WebSphere Lombardi Edition Version 7.1

Automated Runtime Environment Installation and Configuration Guide for UNIX

IBM®

Table of Contents

Introduction

This guide explains how to use the Automated Installer to install and configure Lombardi Process Server and Performance Data Warehouse in a runtime environment. During process development efforts, you can install process applications on Process Servers in configured runtime environments such as staging, test, and, finally, production. The runtime environments enable you to first test and refine your processes before finally rolling them out to your production audience. See Lombardi product components and Lombardi installation options for more information.

Lombardi product components

IBM WebSphere Lombardi Edition includes the following components. The following diagram illustrates how these components are commonly configured.

Component	Function
Process Center	Provides a central development environment and repository for multiple process authors working in the Process Center Console and other interfaces in Lombardi Authoring Environment. The Process Center includes a Process Center Server and a Performance Data Warehouse, allowing you to build and run process applications and also store performance data for testing and playback purposes during development efforts.
Process Server	Executes the processes and services built in Lombardi Authoring Environment, stored in the Process Center repository, and then installed in a runtime environment.
Performance Data Warehouse	Collects and aggregates process data according to tracking requirements established in Lombardi Authoring Environment.
Process Center Console	Enables you to manage and maintain the Lombardi repository, including management of process applications, workspaces, and snapshots. Also enables installation of process applications on Process Servers in runtime environments.
Authoring Environment	Lombardi Authoring Environment consists of several interfaces to enable process authors to model, implement, simulate, and inspect business processes.
Process Portal	Provides an interface that enables process participants to perform assigned tasks, view the history of tasks, and view the performance of their processes and teams. Using Process Portal, process participants can connect to the Process Center Server or a Process Server in any configured runtime environment, such as test or production environments.
Process Admin Console	Provides an interface that enables administrators to configure and maintain Lombardi Process Servers in any configured runtime environment, such as test or production environments. Also enables administrators to configure and maintain the Process Center Server.
Performance Admin Console	Provides an interface that enables administrators to configure and maintain Lombardi Performance Data Warehouses in any configured runtime environment, such as test or production environments. Also enables administrators to configure and maintain the Performance Data Warehouse included in the Process Center.

Lombardi architecture

The following diagram illustrates a typical IBM® WebSphere® Lombardi Edition configuration:
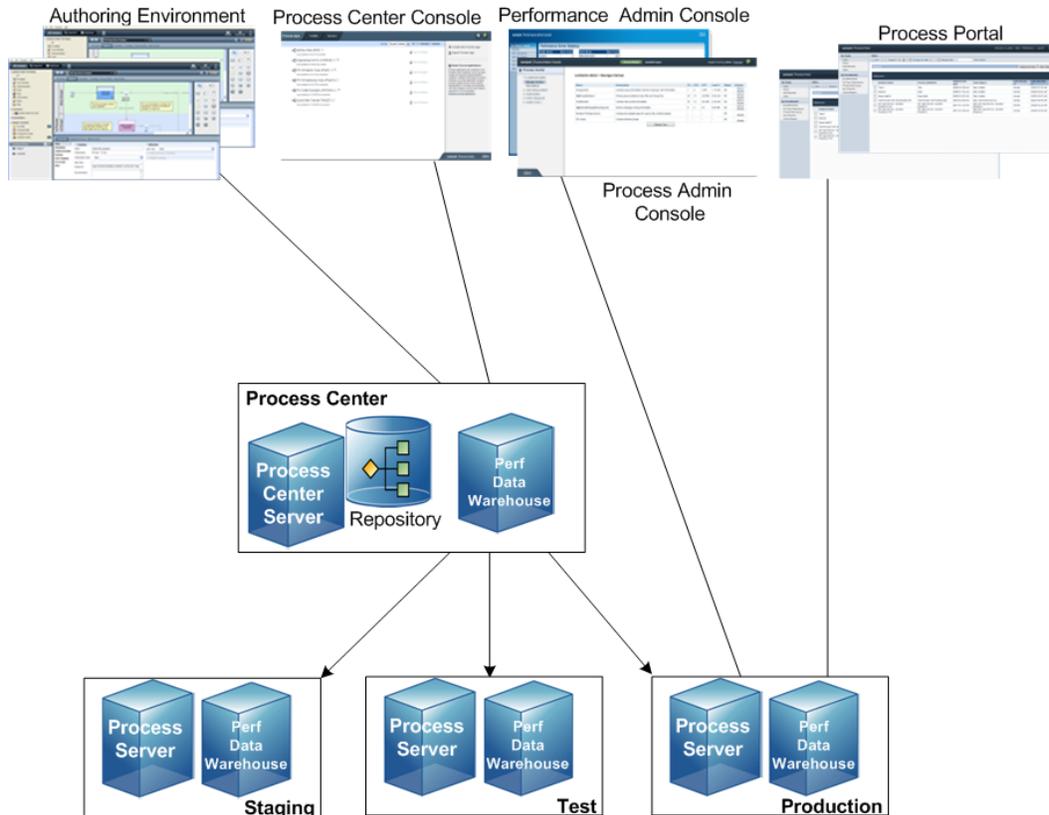
- From Lombardi Authoring Environment, multiple users connect to the Process Center.

