Apache and Tomcat, the Windchill Configuration Assistant can recommend configuration changes for your Web server and servlet engine.

When running the Windchill Configuration Assistant, you can have the assistant do any of the following activities:

- Only calculate recommended values for Windchill properties and for Apache and Tomcat configuration options.
- Calculate and then propagate the recommended values with no user intervention.
- Calculate and display recommended values, allowing you to accept each recommended value or enter a different value. This interactive mode can be used when just calculating values or when calculating and propagating values.
- Restore the original values that were set before the assistant propagated recommended values.

To perform these activities, the assistant accesses information on your server, accesses the property files and configuration files in your installed Windchill environment.

When updating properties files that are managed by the xconfmanager utility (wt.properties, db.properties, and so on), the assistant writes its updated values to a separate xconf file that may be linked to by the site.xconf file. This allows you to keep track of Windchill property changes introduced by the assistant yet still use the xconfmanager utility to propagate changes.

The assistant automatically creates backup properties files to allow changes to be rolled back. The files in your Windchill environment are only affected if you choose to propagate the recommended values that are calculated.

When you choose to propagate recommended values to Apache or Tomcat configuration files, the assistant backs up the appropriate config.properties files and executes one or more ant commands to update settings in your Apache or Tomcat environment.

Note: On an HP-UX system, the Windchill Configuration Assistant may not be able to configure Apache. This is because the PTC Solution Installer is not used to install and configure Apache on HP-UX.

Audience

This document assumes you have the following knowledge and skills:

- An understanding of the basic Windchill architecture (web servers, servlet engines, server manager, method server, background method server, and database) and properties files.
- Use of operating system command shell utilities that allow you to edit configuration files, view log files, and so on.
- Use of the xconfmanager utility.

Windchill System Requirements

To run the Windchill Configuration Assistant successfully, the following items must be true:

- Java JDK version installed must be 5 or higher. On some platforms, Java 5 called Java 1.5. The JDK is necessary because the assistant requires the javac compiler.
- The Windchill version that is installed must be 9.0 or higher.
- Your installed Windchill system must have base data loaded. Loading base data is required for all Windchill solutions and is selected by default when Windchill is installed.

You can run the assistant regardless of the type of database you have installed. However, the assistant can only connect to Oracle databases and not to SQL Server databases. Therefore, some of the properties related to cache sizes cannot be updated by the assistant when you are running Windchill with SQL Server.

The assistant uses the total number of principals, contexts, and context teams that are set up when calculating some of the recommended values. For the assistant to calculate optimal values for a production environment, you should set up the Windchill solution that is being configured as it will be used in a production environment. For example, running the assistant against a test environment that has only a minimal number of users will not recommend the values you should use for a production system that has many users. Instead, update the Windchill solution to reflect the actual number of users you expect to be accessing Windchill on a daily basis and the number of user-defined groups, organization contexts, and application contexts that will be used in a production environment.

Supported Windchill Releases

The Windchill Configuration Assistant is supported on the following Windchill releases:

- Windchill 9.0 M080
- Windchill 9.1 M050 and later 9.1 maintenance releases

Affected Windchill Property Files

The Windchill Configuration Assistant can recommend changes to values in the following property files:

- wt.properties
- db.properties
- service.properties

The calculated values for properties in these files are then used when the assistant propagates the recommendations.

In addition to these properties files, the Windchill Configuration Assistant can update the file <*WSInstall*>/codebase/WEB-INF/methodServerMBeanConfig.xml. This file contains properties that are used by the various method server JMX MBeans.

Note: Property values set by the Windchill Configuration Assistant are not managed by the xconfmanager utility. After you propagate setting changes made by the Windchill Configuration Assistant, you must rerun the Windchill Configuration Assistant to make additional changes. The values set through the Windchill Configuration Assistant override any changes made using the xconfmanager utility.

Affected Apache and Tomcat Files

The Windchill Configuration Assistant can recommend changes to values in the following Apache and Tomcat files:

- *APACHE_HOME*/config.properties
- *TOMCAT_HOME*/config.properties

where *APACHE_HOME* is the Apache installation directory and *TOMCAT_HOME* is the Tomcat installation directory.

The recommended values for properties in these files are then used when the assistant executes ant commands against corresponding Apache and Tomcat config.xml files. The Apache config.xml target makes changes in the ajpWorkers.conf file. The Tomcat config.xml target makes changes in the wttomcat_start script, the web.xml file, and the server.xml file.

2

Installing the Windchill Configuration Assistant

This chapter describes the Windchill Configuration Assistant contents, the installation steps, and the steps used to uninstall the assistant.

Throughout this chapter, <*WCAinstall*> is used to represent the directory where the Windchill Configuration Assistant is installed.

Media Content

The Windchill Configuration Assistant is deployed as a standard ant project. It is comprised of the following files:

- WindchillConfigAssistant.xml

  The WindchillConfigAssistant.xml file is an ant xml build file. It contains the build targets that update the various properties files.

- WCA.jar

  The WCA.jar file contains the complied code for Windchill Configuration Assistant.

- Manifest.mf

  The Manifest.mf file contains the Windchill Configuration Assistant version information.

- Configurator.properties

  The Configurator.properties file is a properties file used by the Windchill Configuration Assistant. It contains various properties that define default or suggested property values.

- GCBaiter.xsl

  The GCBaiter.xsl is the XSLT Stylesheet used for Garbage Collection Baiting (GCBaiter). GCBaiting is used to stave off memory exhaustion caused by memory excessive business operations.

- WCConfigAssistInstallUsage.pdf

  The WCConfigAssistInstallUsage.pdf is this guide.

You can download a ZIP file containing the media content from the PTC Software Downloads page. Start by accessing the Technical Support page using the following URL:

http://www.ptc.com/support/index.htm

From the support page, you can navigate to the page where you can download the Windchill Configuration Assistant. For example:

1. Click **Order or Download Software Updates** and then click the **Order or Download Software Updates** link.
2. Enter your customer number and click **Continue**.
3. Navigate to the Windchill Configuration Assistant and download the ZIP file that has the content.

Installation Steps

Complete the following steps to install the Windchill Configuration Assistant:

1. Download a ZIP file containing the installation media from the PTC web site as described in Media Content.
2. Extract the contents of the ZIP file into a directory on the system where Windchill is installed.

   The location you choose for the installation does not need to be within the Windchill directory tree.
3. Start a windchill shell for the Windchill solution that you are configuring. Working within the windchill shell sets up your environment correctly. Ensure that the user account active while installing the assistant has write permission to the chosen directory and also to the site.xconf file.

   For example on a Windows system where Windchill program shortcuts have been created, navigate to the Windchill menu under **Programs** and click **Windchill Shell**.

   Otherwise, navigate to the Windchill bin directory from a command prompt and enter:

   ```
   windchill shell
   ```
4. Navigate to the directory used in step 2 containing WindchillConfigAssistant.xml.
5. Complete the installation by running the install target using the following ant command:

   ```
   ant -f WindchillConfigAssistant.xml install
   ```

When the installation completes successfully, the following message displays:

```
Windchill Configuration Assistant installed successfully
```

Note: After you have completed step 5, you can run the Windchill Configuration Assistant as described in chapter 3. However, property updates cannot be propagated using the xconfmanager until the site.xconf file has been updated with the ConfigurationRef link. If you want to experiment with the assistant without the risk of propagating changes, then you can skip the remaining installation steps for now.

When you are ready to configure site.xconf to include property updates generated by the assistant, then proceed with remaining steps.

6. Add the ConfigurationRef element to your site.xconf file by running the installConfigRef target using the following ant command:

   ```
   ant -f WindchillConfigAssistant.xml installConfigRef
   ```

   When the installation completes successfully, the following message displays:

   ```
   ConfigurationRef element successfully added to
   site.xconf
   ```

7. Verify that the ConfigurationRef element was added to your site.xconf file. Your site.xconf file is located under the Windchill directory where your Windchill solution is installed.

   Open site.xconf in a text editor and search for ConfigurationRef. For example, if you install the assistant under D:/ptc/Config, you should find the following element:

```
<ConfigurationRef
xlink:href="D:/ptc/Config/conf/windchill/windchillconfigurator.xconf"/>
```

8. Navigate to the <*WCAInstall*>/conf/windchill directory. Verify that the windchillconfigurator.xconf file does not contain any property overrides. For example, the initial contents of the file is as follows:

   ```
   <?xml version="1.0" encoding="UTF-8"?>

   <!DOCTYPE Configuration

     SYSTEM "xconf.dtd">

   <Configuration/>
   ```

Steps for Uninstalling the Assistant

To uninstall the assistant, complete the following steps:

1. Remove the ConfigurationRef element from site.xconf.

   Note: Ensure that the user account you are using has write permission on the site.xconf file.

   You can use a text editor to remove the element, or you can run the ant command for WindchillConfigAssistant.xml with the uninstallConfigRef target.

For example from a windchill shell, navigate to the Windchill Configuration Assistant installation directory and enter:

```
ant –f WindchillConfigAssistant.xml uninstallConfigRef
```

This command does not delete any of the files that were part of the Windchill Configuration Assistant installation nor any of the files created as part of running the assistant.