- In Lombardi Authoring Environment, users create process models and supporting implementations (process applications) and store those applications and associated items in the Process Center repository. Authoring Environment users connected to the Process Center can share items.

- The Process Center includes a Process Center Server and Performance Data Warehouse, allowing users working in Lombardi Authoring Environment to run their process applications and store performance data for testing and playback purposes during development efforts.

- From the Process Center Console, administrators install process applications that are ready for staging, testing, or production on the Process Servers in those environments.

- From the Process Center Console, administrators manage running instances of process applications in all configured environments.

- From the Process Portal, end users perform assigned tasks. The Process Center Server and Process Servers in configured runtime environments can run the process applications that create the assigned tasks.

- Using the Process Portal, process participants can connect to the Process Center Server or a Process Server in any configured runtime environment, depending on whether a process is being developed, tested, or has been released to a production environment.

- Lombardi Performance Data Warehouse retrieves tracked data from the Process Server or Process Center Server at regular intervals. Users can create and view reports that leverage this data in Lombardi Authoring Environment and Process Portal.

- From the Process Admin Console and Performance Admin Console, administrators can manage and maintain all Lombardi servers.

Lombardi installation options

When you install Lombardi Process Center or runtime environment, the Lombardi Installer installs WebSphere® Application Server Version 7.0. The application server handles basic connections and containers for Lombardi product components, as well as underlying security and permissions.

The following table describes Lombardi installation options:

If you want to...	You should install...	For instructions, see...
Create process models, implement the process steps, and share your work by connecting to a central repository.	Lombardi Authoring Environment (Windows® only)	*Lombardi Authoring Environment Installation and Configuration Guide*
Provide a central repository and shared development environment for process authors.	Lombardi Process Center (includes Process Center Server, Performance Data Warehouse, and Authoring Environment)	*Lombardi Process Center Installation and Configuration Guide*
Install Lombardi servers in a runtime environment such as staging, test, or production.	Lombardi runtime environment (includes Process Server and Performance Data Warehouse)	*Lombardi runtime environment Installation and Configuration Guide*
Install an isolated Process Center for a single local user for demonstration purposes only.	Lombardi Process Center using Lombardi Installer's Simple option (Windows only) Do not use the Simple option to install a shared Process Center. For a shared installation, follow the instructions in *Lombardi Process Center Installation and Configuration Guide*.	*Lombardi Express Installation and Configuration Guide*
Automate installation of Lombardi Process Center.	Lombardi Process Center using the Automated Installer	*Lombardi Automated Process Center Installation and Configuration Guide*
Automate installation of Lombardi servers in a runtime environment such as staging, test, or production.	Lombardi runtime environment using the Automated Installer	Automating installation of Lombardi servers

Automating installation of Lombardi servers

Lombardi Automated Install enables you to automate installation of Lombardi servers using a command-line user interface. The following sections describe how you can use the Automated Installer to install Lombardi runtime environment. Because the runtime environment is commonly installed on different machines used for testing, staging, and production, the Automated Installer is particularly well suited for quickly installing the runtime environment on several different machines on your network.

Installing Lombardi runtime environment

If you want to install Lombardi servers in a test, production, or other environment, you can install Lombardi runtime environment on one or more machines on your network. The runtime environment install includes the following Lombardi components (see Lombardi product components for descriptions):

- Process Server

- Process Admin Console

- Process Portal

- Performance Data Warehouse

- Performance Admin Console

The Process Server in each Lombardi runtime environment can be connected to the Process Center. This enables Lombardi users to install snapshots of process applications from the Process Center Console to the connected Process Servers in test, production, and other environments. An example of this configuration (Process Center connected to multiple runtime environments) is shown in Lombardi architecture.

Steps for installing Lombardi runtime environment

The following tables list the steps required to run the Automated Install on your system.

You must have root access to install Lombardi runtime environment.

Pre-installation setup

Task	See...
1. Meet the hardware requirements.	Hardware requirements for installing Lombardi runtime environment
2. Meet the software requirements.	Supported software
3. Create Lombardi Process Server database and Lombardi Performance Data Warehouse database in your database server.	Creating Lombardi databases
4. Install X-Windows Virtual Frame Buffer.	Installing X-Windows Virtual Frame Buffer

Installation

Task	See...
1. Run Lombardi Installer to create the Automated Installer package on your system.	Creating the Automated Installer package on your system
2. View the contents of the Automated Installer package.	Viewing the Automated Installer package
3. Specify configuration parameters in the install.properties file.	Configuring the Automated Installer
4. Run the Automated Installer from the command line.	Installing Lombardi runtime environment using the Automated Installer

Post-installation configuration

Task	See...
For clustered installations in which you are creating a new cluster, start the deployment manager and synchronize the first node.	Configuring a new cluster
For clustered installations in which you are adding nodes to an existing cluster, run configuration scripts and perform other tasks.	Adding a node to an existing cluster
Run the installation scripts to load the initial data for Lombardi, including the system data toolkit and process apps. This step is required only if you do not enable the option to initialize your database.	Loading initial data
Start Lombardi Process Server and Lombardi Performance Data Warehouse.	Starting and stopping Lombardi servers
Configure external security providers.	Configuring an LDAP security provider
Configure accounts for Lombardi users.	Setting up Lombardi user accounts
Perform optional configuration tasks relevant to your environment.	Optional configuration tasks for Lombardi servers

Pre-installation setup

Before installing a Lombardi runtime environment, ensure that you meet the requirements described in the following sections.

Hardware requirements for installing Lombardi runtime environment

The following table describes the hardware requirements for installing Lombardi runtime environment:

By default, Lombardi Installer for Unix uses /tmp for its temporary directory space. You can change the temporary directory using the IATEMPDIR variable.

Lombardi installation	Minimum hardware required
Runtime environment	Dual processors, each 1.3 GHz, or a single 2.2 GHz processor 3 GB available RAM

Lombardi installation	Minimum hardware required
	4 GB free disk space for Lombardi servers, applications, and databases
	2 GB free disk space for additional database growth in development
	10 GB free disk space in your default temporary directory

Supported software

The following table describes the supported software for installation of Lombardi servers.

Software	Version
UNIX® operating systems	IBM® AIX® 6.1 (Power6)
	For AIX: The system JVM must be version pap6460sr6-20090925_01 or later to run Lombardi product components. If default file size of 1 GB is set, edit /etc/security/limits (as root) by changing fsize to -1 in the default section. This change is required to use Lombardi installer.
	Red Hat Enterprise Linux® 5 (x86-32)
	Red Hat Enterprise Linux 5 (x86-64)
	SUSE Linux Enterprise Server 10 (x86-32)
	SUSE Linux Enterprise Server 10 (x86-64)
	For Linux systems, you must install IBM Java SDK 6.0 prior to installation.
	Sun Solaris (Sparc) 10 (only 64-bit Sparc hardware is supported)
DB2® database	DB2 Version 9.7 Fix Pack 1
Oracle database	Oracle Enterprise 11g Release 1
	Oracle Standard 11g Release 1
	XA transactions must be enabled.
Microsoft SQL Server database	Microsoft® SQL Server 2005 SP3 (installed on a separate Windows® server)
	Ensure that your SQL Server installation is not case-sensitive (a case-sensitive SQL Server installation is not supported).
Client browsers	Microsoft Internet Explorer 6 and 7
	Mozilla Firefox 3

When you install Lombardi runtime environment, WebSphere Application Server Version 7.0 is installed. The application server handles basic connections and containers for Lombardi product components, as well as underlying security and permissions.

Creating Lombardi databases

You need to create a Lombardi Process Server database and a Lombardi Performance Data Warehouse database in your database server before performing Lombardi installation. The following sections provide guidelines for setting up Lombardi databases in your database server environment.