2. Optionally, delete the Windchill Configuration Assistant installation directory and its contents.

3

Running the Windchill Configuration Assistant

This chapter describes how to run the Windchill Configuration Assistant using an ant command from within a windchill shell.

Throughout this chapter, <*WCAinstall*> is used to represent the directory where the Windchill Configuration Assistant is installed and <*WSinstall*> is used to represent the directory where your Windchill solution is installed.

Note: Before running the assistant, you should stop both Windchill and Tomcat (or your servlet engine, if you are not using Tomcat). Do this so that the assistant can accurately measure the amount of free RAM on the system. Additionally, stop any other programs that would not normally be running when Windchill is in use**.**

General Procedure for Running the Assistant

Use the following steps to set up the required Windchill environment and run the Windchill Configuration Assistant:

1. Start a windchill shell for the Windchill system that you are configuring. Working within the windchill shell sets up your environment correctly.

   For example on a Windows system where Windchill program shortcuts have been created, navigate to the Windchill menu under **Programs** and click **Windchill Shell**.

   Otherwise, from a command prompt, navigate to the Windchill bin directory and enter:

   ```
   windchill shell
   ```

2. Navigate to the directory containing WindchillConfigAssistant.xml. This file is located in the <*WCAinstall*> directory.

3. Enter an ant command to execute the required target defined in WindchillConfigAssistant.xml file, as described in the next section, and respond to any prompts that display.

All activity is reported to the command window and a record of the activity is stored in the log file that is generated. When the assistant successfully completes its activities, it displays messages similar to the following:

```
******************************************************************

!+! FINISHED @ Tue Apr 27 14:50:48 CST 2010

!+! Log: <WSinstall>\Windchill\buildlogs\0001-WindchillConfigAssistant.log

******************************************************************

 BUILD SUCCESSFUL

 Total time: 2 minutes 10 seconds
```

For log file details, see Windchill Configuration Assistant Log Files.

Ant Command Syntax for Running the Assistant

To run the Windchill Configuration Assistant, enter an ant command in one of the following general formats:

```
ant -f WindchillConfigAssistant.xml -projecthelp
```

Using this format displays a description of the main targets that can be specified.

```
ant -f WindchillConfigAssistant.xml target
[-Dinteractive=false] [-Dpropagate=true]
[-D<Property>=<Value>]
```

Using this format executes the specified target and optionally provides arguments that you can use to turn off the interactive mode, propagate configuration changes, and override values for specific properties.

You can specify one or more –D<*Property*>=<*Value*> arguments to set one or more property values that are then used in place of the default values used by the assistant. The set of property value defaults are stored in Configurator.properties file.

Running Windchill Configuration Assistant in Unattended Mode

The Windchill Configuration Assistant can be run with or without user interaction. By default the assistant runs in interactive mode. In interactive mode, the user is prompted to accept or override property values calculated by the assistant as it runs each internal target.

To run the assistant in unattended mode, include -Dinteractive=false on the ant command:

```
ant -f WindchillConfigAssistant.xml target
-Dinteractive=false
```

In unattended mode the assistant uses the calculated property values as though they were accepted by the user.

The *targets* for which the -Dinteractive=false argument is valid are as follows:

configure
configureWindchill
configureApache
configureTomcat

If you specify other targets on the ant command, the assistant does not check the value of the interactive property when it runs. Therefore, the -Dinteractive=false argument is not used.

Running Windchill Configuration Assistant in Propagation Mode

When the Windchill Configuration Assistant runs a target that calculates recommended configuration values, it stores updated property name/value pairs in the file named <*WCAinstall*>/conf/windchill/windchillconfigurator.xconf. The properties in this XCONF file can be propagated by the xconfmanager utility through to the various property files that it manages if the ConfigurationRef element has been added to site.xconf as described in the Installation Steps.

The Windchill Configuration Assistant has the option to call xconfmanager utility to propagate all calculated property values to the appropriate properties files. By default, the Windchill Configuration Assistant does not propagate values; therefore, unless you explicitly set the value of the propagate property to true on the ant command line, the calculated property values are not propagated.

To run the assistant in propagation mode, include –Dpropagate=true on the ant command:

```
ant -f WindchillConfigAssistant.xml target
-Dpropagate=true
```

The *targets* for which the –Dpropagate=true argument is valid are as follows:

configure
configureWindchill
configureApache
configureTomcat

If you specify other targets on the ant command, the assistant does not check the value of the propagate property when it runs. Therefore, the –Dpropagate=true argument is not used.

Note: You can include both the –Dinteractive=false and –Dpropagate=true arguments if you want to run in both unattended and propagation modes at the same time.

Note: If the Windchill Configuration Assistant is executed in propagation mode, it automatically adds the ConfigurationRef element to the site.xconf file. The addition happens regardless of whether the installConfigRef target has been run.

Targets for WindchillConfigAssistant.xml

The targets for WindchillConfigAssistant.xml can be grouped as follows:

- Targets that calculate recommended configuration values, write them to intermediate properties files and propagate them if –Dpropagate=true is specified.
- Targets that propagate changes already stored in intermediate properties files.
- Targets that you can use to restore your system to a previous configuration.
- Targets for adding and removing the ConfigurationRef element to the site.xconf file. These targets are described the Installation Steps and Steps for Uninstalling the Assistant sections.

The following sections describe the targets that are supported for configuration options.

Targets for Calculating Recommended Values

The following table describes the targets that you can use to calculate recommended configuration values and write them to intermediate properties files.

Note: If the –Dpropagate=true command line argument is set, then the new values are propagated using the xconfmanager utility.

If the –Dinteractive=false command line argument is set, you are not prompted to confirm the recommended values.

Note: After executing the configureWindchill target and propagating the changes, you must manage any additional changes to the properties set by the target through the Windchill Configuration Assistant. Any property changes made using the xconfmanager utility are not used because they are overwritten by values set in <*WCAinstall*>/conf/windchill/windchillconfigurator.xconf.

Target	**Description**
configure	Calls the following targets: configureWindchill configureTomcat configureApache Using the configure target has the same results as running the Windchill Configuration Assistant three times, each time with one of the listed targets. If you are not using Tomcat, the assistant exits after it executes the configureWindchillTarget

Target	Description
configureWindchill	Calculates new values for the following properties and writes them to <*WCAinstall*>/conf/windchill/windchillconfigurator.xconf file wt.manager.monitor.start.MethodServer wt.method.loadbalance.maxRedirects wt.manager.cmd.MethodServer.gc.log.args wt.manager.cmd.MethodServer.java.extra.args wt.manager.cmd.MethodServer.nonibm.java.args wt.manager.cmd.ServerManager.debug.args wt.manager.cmd.ServerManager.gc.log.args wt.manager.cmd.ServerManager.ibm.java.args wt.manager.cmd.ServerManager.java.exra.args wt.manager.cmd.ServerManager.nonibm.java.args wt.manager.cmd.ServerManager.platform.java.args wt.manager.maxHeap wt.manager.minHeapwt.manager.monitor.services wt.manager.monitor.start.BackgroundMethodServer wt.queue.executeQueues wt.method.minHeap wt.method.maxHeap wt.method.bg.maxHeap wt.method.gb.minHeap wt.method.gcInterval wt.manager.gcInterval wt.method.log.tee wt.manager.log.tee wt.manager.cmd.MethodServer.nonibm.java.args wt.manager.cmd.common.java.extra.args wt.pom.maxDbConnections wt.pom.minDbConnections wt.pom.dbConnectionsHardLimit wt.pom.refreshCache.size wt.pom.rowPrefetchCount wt.pom.statementCacheSize wt.pom.paging.snapshotQueryLimit wt.pom.inClauseBindOptimizationCardinality wt.pom.inClauseUseBindOptimization wt.pom.oracle.disableAlwaysSemiJoin wt.pom.queryLimit com.ptc.windchill.search.queryLimit wt.admin.cache.maxDomains wt.cache.size.AclCache wt.cache.size.FederatableServerHelper$RemoteObjectIdCache wt.cache.size.PagingSessionCache wt.cache.size.PreferenceCache wt.cache.size.RoleAccessCache wt.cache.size.SessionCache wt.cache.size.StandardUfidSrvService$RemoteObjectIdCache wt.cache.size.WTPrincipalCache wt.services/rsc/default/ObjectReferenceCacheTable/ContainerCache.Size/null/0 wt.services/rsc/default/ObjectReferenceCacheTable/ContainerTeamCache.Size/null/0

Target	Description
configureApache	Determines the current values for the following properties that are stored in the *APACHE_HOME/*config.properties file and generates an intermediate file containing recommended updates in <*WCAinstall*>/conf/apache/config.properties: ajpSmaxConn ajpMaxConn ajpMinConn Current values are saved in <*WCAinstall*>/conf/apache/configproperties.backup.
configureTomcat	Determines the current values for the following properties that are stored in the *TOMCAT_HOME*/config.properties file and generates an intermediate file containing recommended updates in <*WCAinstall*>/conf/tomcat/config.properties: ajpMaxThreads extraJavaArgs maxHeap minHeap Current values are saved in <*WCAinstall*>/conf/tomcat/configproperties.backup:

Targets for Propagating Values from Intermediate Properties Files

The following table lists the targets that can be selected to propagate property values from intermediate properties files using the xconfmanager utility.