Creating databases in Microsoft SQL Server

Following are guidelines for configuring a SQL Server instance for Lombardi:

1. Enable the **SQL Server authentication mode** option.

2. Create a **Login user** (in the **Security** > **Logins** > **New Login** section for the database server), and be sure to remember the database user and password because you will need to provide them when installing Lombardi. (Lombardi Installer uses `lombardi` as the default database user and password.)

    The password policy specifies that the same value cannot be used for both user name and password. For the Login user, clear the **Enforce password policy** option if you want to use the same value for both the database user and password. If you want to apply the password policy for the Login user, enable the Enforce password policy option and specify a password value that is different from the user name value.

3. Create a database for Lombardi Process Server. For example, name the Process Server database `twProcessDB`. (Lombardi Installer uses this as the default database name.)

    To support Cyrillic characters, you must create the Process Server database with the following collation: SQL_Latin1_General_CP1251_CI_AS

4. Create a database for Lombardi Performance Data Warehouse. For example, name the Performance Data Warehouse database `twPerformanceDB`. (Lombardi Installer uses this as the default database name.)

    Ensure that the owner of each Lombardi database is the default owner and not the Login user created in step 2. If the new Login user is the database owner, when you try to create a database user with that Login in step 6, you will get an error.

5. For each database, set the **Parameterization** to `Forced` (in the **Properties** > **Options** > **Miscellaneous** section for each database).

6. For each database, create a **Database user** (in the **Security** > **Users** section for each database) whose name and password matches the Login user name. Under **Database role membership**, select `db_owner`.

7. For each database, create a **Schema** named `lombardi` (under **Security** > **Schemas** for the database). Assign the `lombardi` user as the schema owner and specify the `lombardi` schema as the default schema for each database.

For more information about installing and configuring the Microsoft SQL Server 2005 database server, see the following resources:

- SQL-Server-Performance.com: *Security enhancements in SQL Server 2005: Schema* http://www.sql-server-performance.com/articles/dba/authorization_2005_p1.aspx

- Microsoft TechNet: *Installing SQL Server 2005* http://technet.microsoft.com/en-us/library/ms143516.aspx

- Microsoft TechNet: *Forced Parameterization* http://technet.microsoft.com/en-us/library/ms175037.aspx

Creating databases in Oracle

You can create a new Oracle instance for use with Lombardi or you can use an existing instance. Following are guidelines for configuring an Oracle instance for Lombardi:

1. In Oracle Enterprise Manager, create a new user for Lombardi Process Server.

2. Supply a password for the Process Server user. It is important that you remember the database user and password because you will need to provide them when installing Lombardi. (Lombardi Installer uses `lombardi` as the default database user and password.)

3. Add the DBA role to the Process Server user.

4. In Oracle Enterprise Manager, create a new user for Lombardi Performance Data Warehouse.

5. Supply a password for the Performance Data Warehouse user. It is important that you remember the database user and password because you need to provide them when installing Lombardi. (Lombardi Installer uses `lombardi` as the default database user and password.)

6. Add the DBA role to the Performance Data Warehouse user.

Creating databases in DB2

The following is required for IBM DB2 databases.

> For more information about configuring DB2 databases, see the DB2 documentation at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp.

- When setting up local database users, assign names with eight characters or less. For example, `lombardiusr` as a user name will make the Lombardi database initialization scripts fail while `lombardi` as a user name will not.

- When creating the databases, enable the databases for Unicode (UTF-8 encoding) if you are using a non-ASCII character set such as Russian.

- Create a buffer pool with a page size that is 32 KB.

- Create a regular table space with a buffer pool that is 32 KB.

- Create a system temporary table space with a buffer pool that is 32 KB.

Installing X-Windows Virtual Frame Buffer

The X-Windows Virtual Frame Buffer (Xvfb) is necessary for Lombardi to properly render reports in a UNIX system. The X-Windows Virtual Frame Buffer needs to be running as root on the machine that runs Lombardi Process Server. If you are running Lombardi in a clustered environment, it needs to be installed on all Process Servers in your cluster.

When you begin the installation process, make sure that you set the following:

- Xvfb needs to be running on display 1. This is a flag that you can set when you start up. See Xvfb documentation for more information.

- Xvfb needs to be running in the background.

Xvfb setup may vary depending on your UNIX platform. For more information about Xvfb, see http://www.visualmining.com/documentation/xvfb.pdf.

Automated installation

To run the Automated Installer, you must create and configure the Automated Installer package on your system. The Automated Installer package is a set of binary files that Lombardi Installer copies to a specified directory on your system.

> Once you have created the Automated Installer package on your file system, you can copy the package to each target system where you want to install Lombardi servers. For example, if you want to install Lombardi runtime environments on several different machines on your network, you can copy the Automated Installer package to each machine and modify the configuration properties as needed to install each runtime environment.

The following sections describe how to:

1. Create the Automated Installer package on your system using Lombardi Installer.

2. View the contents of the Automated Installer package.

3. Configure the installation properties for the Automated Installer.

4. Install Lombardi runtime environment by running the Automated Installer from the command line.

> Lombardi Automated Installer is specifically designed to be integrated with other installation systems and can be easily configured to suit your needs. For more information, see Using ANT targets in an integrated installation.

Creating the Automated Installer package on your system

You must run Lombardi Installer to create the Automated Installer package on your system. Lombardi Installer for Unix operating systems uses a command-line interface that enables you to create the Automated Installer package, as described in the following steps.

Before you begin:

- As you complete the following steps, Lombardi Installer prompts you to enter values for specific installation parameters, which are described in the following tables. The values you supply will be used

to set the initial values of properties in the `install.properties` file that is included with the Automated Installer package.

- If you do not have all the values you need prior to creating the Automated Installer package, or if you enter incorrect values, the Automated Installer package will still be created but you will need to correct or supply the values in the `install.properties` file (see Configuring the Automated Installer for more information). You can always change the settings in the `install.properties` file when configuring the Automated Installer.

- The following steps assume that you already have created databases for Lombardi Process Server and Lombardi Performance Data Warehouse on your database server. For more information, see Creating Lombardi databases.

Steps to follow

1. Copy the WebSphere Lombardi Edition 7.1.0 installer to a temporary directory on your file system.

Several Unix installers are available. Be sure to copy and use the installer that is appropriate for your operating system.

2. From a shell prompt, run `./ [installer_name] .bin` to start the Installer.

If the installer prompts you to select another location, be sure to select a drive that has the required free disk space (10 GB) in the default temporary directory.

3. Read the Introduction, and then press **Enter**. You must supply each parameter requested by Lombardi Installer at the shell prompt, as described in the following steps. Press **Enter** to accept the default selection for the parameter.

If you make a mistake when supplying installation parameters, you can type `back` at any prompt to return to a previous step in the installation procedure.

If you press `Ctrl+C` to cancel the installation procedure and then attempt to reinstall, you might receive a "files are corrupted" error and the reinstallation fails. This is because the Installer creates temporary files that are not deleted when the installation is interrupted. You can resolve this issue by deleting the temporary files in the `/tmp` directory and then reinstalling.

If you must interrupt an installation procedure, an alternative to pressing `Ctrl+C` is to issue a `QUIT` command at any time during the installation procedure.

4. Read the Lombardi Software License Agreement. If you agree, select **I accept the terms of the license agreement** and then press **Enter**.

5. At the Choose Server Type panel, select one of the following runtime environment options and then press **Enter**.

Testing Processes or Business Playback
Staging Processes for Production, User Training, or User Acceptance
Production Process Execution

When you choose a runtime environment option, Lombardi Process Server and Performance Data Warehouse are installed on the same host.

If you want to run the Process Server without a Performance Data Warehouse, see Running a Process Server without a Performance Data Warehouse.

6. At the Process Server Options prompt, provide the following information for the runtime environment host:

Parameter name	Description
Host Name	The name of the machine on which you want to install Lombardi runtime environment. This field defaults to the current host.
HTTP Port	The port to use for Lombardi runtime environment on the host machine. The default port is 9086.
SMTP Server	The SMTP server that Lombardi runtime environment should use to send e-mail notifications.
Process Center Host	Specify the host name for the Process Center to which the Process Server in your runtime environment will connect.
Process Center HTTP Port	Specify the port that the Process Server in your runtime environment should use to connect to the Process Center.

To install an offline server that is not connected to the Process Center, type a space and press **Enter** instead of specifying a Process Center Host and a Process Center HTTP port when prompted.

7. At the **Enable clustering** prompt:

Type Y and press **Enter**	If you want to configure a server cluster for your runtime environment.
Type N and press **Enter**	If you do not want to configure a server cluster for your runtime environment.