Note: To use the xconfmanager utility, the ConfigurationRef element must be in your site.xconf file. Steps to add this element are documented in the Installation Steps section.

Before running the propagate targets, the assistant checks to ensure that the associated intermediate files exist. If a required intermediate file does not exist, the assistant returns an error stating that fact before attempting to propagate related property values. To correct the problem, you can rerun the corresponding target that produces the intermediate file.

Tomcat and Apache properties are not managed by the xconfmanager utility. Following the initial installation, both the Apache and Tomcat installation directories contain a properties file named config.properties and an ant build script called config.xml. The config.xml scripts contain targets that regenerate Apache and Tomcat configuration files. These files are regenerated from a template file using values from the associated config.properties. Because these files are regenerated when the config.xml script is run, any manual edits (if any) that had been applied to the following files are lost:

TOMCAT_HOME/bin/wttomcat_start

TOMCAT_HOME/bin/wttomcat_start.bat

TOMCAT_HOME/conf/server.xml

APACHE_HOME/conf/extra/ajpWorkers.conf

In an effort to protect manual edits in the above files, the Windchill Configuration Assistant checks the consistency of properties in config.properties and associated Apache and Tomcat generated configuration file. This check is performed prior to running both the propagateApache and propagateTomcat targets. For Tomcat, the Windchill Configuration Assistant compares values for properties jvmRoute and ajp13Port in server.xml with the corresponding values in config.properties. If a mismatch is found, the assistant does not continue with the propagation. Instead, the Windchill Configuration Assistant issues a warning message requesting that config.properties be manually updated.

The assistant performs similar checks on APACHE_HOME/config.properties and APACHE_HOME/conf/extra/ajpWorkers.conf files prior to propagating changes. In this case, the tool checks ajpWorkers.conf for multiple BalancerMember elements. If multiple BalancerMember elements are found in ajpWorkers.conf, it does not continue with the propagation.

Note: These targets propagate property changes regardless of the value of propagate property*.*

Target	**Description**
propagate	Calls the following targets: propagateWindchill propagateApache propagateTomcat Using the propagate target has the same results as running the Windchill Configuration Assistant three times, each time with one of the listed targets. If you are not using Apache or Tomcat, the assistant only executes the propagateWindchill target.
propagateWindchill	Calls xconfmanager –p –F to propagate values from <*WCAinstall*>/conf/windchill/windchillconfigurator.xconf to windchill properties files.
propagateApache	Propagates property values found in the intermediate property file named <*WCAinstall*>/conf/apache/config.properties to the target Apache config.properties file and executes the configureAJPWorkers ant target in APACHE_HOME/config.xml.
propagateTomcat	Propagates property values found in the intermediate property file named <*WCAinstall*>/conf/tomcat/config.properties to the target Tomcat config.properties file and executes the configureConnectors and configureScripts ant targets in TOMCAT_HOME/config.xml.

Targets for Rolling Back Configuration Changes Made Through the Assistant

The following table describes the targets that you can use to roll back the configuration changes that were propagated by the assistant.

Note: Each time a target that calculates recommended values is run, the assistant generates a separate backup file. The assistant uses the latest backup files that were created for the restoration. Before running a target to roll back configuration changes, ensure that the latest backup files have the values you want to use.

Target	**Description**
restoreWindchill	Rolls back any changes that were made when the configureWindchill target was last run. It replaces values in <*WCAinstall*>/conf/windchill/windchillconfigurator.xconf with backup values.
restoreApache	Rolls back any changes that were made when the configureApache target was run using the property values stored in the backup file.
restoreTomcat	Rolls back any changes that were made when the configureTomcat target was run using the property values stored in the backup file.
restore	Calls the following targets: restoreWindchill restoreApache restoreTomcat

Running the Assistant in Interactive Mode

When you are working in interactive mode, each recommended value is displayed in an [input] line on the command line. For example:

```
[input] Number of MethodServers to configure (current=1): [2]
```

If there is a current value defined, it is included on the line in the parenthetical phrase (current=*x*) where *x* is the current value. In the example, the current value is **1**.

The recommended value is the value supplied at the end of the line inside the brackets []. In the example, the recommended value is **2**.

Some inputs require specific values. When this is the case, the acceptable values are presented in parentheses at the end of the input line and the recommended value is inside brackets []. For example, the following input line shows the **free** and **total** values as the values you can enter and the recommended value is **free**:

```
[input] Allocate Windchill Memory region from total (16378 MB)
or free (10986 MB) physical memory ([free], total)
```

To override the recommended value, enter a new value at the prompt and press return. To accept the recommended value, just press return.

Setting Property Values on the Command Line

Include the –D<*Property*>=<*Value*> argument on the command line to override a property value.

The assistant defines the following properties so that you can override calculated values for the CPU count, physical memory (in mega bytes), free memory (in mega bytes), and vendor that are calculated when running the configureWindchill target:

CPUCount
physMem
freeMem
vmVendor

For a specified property value to be used, the named property must be one that is calculated through the specified target. For example, executing the following command on a system with 4 CPUs and 8GB of RAM overrides the calculated values for CPU, physical memory, free memory, and vendor. The command specifies 16 CPUs, 36 GB of RAM, 32 GB of free memory, and IBM Corporation as the vendor:

```
ant -f WindchillConfigAssistant.xml configureWindchill
-DCPUCount=16 -DphysMem=36000
–DfreeMem=32000 –DvmVendor="IBM Corporation"
```

Setting the CPUCount, physMem, and freeMem property values on a command that is run on a desktop class system allows the assistant to calculate other values that can then be used on a server class target system.

For a list of additional properties affected by each target, see Targets for WindchillConfigAssistant.xml.

Troubleshooting xconfmanager Issues

The following sections describe common issues related to the Windchill Configuration Assistant and the xconfmanager utility that can occur.

Missing windchillconfigurator.xconf File

If you deleted the Windchill Configuration Assistant installation directory or deleted just the windchillconfigurator.xconf file from this directory, but did not remove the ConfigurationRef element from the site.xconf file, propagating changes by running the xconfmanager utility from the command line causes an error. The ConfigurationRef element names a nonexistent windchillconfigurator.xconf file. In this case, the xconfmanager utility fails with an error similar to the following:

```
ERROR: Unexpected error (stack trace below):

com.ptc.windchill.structconf.StructConfManagerException: Error scanning xconf files
to determine property-to-xconf dependencies caused by: Error parsing
file:<WSinstall>/Windchill/site.xconf, line 357 (starting at root doc file:
<WSinstall>/Windchill/site.xconf) caused by: Unable to generate URL for
href=<WCAinstall>/Tuner/conf/windchill/
windchillconfigurator.xconf'.
```

To recover from this error, you can either remove the ConfigurationRef element from site.xconf or recreate the windchillconfigurator.xconf file:

- To remove the ConfigurationRef element you can use a text editor to edit the site.xconf file or run the uninstallRef target as described in the Steps for Uninstalling the Assistant section.
- To recreate the windchillconfigurator.xconf file you can either use the installConfigRef target as part of installing the assistant to create an empty windchillconfigurator.xconf file as described in the Installation Steps section or run the configureWindchill target.

Missing ConfigurationRef Element

If you attempt to propagate the changes suggested by the Windchill Configuration Assistant, but have not added the ConfigurationRef element to the site.xconf file, the build fails at the point where the ant script attempts to locate the element.

Note: Any intermediate properties files updated from all successful internal target runs that were completed prior to the propagation attempt remain. For example, assume that the propagation is attempted when running the configureWindchill target. Then the intermediate properties files generated before the build failure are not deleted.

An error similar to the following identifies the failure:

```
ERROR: site.xconf has no ConfigurationRef to
<WCAinstall>/conf/windchill/windchillconfigurator.xconf
```

To recover from this failure, you can add the ConfigurationRef element to site.xconf as described in the Installation Steps section and then run a target that propagates values as described in Targets for Propagating Values from Intermediate Properties Files.

Setting Property Values Using the xconfmanager Utility

If you attempt to change the value of a property using the xconfmanager utility and that property has a value set in the windchillconfigurator.xconf file, the change you make is not used by Windchill even though the utility sets the property value in site.xconf.

The windchillconfigurator.xconf file is maintained by the Windchill Configuration Assistant and is referenced at the end of the site.xconf file. Therefore, the values set in the windchillconfigurator.xconf file override any values set using the xconfmanager utility.

For a list of affected properties, see Targets for Calculating Recommended Values.

Windchill Configuration Assistant Log Files

Each time you run the Windchill Configuration Assistant, it creates a log file that captures all activity.

Log files are stored in the <*WSinstall*>/Windchill/buildlogs directory and each log file name has the following format:

<#>-WindchillConfigAssistant.log.

where <#> is the number of the log file. Log file numbering starts with 0000 and increments by one for each log file created.

The log file content is the same as the output displayed in the command window for each time you run the Windchill Configuration Assistant.

If the Windchill Configuration Assistant cannot locate the <*WSinstall*>/Windchill/buildlogs directory, then it creates the log file in the same directory as the WindchillConfigAssistant.xml file.