8. If you enable clustering in the preceding step, specify whether to **Add to existing cluster**:

Type Y and press **Enter**	If you want to add a node to an existing server cluster.
Type N and press **Enter**	If you do not want to add a node to an existing server cluster.

When adding nodes to a cluster, be sure to install Lombardi product components in the same directory location for each node in your cluster.

If the **Enable clustering** option is enabled and the **Add to existing cluster** option is not enabled, the installer creates a new cluster (deployment manager plus the first node in the cluster).

9. The Database Options prompt enables you to choose the database that you want to use to store Lombardi data.

If you enable clustering in the preceding steps, be sure to specify the same database for a new cluster and each node that you add to that cluster.

After choosing the database that you want, supply the following information:

Parameter name	Description
Database Host	The name of the host machine for Lombardi Process Server and Performance Data Warehouse databases.
Port	The port that Lombardi should use to connect to the databases.
Initialize new database?	If you enable this option, the installer fully populates the Lombardi databases (specified in the following fields) with required data. If you do not enable this option, you need to initialize the Lombardi databases after installation as described in Loading initial data. If you enable clustering and elect to **Add to existing cluster** in the preceding steps, this option is disabled. When you add a node to an existing cluster, you must configure the new node before initializing Lombardi databases. See Adding a node to an existing cluster for more information.
Process Server Database Name	The name of the database created for the Process Server as instructed in Creating Lombardi databases. The default name is `twProcessDB`.
Process Server Database User	The name of the user who can access the specified database, as described in Creating Lombardi databases. The default user name is `lombardi`.
Process Server Database Password	The password required for the specified user to access the database as described in Creating Lombardi databases. The default password is `lombardi`.
Performance Data Warehouse Database Name	The name of the database created for the Performance Data Warehouse as instructed in Creating Lombardi databases. The default name is `twPerformanceDB`.
Performance Data Warehouse Database User	The name of the user who can access the specified database, as described in Creating Lombardi databases. The default user name is `lombardi`.
Performance Data Warehouse Database Password	The password required for the specified user to access the database as described in Creating Lombardi databases. The default password is `lombardi`.

Ensure that the database information is correct, as the values are case-sensitive.

When you are finished configuring the database parameters, press **Enter**.

10. At the Choose Install Folder prompt, provide the directory where Lombardi product components should be installed, and then press **Enter**.

> If you choose the option to create an Automated Installer Package at the following Pre-Installation Summary prompt, the directory that you specify is the directory where the Automated Installer package is created.
>
> When adding nodes to a cluster, be sure to install Lombardi product components in the same directory location for each node in your cluster.

11. On the Pre-Installation Summary panel, make sure that the information is correct. If not, go back to alter any incorrect configuration information.

    At the **Create Automated Installer Package** prompt:

Type Y and press **Enter**	If you want to use Lombardi Automated Installer. If you select this option, only the installer package is created in the installation directory; the product components are not installed.
Type N and press **Enter**	To install Lombardi product components in the installation directory.

If you enable clustering and elect to **Add to existing cluster** in step 8, the **Create Automated Installer Package** option is enabled by default. Adding a node to an existing cluster is not possible without creating an automated installation package.

Installation may take several minutes to complete.

12. The installer displays an Installation Complete message when it has finished creating the installer package.

When you run Lombardi Installer, an `installvariables.properties` file is created in the `[Lombardi_home]/UninstallerData` directory. This file may contain plain-text password values that are displayed in the Uninstaller during the removal of Lombardi product components. If this causes security concerns for you, delete this file after completing the installation.

Lombardi Installer creates the Automated Installer package in the directory that you specified. For more information about the contents of the Automated Installer package, see Viewing the Automated Installer package. Before you can run the Automated Installer, you must configure settings for the Automated Install in the `install.properties` file, as described in Configuring the Automated Installer.

Viewing the Automated Installer package

The Automated Installer package contains the files needed to automate the installation of Lombardi components on your system. To view these files, navigate to the directory where you created the Automated Installer package. The root of this directory contains subdirectories for individual Lombardi components as well as the files that enable you to configure and run the Automated Installer. These files are described in the following table.