4

Configuration Walkthrough

This chapter describes the actions taken by the Windchill Configuration Assistant when it runs the following targets:

configureWindchill
configureApache
configureTomcat

It describes the rules used and properties affected when configuring multiple method servers and background method servers. It also describes the properties used for configuring the Tomcat servlet engine and the Apache Web server.

configureWindchill

The running of the configureWindchill target can be broken down into the following steps:

1. Determining system memory and CPUs
2. Calculating memory regions
3. Calculating the optimum server manager and method server configuration
4. Setting additional server manager and method server options
5. Setting Windchill application properties
6. Configuring remote access from JMX clients and Garbage Collection Baiting, when running in interactive mode.

Within each of these steps, the assistant executes internal targets to complete specific tasks. The names of these targets are displayed in the output as the assistant executes them.

The sample output shown in the following sections is not intended to provide accurate data about any given system configuration, but just shows an example that is relevant to the discussion.

Determining System Memory and CPUs

The first step performed by the Windchill Configuration Assistant when running the configureWindchill target is to determine operating system characteristics, total memory, free memory, and how many CPUs are available in the system.

During the initialization of the assistant, it reports server metrics similar to the following using the init target:

```
[echo] Number of CPUs detected: 4

[echo] Physical Memory: 7744 MB

[echo] Free Memory: 7744 MB

[echo] Maximum Java heap size=2400 MB

[echo] WT_HOME=/apps/ptc/Windchill

[echo] VmVendor=IBM Corporation

[echo] Operating System=AIX

[echo] OS Arch=ppc
```

Before allocating memory to Windchill, the assistant determines how much memory is available. The server metrics (shown previously) identifies how much total memory is installed on the system and how much memory is free.

To avoid over committing memory to Windchill, the assistant reserves 20% of physical memory for the operating system. The percent allocated to the operating system is governed by property percentPhysicalMemoryReservedForOS. The assistant also discovers the location of the RDBMS and LDAP servers. If either server is configured to run on the same host as Windchill, then the assistant further reduces the amount of free memory available to Windchill.

To assist in determining in the memory allocation, the assistant calls the allocatePhysmem target. The prompts provided when you are running in interactive mode with this target allow you to:

- Choose whether to size the overall Windchill memory footprint based on total physical memory or free memory.
- Specify the percentage of memory to allocate.

By default, the assistant uses free memory and 80% in its calculation. Therefore, you should stop Windchill before running this step. You should also stop any other applications running on the server that are not expected to be running alongside Windchill.

The percentMemoryRegionReservedForWindchill property governs the percentage of available memory to use for Windchill. The default value is 80. You can change the default value by editing the value in the <*WCAinstall*>/Configurator.properties file.

In the following example, 5940 MB of memory is reported as free. This value is calculated using 7740MB (physical memory) and subtracting both 1548MB (operating system memory) and 256MB (LDAP). Based on the 80% default value, the assistant allocates 4752 MB to Windchill:

```
allocatePhysMem:

    [input] Percent of physical memory to reserve for Operating System [20]

     [echo] OS memory requirement=1548 MB

     [echo] LDAP memory requirement: 256 MB

     [echo] Minimum memory requirement for MethodServer: 192 MB

     [echo] Minimum memory requirement for ServerManager: 16 MB

     [echo] Minimum memory requirement for ServletEngine: 128 MB

     [echo] Windchill Requires a minimum of 336 MB

     [echo] Total Memory requirement: 2140.0 MB

     [echo] Adjusting free memory based on 3rd party application memory
requirements: 7744->5940 MB

    [input] Allocate Windchill Memory region from total (7744 MB) or free (5940 MB)
physical memory ([free], total)

[echo] Allocating Windchill Memory region from free memory

    [input] Enter % of free RAM (5940 MB) to allocate to Windchill: [80]

[echo] % free memory allocated to Windchill: 80

     [echo] Windchill Memory Region= 4752 MB
```

Calculating Memory Regions

In this next step, the assistant partitions the allocated memory into three regions. These regions are for the method servers, server manager, and Tomcat. The assistant uses percentages to calculate the size of the three memory regions. The default allocations are as follows:

- Method servers get 75%
- Tomcat gets 20%
- Server manager gets 5%

When you are running in interactive mode, you can change the allocation percentages. The following example accepts the defaults and shows the resulting memory allocations:

```
[input] Enter % of Windchill Memory (4752) to allocate to MethodServer Heap: [75]

     [echo] % Windchill memory to allocate to MethodServers: 75

    [input] Enter % of Windchill Memory (4752) to allocate to ServerManager Heap:
[5]

     [echo] % Windchill memory to allocate to ServerManager: 5

    [input] Enter % of Windchill Memory (4752) to allocate to ServletEngine Heap:
[20]
```

```
     [echo] % Windchill memory to allocate to ServletEngine: 20
     [echo] MethodServer memory region..... 3564 MB
     [echo] ServerManager memory region..... 237 MB
     [echo] ServletEngine memory region..... 950 MB
```

Calculating Optimum Server Manager and Method Server Configuration

After the memory region sizes have been calculated, the assistant determines the number of background method servers and method servers to configure. It does this by considering both the size of the method server memory region and the number of CPUs.

The configureMethodServerService target calculates the number of method servers and background method servers based on the number of CPUs reported by the operating system. It then uses a lookup table containing the ratio of CPU to method servers that is defined in Configurator.properties to determine the ideal number of method servers for the number of CPUs. A second lookup table is used to determine the optimum number of method servers based on operating system and method server memory region size. When you are running in interactive mode and the assistant finds sufficient resources to run multiple method servers, it may prompt you to configure a background method server. If you do not want a background method server configured, enter **n** when prompted. If the assistant determines that the current Windchill installation belongs to a cluster, then it will only offer to configure a background method server on the host configured as the cluster master.

In the following example, the assistant recommends 2 method servers for the 4-CPU server and offers to configure a background method server:

```
configureMethodServerService:
     [echo] Recommended number of MethodServers for 4 CPUs = 2
     [echo] Recommended number of MethodServers for 3564MB on AIX (32bit) is 2
     [echo] Recommended number of MethodServers based on available resources=2
    [input] Enter number of MethodServers to configure (current=1): [2]
    [input] Enter MethodServer max heap size (MB): [1782]
     [echo] Setting max heap size for MethodServer to 1782
    [input] Enter MethodServer initial heap size (MB): [594]
     [echo] Setting initial heap size for MethodServer to 768
    [input] Do you want to configure a BackgroundMethodServer (y/n)? ([y], n)
     [echo] Configuring Background MethodServer
```

When the assistant configures a background method server it appoints one of the foreground method servers as the background method server and reduces the number of foreground method servers by one. Memory for method servers (including the background method server - if any) is allotted from the method server memory region.

All method servers (including background method server) are assigned identical java heap sizes. The heap size is calculated by simply dividing the method server heap region by the sum of background plus foreground method servers. The calculated heap size must not exceed the heap size limit defined for the operating system and memory model (ie 32bit or 64 bit). In the example above, both the background and foreground method servers were assigned heaps of 1782MB (ie 3564MB divided by two).

After the number of method servers and heap sizes has been set, the assistant configures additional Java command line options. These command line options fine tune the heap generation sizes for maximum garbage collection throughput.

The assistant configures the JVM initial and maximum permanent generation size, new generation size and survivor space size. The permanent generation sizes are governed by properties in Configurator.properties. The new generation size is calculated as a percentage of the maximum heap size where, again, the default percentages are defined in Configurator.properties.

If multiple foreground method servers are being configured, the assistant configures appropriate load balancing properties as shown in the following example:

```
[input] Enter active MethodContext loadbalance threshold: [6]

[input] Enter RMI call redirect count limit: [1]
```

These properties govern when to redirect client requests (based on the number of active contexts in a method server) and limit the number of times a request can be redirected between method servers. This redirect limit is calculated by subtracting 1 from the number of foreground method servers.

Setting Windchill Application Properties

After the JVM heap and garbage collector configurations have been defined, the assistant sets Windchill application properties. The first set of properties it considers are those that size in-memory data caches.

Note: The initial release of the Windchill Configuration Assistant can only query data from an Oracle database; tables in an SQL Server database cannot be queried. As a result the cache sizing targets only run where the Windchill database is served by Oracle.

Optimal cache tuning is difficult unless the system is under load. The Windchill Configuration Assistant simply suggests initial cache sizes based on the number of rows it finds in specific tables in the database.

The following Windchill cache properties are configured:

wt.services/rsc/default/ObjectReferenceCacheTable/ContainerCache.Size
wt.services/src/default/ObjectReferenceCacheTable/ContainerTeamCache
wt.cache.size.WTPrincipalCache
wt.cache.size.PolicyAcl
wt.cache.size.StandardUfidSrvService$RemoteObjectIdCache
wt.cache.size.StandardFederatedServerHelper$RemoteObjectIdCache
wt.cache.size.SessionCache
wt.cache.size.RoleAccessCache
wt.cache.size.PreferenceCache
wt.cache.size.PagingSessionCache
wt.admin.cache.maxDomains

Each cache comprises multiple entries, where each entry effectively represents one row in the corresponding database table. Windchill caches should be sized to maintain only recently accessed data rows (also known as the “working set”).

The assistant uses two basic algorithms to calculate cache sizes. The first cache sizing algorithm obtains the total number of rows in a database table and sizes the cache based on a configurable percentage of the table row count total.

The configurable percentages are defined using the following properties in configurator.properties:

WTContainer.cache.percent
ContainerTeam.cache.percent
WTPrincipal.cache.percent
PolicyAcl.cache.percent
RemoteObjectInfo.cache.percent
AdministrativeDomain.cache.percent
Session.cache.percent
RoleAccess.cache.percent
Preference.cache.percent
PagingSession.cache.percent

The second cache sizing algorithm sizes caches based on the number of users defined in the system and the percentage of those users that are expected to be concurrently executing Windchill requests during peak workloads. The predicted percentage of all users that will be concurrently active is governed by property peakActiveUsersPercent and the default value is 10. For more information on cache sizing rules refer to the spreadsheet.

Regardless of which algorithm is executed, the resulting cache size is the larger of the out-of-the-box size and the calculated cache size. A configurable size limit is applied to each cache in case the sizing algorithm over allocates memory. The maximum size for all of the above caches is governed by property <*CacheName*>.cache.size.limit and is defined in Configurator.properties.

After sizing the data caches, the assistant then considers various properties that configure database connections, JDBC statement cache sizes, and so on. These properties all reside in <*WSinstall*>/Windchill/db/db.properties. Many of these properties have initial sizes that are suited for environments with minimal resources such as laptop or small desktop systems. When configuring Windchill on servers with larger heaps these properties should be adjusted accordingly.

The assistant increases the database property values based on the method server heap sizes. The table below shows how the property values are affected by different heap sizes.

Property Name	<1000 MB	1000 - 2000 MB	>2000 MB	9.0 / 9.1 Default
wt.pom.maxDbConnections	5	10	15	5
wt.pom.minDbConnections	5	10	15	5
wt.pom.refreshCache.size	100	200	300	100
wt.pom.rowPrefetchCount	20	30	40	20
wt.pom.statementCacheSize	50	60	70	50

Other database related properties that are considered by the assistant are not constrained by memory or CPU resources. These properties are shown in the following table

Property Name	Configured Value	9.0 / 9.1 Default
wt.pom.paging.snapshotQueryLimit	50000	-1
wt.pom.inClauseBindOptimizationCardinality	-1	-1
wt.pom.inClauseUseBindOptimization	true	true
wt.pom.oracle.disableAlwaysSemiJoin	false	true

Interactive Mode Configuration Options

When you run the configureWindchill target in interactive mode, you have addition configuration options as described in the following sections.

Configuring Remote Access from JMX Clients

The Windchill Configuration Assistant prompts you to configure the server manager to allow remote access from JMX clients, such as JConsole. Such clients can be used to interact with the Windchill MBeans to manage and monitor the application. However, to connect using a JMX client from a remote workstation, you must configure additional Windchill properties.

Two modes of access are supported:

- Authenticated

  When configuring authenticated remote access, the assistant creates two users named monitorRole and controlRole and assigns each a password. Both of these users must have an entry in the JDK jmxremote.access file found at <*JRE*>/lib/management/jmxremote.access:

  – The monitorRole provides read-only access.

  – The controlRole is granted read-write access.

  Remote users must supply one of the two role names and the associated password to gain access to Windchill MBeans. The user names and passwords for the two roles are written to a file named <WSInstall>/Windchill/jmxremote.password and that file is referenced by properties on the server manager java command line.

- Anonymous

  When anonymous remote access is enabled, any remote client can connect without supplying a user name or password.

  Note: Anonymous users are granted unrestricted access.

The following example shows the prompts that are displayed to set up authenticated remote access:

```
[input] Do you want to enable authenticated (non SSL) remote JMX access to the
ServerManager(y/n)? (y, [n])

y

     [echo] Configuring authenticated remote JMX access

    [input] servermanager monitor role name: [monitorRole]

    [input] servermanager monitor role password: [ardenhill5]

    [input] servermanager control role name: [controlRole]

    [input] servermanager control role password: [n33dham]
```

Configuring Garbage Collection Baiting (GCBaiter)

To reduce the likelihood of method server heap exhaustion due to an unbounded memory operation, the assistant configures a feature known as Garbage Collection Baiting (GCBaiter). When the method server heap is 100% utilized, the JVM will generally crash with an OutOfMemoryError. The GCBaiter attempts to identify the user operation that is consuming most of the available heap prior to the OutOfMemoryError and then terminates it. The GCBaiter is initiated automatically by the JVM when it detects a low memory condition (for example, when the JVM Garbage Collector needs to collect SoftReferences). The GCBaiter uses the MethodServer.gcBaiting.suggestedMinAvailableHeapPercent property to detect when the method server available heap space has fallen below the minimum requirement.

The Windchill Configuration Assistant calculates the minimum available heap value used by GCBaiter as a percentage of the maximum method server heap size. The default value is 5% and is governed by property MethodServer.gcBaiting.suggestedMinAvailableHeapPercent. Note that the assistant does not set the DisableExplicitGC Java command line option when GCBaiting is enabled.

The following example shows the prompts that are displayed to set up Garbage Collection Baiting target configureGCBaiter:

```
configureGCBaiter:

    [input] Do you want to enable automatic termination of requests to avoid
OutOfMemory (y/n)? ([y], n)

    [input] Minimum percentage of heap available: [5]

     [xslt] Processing /opt/ptc/windchill91/codebase/WEB-
INF/methodServerMBeanConfig.xml to
/opt/ptc/modules/Tuner/conf/windchill/methodServerMBeanConfig.xml

     [xslt] Loading stylesheet /opt/ptc/modules/Tuner/GCBaiter.xsl

     [xslt] Replace minAvailableHeapThreshold: 0 with: 77594624
```

configureTomcat

Running the configureTomcat target adjusts four Tomcat properties. These properties configure the Tomcat heap size, new generation size, and survivor spaces, and ensure that the Tomcat JSP compilation mode is set to **prod** (which minimizes JSP compilation frequency).

Property Name	**Configured Value**	**9.0 / 9.1 Default Value**
maxHeap	Calculated based on memory region size	128
minHeap	Calculated based on memory region size	64
extraJavaArgs	-XX\:+HandlePromotionFailure -XmnYYYm -XX\:SurvivorRatio\=8 -XX\:+DisableExplicitGC	
mode	prod	prod

When the configureTomcat target is run in interactive mode, the assistant prompts to configure remote JMX access to Tomcat. The process for configuring Tomcat for remote JMX access is similar to the process used for the server manager (described previously). The only significant difference is that the jmxremote.password file is written to TOMCAT_HOME. For example, entries similar to the following are added to the jmxremote.password file:

```
monitorRole ardenhill5

controlRole n33dham
```

Note: When Tomcat is run as a service on Windows, the assistant does not prompt to configure remote JMX access. This is because the user account under which the assistant is run is not the same account that runs the Tomcat service (for example, the Local System account). Java imposes a restriction on the jmxremote.password file that requires it to be readable only by the user ID of the running Java process.

configureApache

Running the configureApache target adjusts properties that configure the proxy balancer BalancerMember element. This element is defined in file apache/conf/extras/ajpworkers.conf.

The max property is used to limit the number of simultaneous requests that Apache will forward to Windchill. After this limit is reached, additional requests are queued until a slot becomes available. This property effectively throttles request concurrency in Windchill and, therefore, can be used to avoid server saturation. To calculate the value for the max property, the assistant multiplies the number of foreground method servers by the value in the apache.activeContextsPerMethodServer property. The apache.activeContextsPerMethodServer default value is 15.

The configureApache target also updates BalancerMember attributes smax and min to configure the minimum and soft limit for connections. The assistant sets the both smax and min attributes to one half the value of max.

A

Sample Ant Command Output

The appendix has sample Windchill Configuration Assistant output that has been formatted to fit the page.

The appendix shows the output from the following command that was entered in a Windows environment that includes an Oracle database.

```
ant –f WindchillConfigAssistant.xml configureWindchill
-Dinteractive=false
```

For information about entering this command and the ant syntax used in the command, see Ant Command Syntax for Running the Assistant.

Using the configureWindchill target without the propagation option updates the intermediate property files with the recommended values from this run. If the files do not exist, the assistant creates them. For details on the configureWindchill target, see the configureWindchill target walkthrough.