File name	Description	To learn more...
`install.properties`	Contains configuration settings for the Automated Installer.	See Configuring the Automated Installer
`build-install.xml`	Contains ANT targets for performing installation tasks. This file references the `build-setup.xml` file, which contains predefined values for various operating systems, databases, and application servers.	See Installing Lombardi runtime environment using the Automated Installer See Using ANT targets in an integrated installation

Configuring the Automated Installer

Before you can use the Automated Installer to install Lombardi components, you must configure the `install.properties` file, which is located in the root of the directory where you created the Automated Installer package (see Creating the Automated Installer package on your system). The `install.properties` file is a standard Java™ property file that contains a collection of properties that enable you to customize your Lombardi installation. The values of the properties in the `install.properties` file reflect the values you supplied in Lombardi Installer when creating the Automated Installer package. If you need to make

changes to the values you specified in Lombardi Installer or customize other settings, you can set the properties to suit your environment and requirements.

The following tables list the properties in the `install.properties` file that are required to enable the Automated Installer to install Lombardi runtime environment. If you did not select a runtime server option in Lombardi Installer when creating the Automated Installer package, the values of some properties in the `install.properties` file may be incorrect.

Do not use special characters in the `install.properties` file when using ANT targets to install Lombardi. For example, you must use '/' in file paths instead of '\' because ANT considers the '\' character an escape character.

General Lombardi installation properties

Variable Name	Description
deploy.target.base	The directory where you want to install Lombardi components. This defaults to the directory where you created the Automated Installer package. Most other installation path properties are defined in terms of this property, either directly or indirectly. Always change the value of this property to avoid installing Lombardi components into the same directory as the Automated Installer package.
target.os	The operating system on which to install Lombardi. Possible values are `win` for Microsoft Windows and `linux` for UNIX-based operating systems.
temp.dir	The path of the directory that the Automated Installer and Lombardi use to hold temporary files. This value must end with a path separator ('/').
install.ps	Set the value to `false` for Process Center installations, so that the Automated Installer does not install a runtime Process Server. Set the value to `true` for runtime environments, so that the Automated Installer installs Lombardi Process Server. The Process Server is installed to the `[Lombardi_home]/process-server` directory.
install.ts	Set the value to `true` so that the Automated Installer installs Lombardi Performance Data Warehouse. The Performance Data Warehouse is installed to the `[Lombardi_home]/performance-data-warehouse` directory.
install.pcs	Set the value to `false` for runtime environment installations, so that the Automated Installer does not install Lombardi Process Center Server.
install.reporting	Lombardi comes with reporting functionality as provided by NetCharts. If you set this property to `true`, Lombardi components are installed with reporting capabilities. If set to `false`, Lombardi components are installed without NetCharts, and reporting will not be available.
install.examples	If set to `true`, the Automated Installer installs a set of examples of using Lombardi API to the `[Lombardi_home]/web-api` directory. These examples are based on the Java and .NET platforms.
install.web.files	If set to `true`, the Automated Installer installs Web server files to the `[Lombardi_home]/tw_web_files` directory.
install.setup.type	Always set to `Custom` for the Automated Installer. (The value `Simple` is used by the Express Installer.)
run.twinit	Set to `true` to run a default post-installation script that initializes databases and imports and deploys standard toolkits and process applications. See Loading initial data for more information about this script, including instructions for running it after installation, for example, if this property is set to `false`.
smtp.server	The URL of the SMTP server that Lombardi uses to send e-mail notifications.
startup.failure.to	Comma-separated e-mail addresses that receive notifications about startup failures.
startup.failure.from	An e-mail address used for sending notifications about startup failures.
startup.failure.name	The name of the sender used to send notifications about startup failures.

Variable Name	Description
em.enable	Set to `true` to enable Event Manager. Set to `false` to disable Event Manager, for example, if you want only some servers in a cluster to handle events.
em.paused	Set to `false` so that Event Manager begins processing events as soon as Process Server is started. Set to `true` so that Event Manager does not begin handling events when Process Server is started, for example, when you are performing an upgrade of Lombardi. Refer to *IBM WebSphere Lombardi Edition 7.1 Upgrade Guide* for more information on pausing and unpausing Event Manager when upgrading.
em.smtp.server	The URL of the SMTP server that the Event Manager uses to send notifications.
em.failure.email	The e-mail address from which the Event Manager sends notifications.
em.notify.error	The Lombardi user(s) whom the Event Manager notifies that a job failed.
em.user	The user name of the Lombardi user that the Event Manager runs as.
em.password	The (optionally encrypted) password of the Lombardi user that the Event Manager runs as.
stage.jdk.home	The path to the Java virtual machine to use during installation.