In the output, <*WSinstall*> designates the directory where your Windchill solution is installed (such as D;\ptc\Windchill_9.1), <*WCAinstall*> is the directory where the Windchill Configuration Assistant is installed, and <*snip*> is used in place of the classpath. The hostname is shown as servername.com. By default, the command runs in interactive mode. The example shown here shows the non-interactive mode:

```
Buildfile: WindchillConfigAssistant.xml
********************************************************************************
!+! STARTED @ Tue Apr 27 14:50:35 CST 2010
********************************************************************************
Didn't locate Windchill at or above build file's directory.  Searching environment
and System.properties...
Using environment's WT_HOME variable
Log file is '<WSinstall>\Windchill\buildlogs\0001-WindchillConfigAssistant.log'

--------------------------------------------------------------------------------

!+! ENTERING Windchill Configuration
Assistant[<WCAinstall>\WindchillConfigAssistant.xml]
--------------------------------------------------------------------------------

Build sequence for target(s) `configureWindchill' is [initWCAVersion,
initMemoryModel, initWin, initUnix, declare, registerBuildListener,
initServerResource, initInputHandler, locateTomcat, locateApache, locateRDBMS,
checkOracleLocal, initWTContext, calcOracleMemoryRequirement, checkSQLServerLocal,
calcSQLServerMemoryRequirement, calcRDBMSMemoryRequirement, locateLDAP,
initInstaller, initWindchillJavaVersion, initWindchillVersion,
initIBMJavaDirectives, initJavaDirectives, init, backupXconfFile,
doInstallConfigRef, configureIBMJavaArgs, initJDBCDriverVersion,
checkDatabaseSupport, checkRepository, initWindchillStatus, checkTomcatEmbedded,
checkTomcatAJP13Listener, initServletEngineStatus, allocatePhysMem,
checkMasterCacheServer, configureMethodServerService, configureLoadBalancing,
configureServerManager, checkCacheSizeSkip, configureCacheManagerSizes,
configureDBProperties, checkRemoveTerminalWindows, removeTerminalWindows,
checkGCBaiterSupport, configureGCBaiter, configureWindchill]

initWCAVersion:
     [echo] WCA version: 9.1.1
     [echo] Ant version: Apache Ant version 1.7.1 compiled on June 27 2008

initMemoryModel:

initWin:

initUnix:
Skipped because property 'isOnUnix' not set.

declare:

registerBuildListener:

initServerResource:

initInputHandler:

locateTomcat:

locateApache:

locateRDBMS:
     [echo] wt.pom.jdbc.host=sh-dums01
     [echo] Database is not installed locally

checkOracleLocal:
Skipped because property 'isLocalRDBMS' not set.

initWTContext:
[wtcontextinit] wt.server.codebase=http://servername.com:80/Windchill

calcOracleMemoryRequirement:
Skipped because property 'OracleIsLocal' not set.

checkSQLServerLocal:
Skipped because property 'isLocalRDBMS' not set.

calcSQLServerMemoryRequirement:
Skipped because property 'SQLServerisLocal' not set.

calcRDBMSMemoryRequirement:
Skipped because property 'isLocalRDBMS' not set.

locateLDAP:
     [echo] wt.federation.ie.ldapServer=ldap://servername.com:3890
     [echo] LDAP hostname=servername.com
     [echo] LDAP server is installed locally
```

```
initInstaller:
     [echo] Apache home: <WSinstall>\Apache
     [echo] Tomcat home: <WSinstall>\Tomcat

initWindchillJavaVersion:
     [echo] Java version is 1.6.0_14

initWindchillVersion:
     [echo] PDMLink datecode=M040 release=9.1 releaseid=pdml.9.1.40.09 MOR=40
     [echo] ProjectLink datecode=M040 release=9.1 releaseid=pjl.9.1.40.09 MOR=40


initIBMJavaDirectives:
Skipped because property 'isIBMJVM' not set.

initJavaDirectives:

init:
     [echo] Number of CPUs detected: 8
     [echo] Physical Memory: 2047 MB
     [echo] Free Memory: 1972 MB
     [echo] Maximum Java heap size=1250 MB
     [echo] WT_HOME=<WSinstall>\Windchill
     [echo] VmVendor=Sun Microsystems Inc.
     [echo] Operating System=Windows 2003
     [echo] OS Arch=x86

backupXconfFile:
[propertyfile] Updating property file:
<WCAinstall>\conf\windchill\windchillconfigurator.xconf.build.number
     [move] Moving 1 file to <WCAinstall>\conf\windchill

doInstallConfigRef:
Skipped because property 'doPropagate' not set.

configureIBMJavaArgs:
Skipped because property 'isIBMJVM' not set.

initJDBCDriverVersion:
     [echo] Database Product: Oracle
     [echo] Database Major Version: 11
     [echo] Database Minor Version: 1
     [echo] Driver Name: Oracle JDBC driver
     [echo] JDBC Driver Major Version: 10
     [echo] JDBC Driver Minor Version: 2

checkDatabaseSupport:
     [echo] wt.db.dataStore=Oracle

checkRepository:
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  Repository
   [delete] Deleting: <WCAinstall>\conf\count.Repository.log

initWindchillStatus:

checkTomcatEmbedded:

checkTomcatAJP13Listener:
```

```
initServletEngineStatus:
      [get] Getting: http://servername.com:80/Windchill/wtcore/test/dynAnon.jsp
      [get] To: <WCAinstall>\urlTestServerResponse1967299706.tmp
      [get] Error opening connection java.io.IOException: Server returned HTTP
response code: 503 for URL:
http://p524623d.ptcnet.ptc.com:80/Windchill/wtcore/test/dynAnon.jsp
      [get] Error opening connection java.io.IOException: Server returned HTTP
response code: 503 for URL:
http://p524623d.ptcnet.ptc.com:80/Windchill/wtcore/test/dynAnon.jsp
      [get] Error opening connection java.io.IOException: Server returned HTTP
response code: 503 for URL:
http://p524623d.ptcnet.ptc.com:80/Windchill/wtcore/test/dynAnon.jsp
      [get] Can't get
http://p524623d.ptcnet.ptc.com:80/Windchill/wtcore/test/dynAnon.jsp to
<WCAinstall>\urlTestServerResponse1967299706.tmp
     [echo] ServletEngine is down - as expected
     [echo]

allocatePhysMem:
     [echo] OS memory requirement=409 MB
     [echo] LDAP memory requirement: 256 MB
     [echo] Minimum memory requirement for MethodServer: 192 MB
     [echo] Minimum memory requirement for ServerManager: 16 MB
     [echo] Minimum memory requirement for ServletEngine: 128 MB
     [echo] Windchill Requires a minimum of 336 MB
     [echo] Total Memory requirement: 1001.0 MB
     [echo] Adjusting free memory based on 3rd party application memory
requirements: 1972->1382 MB
     [echo] Allocating Windchill Memory region from free memory
     [echo] % free memory allocated to Windchill: 80
     [echo] Windchill Memory Region= 1105 MB
     [echo] % Windchill memory to allocate to MethodServers: 75
     [echo] % Windchill memory to allocate to ServerManager: 5
     [echo] % Windchill memory to allocate to ServletEngine: 20
     [echo] MethodServer memory region..... 828 MB
     [echo] ServerManager memory region..... 55 MB
     [echo] ServletEngine memory region..... 221 MB

checkMasterCacheServer:
     [echo] ServerManager is not MasterCacheServer

configureMethodServerService:
     [echo] Recommended number of MethodServers for 8 CPUs = 3
     [echo] Recommended number of MethodServers for 828MB on Windows 2003(32bit is
1
     [echo] Recommended number of MethodServers based on available resources=1
     [echo] Setting max heap size for MethodServer to 828
     [echo] Setting initial heap size for MethodServer to 828
 [xconfadd] New value for property=wt.manager.monitor.start.BackgroundMethodServer,
new= 0, current=<UNDEFINED>
     [echo] Setting sun.rmi.dgc intervals for MethodServer
     [echo] java.exe -server -Djava.awt.headless=true -classpath "<snip>" -noverify
-Dwt.logs.dir=<WSinstall>\Windchill\logs -Dwt.manager.serviceName={1} -
Dcom.sun.management.jmxremote -
Djmx.remote.protocol.provider.pkgs=wt.jmx.remote.protocol  -
Dlog4j.configuration=WEB-INF/log4jMethodServer.properties  -Xms828m -Xmx828m  -
XX:NewSize=332m -XX:MaxNewSize=332m -XX:SurvivorRatio=4  -XX:PermSize=72m -
XX:MaxPermSize=192m -Xloggc:<WSinstall>\Windchill\logs\%Y%m%d%H%M%S-GC.log -
XX:+PrintGCDetails -XX:+PrintGCTimeStamps -Dsun.rmi.dgc.client.gcInterval=3600000 -
Dsun.rmi.dgc.server.gcInterval=3600000
     [echo] Attempting to start JVM with these command line args to print java
version
     [exec] java version "1.6.0_14"
     [exec] Java(TM) SE Runtime Environment (build 1.6.0_14-b08)
     [exec] Java HotSpot(TM) Server VM (build 14.0-b16, mixed mode)
     [echo] MethodServer command line OK...
     [echo] Set wt.queue.executeQueues=true
 [xconfadd] New value for property=wt.method.maxHeap, new= 828, current=256
 [xconfadd] New value for property=wt.method.minHeap, new= 828, current=128
 [xconfadd] New value for property=wt.manager.cmd.MethodServer.nonibm.java.args,
new=  -XX:NewSize=332m -XX:MaxNewSize=332m -XX:SurvivorRatio=4  -XX:PermSize=72m -
XX:MaxPermSize=192m -Xloggc:$(wt.logs.dir)$(dir.sep)%Y%m%d%H%M%S-GC.log -XX:+
PrintGCDetails -XX:+PrintGCTimeStamps, current=-XX:PermSize=72m -
XX:MaxPermSize=200m
 [xconfadd] New value for property=wt.manager.cmd.MethodServer.ibm.java.args, new=
-Xloggc:$(wt.logs.dir)$(dir.sep)%Y%m%d%H%M%S-GC.log -XX:+PrintGCDetails -
XX:+PrintGCTimeStamps, current=<UNDEFINED>
 [xconfadd] New value for property=wt.manager.cmd.MethodServer.java.extra.args,new=
-Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000,
current=<UNDEFINED>

configureLoadBalancing:
     [echo] Skip loadbalancing for single MethodServer

configureServerManager:
     [echo] Setting max heap size for ServerManager to 55
     [echo] Setting initial heap size for ServerManager to 55
     [echo] Skip configure Remote JMX access for ServerManager
     [echo] Disable ExplicitGC for ServerManager
     [echo] java.exe -server -Djava.awt.headless=true -classpath "<snip>" -noverify
-Dwt.logs.dir=<WSinstall>\Windchill\logs -Dwt.manager.serviceName={1} -
Dcom.sun.management.jmxremote -
Djmx.remote.protocol.provider.pkgs=wt.jmx.remote.protocol  -
Dlog4j.configuration=WEB-INF/log4jServerManager.properties  -Xms55m -Xmx55m -
XX:+DisableExplicitGC  -XX:NewSize=11m -XX:MaxNewSize=11m -XX:SurvivorRatio=8  -
XX:PermSize=64m -XX:MaxPermSize=64m-Xloggc:<WSinstall>\Windchill\logs\%Y%m%d%H%M%S-
GC.log -XX:+PrintGCDetails -XX:+PrintGCTimeStamps
     [echo] Attempting to start JVM with these command line args...
     [exec] java version "1.6.0_14"
     [exec] Java(TM) SE Runtime Environment (build 1.6.0_14-b08)
     [exec] Java HotSpot(TM) Server VM (build 14.0-b16, mixed mode)
     [echo] ...ServerManager command line OK...
 [xconfadd] New value for property=wt.manager.maxHeap, new= 55, current=64
 [xconfadd] New value for property=wt.manager.minHeap, new= 55, current=32
 [xconfadd] New value for property=wt.manager.cmd.ServerManager.nonibm.java.args,
new=  -XX:NewSize=11m -XX:MaxNewSize=11m -XX:SurvivorRatio=8  -XX:PermSize=64m -
XX:MaxPermSize=64m -Xloggc:$(wt.logs.dir)$(dir.sep)%Y%m%d%H%M%S-GC.log -
XX:+PrintGCDetails -XX:+PrintGCTimeStamps, current=<UNDEFINED>
 [xconfadd] New value for property=wt.manager.cmd.ServerManager.ibm.java.args, new=
-Xloggc:$(wt.logs.dir)$(dir.sep)%Y%m%d%H%M%S-GC.log -XX:+PrintGCDetails -
XX:+PrintGCTimeStamps, current=<UNDEFINED>
 [xconfadd] New value for property=wt.manager.cmd.ServerManager.gc.log.args, new= -
Xloggc:$(wt.logs.dir)$(dir.sep)%Y%m%d%H%M%S-GC.log -XX:+PrintGCDetails -
XX:+PrintGCTimeStamps, current=$(wt.manager.cmd.MethodServer.gc.log.args)
 [xconfadd] New value for property=wt.manager.cmd.ServerManager.debug.args, new=
 $(wt.manager.cmd.ServerManager.gc.log.args), current=<UNDEFINED>
 [xconfadd] New value for property=wt.manager.cmd.ServerManager.platform.java.args,
new= $(wt.manager.cmd.ServerManager.nonibm.java.args), current=<UNDEFINED>
 [xconfadd] New value for property=wt.manager.cmd.ServerManager.java.extra.args,
new= $(wt.manager.cmd.ServerManager.platform.java.args) -XX:+DisableExplicitGC,
current=<UNDEFINED>

checkCacheSizeSkip:
```

```
configureCacheManagerSizes:
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  PolicyAcl
   [delete] Deleting: <WCAinstall>\conf\count.PolicyAcl.log
     [echo] PolicyAcl rowcount=177
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  AdministrativeDomain
   [delete] Deleting: <WCAinstall>\conf\count.AdministrativeDomain.log
     [echo] AdministrativeDomain rowcount=58
     [echo] Configuring wt.cache.size.AclCache currentSize=200 rowcount=177
percent=10
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  RemoteObjectInfo where classnamekeyA3 in
('wt.org.WTGroup', 'wt.org.WTUser', 'wt.org.WTOrganization')
   [delete] Deleting: <WCAinstall>\conf\count.RemoteObjectInfo.log
     [echo] RemoteObjectId rowcount=19
     [echo] Configuring wt.cache.size.StandardUfidSrvService$RemoteObjectIdCache
currentSize=${wt.cache.size.StandardUfidSrvService$RemoteObjectIdCache} rowcount=19
percent=50
     [echo] Configuring wt.cache.size.FederatableServerHelper$RemoteObjectIdCache
currentSize=1000 rowcount=19 percent=50
    [input] skipping input as property RemoteObjectInfo.input.cacheSize has already
been set.
 [xconfadd] New value for
property=wt.cache.size.FederatableServerHelper$RemoteObjectIdCache, new= 100,
current=1000
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  ContainerTeam
   [delete] Deleting: <WCAinstall>\conf\count.ContainerTeam.log
     [echo] ContainerTeam rowcount=20
     [echo] Configuring
wt.services/rsc/default/ObjectReferenceCacheTable/ContainerTeamCache.Size/null/0
currentSize=${wt.services/rsc/default/ObjectReferenceCacheTable/ContainerTeamCache.
Size/null/0} rowcount=20 percent=10
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  PDMLinkProduct
   [delete] Deleting: <WCAinstall>\conf\count.PDMLinkProduct.log
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  Project2
   [delete] Deleting: <WCAinstall>\conf\count.Project2.log
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  WTLibrary
   [delete] Deleting: <WCAinstall>\conf\count.WTLibrary.log
     [echo] WTContainer rowcount=20
     [echo] Configuring
wt.services/rsc/default/ObjectReferenceCacheTable/ContainerCache.Size/null/0
currentSize=${wt.services/rsc/default/ObjectReferenceCacheTable/ContainerCache.Size
/null/0} rowcount=20 percent=10
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  WTUser
   [delete] Deleting: <WCAinstall>\conf\count.WTUser.log
     [echo] Number of WTUsers=2
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  WTGroup
   [delete] Deleting: <WCAinstall>\conf\count.WTGroup.log
     [echo] Number of WTGroups=385
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  MembershipLink where
classnamekeyroleBObjectRef='wt.org.WTUser'
   [delete] Deleting: <WCAinstall>\conf\count.MembershipLink.log
     [echo] Number of User/Group MembershipLinks=44
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select avg(A.c) from (select count(*) c from MembershipLink where
classnamekeyroleBObjectRef='wt.org.WTUser' group by idA3B5) A
   [delete] Deleting: <WCAinstall>\conf\count.A.log
     [echo] Average number of groups per user=22
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select avg(B.c) from (select count(*) c from MembershipLink where
classnamekeyroleBObjectRef='wt.org.WTUser' group by idA3A5) B
   [delete] Deleting: <WCAinstall>\conf\count.B.log
     [echo] Average number of users per group=1.1
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(C.c) from (select distinct idA3A5 c from MembershipLink
where classnamekeyroleBObjectRef='wt.org.WTUser' ) C
   [delete] Deleting: <WCAinstall>\conf\count.C.log
     [echo] Total number of parent groups=40
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  WTOrganization
   [delete] Deleting: <WCAinstall>\conf\count.WTOrganization.log
     [echo] Number of WTOrganizations=2
     [echo] Named users: 100
     [echo] WTPrincipal rowcount= 100(WTUsers) + 40(ParentGroups) +
2(WTOrganizations) = 142
     [echo] Configuring wt.cache.size.WTPrincipalCache currentSize=20000
rowcount=142 percent=50
     [echo] Expected peak active users: 10
     [echo] Configuring wt.cache.size.SessionCache currentSize=500 rowcount=100
percent=20
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  UIAccess
   [delete] Deleting: <WCAinstall>\conf\count.UIAccess.log
     [echo] Configuring wt.cache.size.RoleAccessCache currentSize=1000
rowcount=1213 percent=100
 [xconfadd] New value for property=wt.cache.size.RoleAccessCache, new= 1213,
current=1000
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  PreferenceInstance
   [delete] Deleting: <WCAinstall>\conf\count.PreferenceInstance.log
     [echo] Number of PreferenceInstance=141
     [echo] Configuring wt.cache.size.PreferenceCache
currentSize=${wt.cache.size.PreferenceCache} rowcount=1014 percent=100
 [xconfadd] New value for property=wt.cache.size.PreferenceCache, new= 1014,
current=100
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  AdministrativeDomain
   [delete] Deleting: <WCAinstall>\conf\count.AdministrativeDomain.log
     [echo] Configuring wt.admin.cache.maxDomains currentSize=2000 rowcount=58
percent=25
```

```
     [echo] Configuring wt.cache.size.PagingSessionCache currentSize=100
rowcount=100 percent=5
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  BooleanDefinition
   [delete] Deleting: <WCAinstall>\conf\count.BooleanDefinition.log
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  FloatDefinition
   [delete] Deleting: <WCAinstall>\conf\count.FloatDefinition.log
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  IntegerDefinition
   [delete] Deleting: <WCAinstall>\conf\count.IntegerDefinition.log
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  RatioDefinition
   [delete] Deleting: <WCAinstall>\conf\count.RatioDefinition.log
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  ReferenceDefinition
   [delete] Deleting: <WCAinstall>\conf\count.ReferenceDefinition.log
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  StringDefinition
   [delete] Deleting: <WCAinstall>\conf\count.StringDefinition.log
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  TimestampDefinition
   [delete] Deleting: <WCAinstall>\conf\count.TimestampDefinition.log
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  UnitDefinition
   [delete] Deleting: <WCAinstall>\conf\count.UnitDefinition.log
   [sqlptc] Executing commands
   [sqlptc] 1 of 1 SQL statements executed successfully
     [echo] select count(*) from  URLDefinition
   [delete] Deleting: <WCAinstall>\conf\count.URLDefinition.log
     [echo] Total IBA DEFINITIONS=61
     [echo] Configuring wt.cache.size.IBADefViewManager$IBADefViewCache
currentSize=${wt.cache.size.IBADefViewManager$IBADefViewCache} rowcount=61
percent=100

configureDBProperties:
     [echo] MethodServer Max Heap Size=828
 [xconfadd] New value for property=wt.pom.maxDbConnections, new= 10, current=5
     [echo] Setting wt.pom.maxDbConnections=10
     [echo] Setting wt.pom.minDbConnections=5
 [xconfadd] New value for property=wt.pom.minDbConnections, new= 5, current=10
     [echo] Setting wt.pom.StatementCacheSize=50
     [echo] Setting wt.pom.rowPrefetchCount=20
     [echo] Setting wt.pom.refreshCache.size=100
 [xconfadd] New value for property=wt.pom.refreshCache.size, new= 100,
current=<UNDEFINED>
     [echo] Setting wt.pom.paging.snapshotQueryLimit=5000
 [xconfadd] New value for property=wt.pom.paging.snapshotQueryLimit, new= 5000,
current=-1
     [echo] Setting com.ptc.windchill.search.queryLimit=5000
 [xconfadd] New value for property=com.ptc.windchill.search.queryLimit, new= 5000,
current=<UNDEFINED>
     [echo] Setting wt.pom.queryLimit=-1
 [xconfadd] New value for property=wt.pom.queryLimit, new= -1, current=<UNDEFINED>
     [echo] Setting wt.pom.inClauseBindOptimizationCardinality=-1
 [xconfadd] New value for property=wt.pom.inClauseBindOptimizationCardinality, new=
-1, current=<UNDEFINED>
     [echo] Setting wt.pom.inClauseUseBindOptimization=true
 [xconfadd] New value for property=wt.pom.inClauseUseBindOptimization, new= true,
current=<UNDEFINED>
 [xconfadd] New value for property=wt.pom.dbConnectionsHardLimit, new= 15,
current=<UNDEFINED>

checkRemoveTerminalWindows:

removeTerminalWindows:
Skipped because property 'doRemoveTerminalWindows' not set.

checkGCBaiterSupport:

configureGCBaiter:
     [xslt] Processing <WSinstall>\Windchill\codebase\WEB-
INF\methodServerMBeanConfig.xml to
<WCAinstall>\conf\windchill\methodServerMBeanConfig.xml
     [xslt] Loading stylesheet <WCAinstall>\GCBaiter.xsl
     [xslt] Add minAvailableHeapThreshold: 42991616

configureWindchill:
 [xconfadd] New value for property=com.ptc.windchill.wca.buildDate, new=
20100427145047, current=<UNDEFINED>
 [xconfadd] Non numeric value=20100427145047

------------------------------------------------------------------------------

!+! ENTERING ANTCALL WITHIN Windchill Configuration
Assistant[<WCAinstall>\WindchillConfigAssistant.xml]
------------------------------------------------------------------------------

Build sequence for target(s) `doPropagateWindchill' is [initWCAVersion,
initMemoryModel, initWin, initUnix, declare, registerBuildListener,
initServerResource,initInputHandler, locateTomcat, locateApache, locateRDBMS,
checkOracleLocal, initWTContext, calcOracleMemoryRequirement, checkSQLServerLocal,
calcSQLServerMemoryRequirement, calcRDBMSMemoryRequirement, locateLDAP,
initInstaller, initWindchillJavaVersion, initWindchillVersion,
initIBMJavaDirectives, initJavaDirectives,init, checkXconfExists,
checkConfigRefExists, doPropagateWindchill]
  [antcall] Entering <WCAinstall>\WindchillConfigAssistant.xml...
Build sequence for target(s) `doPropagateWindchill' is [initWCAVersion,
initMemoryModel, initWin, initUnix, declare, registerBuildListener,
initServerResource,initInputHandler, locateTomcat, locateApache, locateRDBMS,
checkOracleLocal, initWTContext, calcOracleMemoryRequirement, checkSQLServerLocal,
calcSQLServerMemoryRequirement, calcRDBMSMemoryRequirement, locateLDAP,
initInstaller, initWindchillJavaVersion, initWindchillVersion,
initIBMJavaDirectives, initJavaDirectives,init, checkXconfExists,
checkConfigRefExists, doPropagateWindchill]

initWCAVersion:
Skipped because property 'wcaVersionDone' set.

initMemoryModel:
Skipped because property 'memoryModelDone' set.

initWin:

initUnix:
Skipped because property 'isOnUnix' not set.

declare:
```

```
registerBuildListener:

initServerResource:

initInputHandler:

locateTomcat:
Skipped because property 'TOMCAT_HOME' set.

locateApache:
Skipped because property 'APACHE_HOME' set.

locateRDBMS:
Skipped because property 'RDBMSLocated' set.

checkOracleLocal:
Skipped because property 'isLocalRDBMS' not set.

initWTContext:
[wtcontextinit] wt.server.codebase=http://servername.com:80/Windchill

calcOracleMemoryRequirement:
Skipped because property 'OracleIsLocal' not set.

checkSQLServerLocal:
Skipped because property 'isLocalRDBMS' not set.

calcSQLServerMemoryRequirement:
Skipped because property 'SQLServerisLocal' not set.

calcRDBMSMemoryRequirement:
Skipped because property 'isLocalRDBMS' not set.

locateLDAP:
     [echo] wt.federation.ie.ldapServer=ldap://servername.com:3890
     [echo] LDAP hostname=servername.com
     [echo] LDAP server is installed locally

initInstaller:
Skipped because property 'installerDone' set.

initWindchillJavaVersion:
Skipped because property 'javaVersionDone' set.

initWindchillVersion:
Skipped because property 'versionDone' set.

initIBMJavaDirectives:
Skipped because property 'isIBMJVM' not set.

initJavaDirectives:

init:
Skipped because property 'initDone' set.

checkXconfExists:
Skipped because property 'doPropagate' not set.

checkConfigRefExists:
Skipped because property 'doPropagate' not set.
```

```
doPropagateWindchill:
Skipped because property 'doPropagate' not set.
  [antcall] Exiting <WCAinstall>\WindchillConfigAssistant.xml.

--------------------------------------------------------------------------------

!+! EXITING FROM ANTCALL WITHIN Windchill Configuration
Assistant[<WCAinstall>\WindchillConfigAssistant.xml]
--------------------------------------------------------------------------------



--------------------------------------------------------------------------------

!+! ENTERING ANTCALL WITHIN Windchill Configuration
Assistant[<WCAinstall>\WindchillConfigAssistant.xml]
--------------------------------------------------------------------------------

Build sequence for target(s) `-copyLogs' is [-copyLogs]
  [antcall] Entering <WCAinstall>\WindchillConfigAssistant.xml...
Build sequence for target(s) `-copyLogs' is [-copyLogs]

-copyLogs:

--------------------------------------------------------------------------------

!+! ENTERING ANTCALL WITHIN Windchill Configuration
Assistant[<WCAinstall>\WindchillConfigAssistant.xml]
--------------------------------------------------------------------------------

Build sequence for target(s) `-doCopyLogs' is [-doCopyLogs]
  [antcall] Entering <WCAinstall>\WindchillConfigAssistant.xml...
Build sequence for target(s) `-doCopyLogs' is [-doCopyLogs]

-doCopyLogs:
     [copy] Copying 2 files to <WCAinstall>\logs
  [antcall] Exiting <WCAinstall>\WindchillConfigAssistant.xml.

--------------------------------------------------------------------------------

!+! EXITING FROM ANTCALL WITHIN Windchill Configuration
Assistant[<WCAinstall>\WindchillConfigAssistant.xml]
--------------------------------------------------------------------------------

  [antcall] Exiting <WCAinstall>\WindchillConfigAssistant.xml.

--------------------------------------------------------------------------------

!+! EXITING FROM ANTCALL WITHIN Windchill Configuration
Assistant[<WCAinstall>\WindchillConfigAssistant.xml]
--------------------------------------------------------------------------------

********************************************************************************

!+! FINISHED @ Tue Apr 27 14:50:48 CST 2010
!+! Log: <WSinstall>\Windchill\buildlogs\0006-WindchillConfigAssistant.log
********************************************************************************


BUILD SUCCESSFUL
Total time: 2 minutes 10 seconds
[buildListener]
[buildListener] Windchill Configuration Assistant Build Successful
```