Properties for installing Lombardi servers in the application server environment

Variable Name	Description
iip.home.dir	Directory containing the WebSphere integrated installation package (IIP) for Lombardi. Defaults to `[Lombardi_Auto_Installer_directory]/was-iip`. Change this property only if you move the WebSphere IIP to a location that is different from the rest of the Automated Installer package.
websphere.platform.name	The name of the target operating system in the format expected by WebSphere Application Server.
appserver.root	The root directory of the application server installation (containing directories such as `bin` and `profiles`). Defaults to `${deploy.target.base}/AppServer`.
server.jvm.path	The path to the Java virtual machine for the Lombardi servers to use. Defaults to `${appserver.root}/java`.
websphere.profile	The WebSphere Application Server profile to use.
websphere.cluster.enabled.checked	Set to `true` if you want to configure a server cluster for a runtime environment. Otherwise, clustering is not enabled.
websphere.cluster.add	Set to `true` to add this node to an existing cluster. Otherwise, the installer creates a new cluster by creating a deployment manager and the first node in the cluster. This property is ignored if `websphere.cluster.enabled.checked` is not set to `true`.
appserver.jndiFactory	The Java class for the application server to use as an initial context factory for the directory service.
appserver.envname	The environment name can be any string that uniquely identifies your environment. It must be the same for all servers in a cluster.
appserver.envname.description	Describes the Process Server and Performance Data Warehouse environment.
appserver.cell	When adding Process Server to an existing cluster, specify the WebSphere Application Server cell to use to create the node. This must be a valid WebSphere cell name that is *different* from the cell that you are federating this server into.
appserver.node	When adding Process Server to an existing cluster, specify the WebSphere Application Server node to use for this Process Server. This must be a valid WebSphere node name that is *unique* among all nodes of the cell that are you federating this server into.
appserver.servername	The WebSphere Application Server name to use for Process Server. Change to a unique value other than the default when adding a Process Server to an existing cluster. See Adding a node to an existing cluster for more information.
appserver.virtualhost	The WebSphere Application Server virtual host name to use for Process Server.
appserver.dmgr.profile	The profile name of the deployment manager when you have a clustered WebSphere Application Server configuration for Lombardi Process Server.
script.websphere.twinit	Script to run to initialize databases, process applications, and toolkits. Whether this script is run is determined by the `run.twinit` property. This is the platform-independent form of the script name, without any platform-specific file extension.

For more information about embedded application server settings, see *WebSphere Application Server Version 7.0 Information Center*.

Properties for Lombardi Process Server

Variable Name	Description
ps.database	The type of database to which the Process Server saves all Lombardi data. The supported database types are `db2`, `mssql`, and `oracle11g`.
ps.database.use_windows_auth	This property is only needed when `ps.database` is set to `mssql`. If set to `true`, Microsoft SQL Server is configured to use Windows authentication.
ps.database.url	The JDBC URL to the Process Server database. This string might differ depending on the database type, host, and database name. Make sure the name of the Lombardi Process Server database in this URL is correct.
ps.database.username	The name of the user who can access the Process Server database.
ps.database.password	The password required for the specified user to access the Process Server database.
ps.database.additionalXaProperties	Distributed transaction properties for the Process Server database.
ps.database.driver	Fully qualified JDBC driver class name.
ps.database.jar.file	JAR archive name that contains the JDBC driver.
ps.host	Optional: The host used by the Process Server. This defaults to `localhost`. If the Process Server is to be used in a multi-user environment, `localhost` cannot be used; you must set the host to the name of the machine.
ps.port	Optional: The port used by the Process Server for Web access. Defaults to 9086 if no value is set.
ps.jndiPort	Optional: The port used for data access. Defaults to 2812 if no value is set.
ps.url	The URL used to access the HTTP service of the Process Server. Defaults to `http://[ps.host]:[ps.port]`.
ps.jndiUrl	The URL used to access the directory service of the Process Server. Defaults to `corbaname:iiop:[ ps.host]:[ ps.jndiPort]`. When adding a node to a cluster, edit this property to include the new node as described in Adding a node to an existing cluster.
ps.loglevel	The log level for the logs stored in `ps.logpath`.
ps.initpool	Number of connections to the database that the application server opens at start-up time. The default value is 2.
ps.maxpool	Maximum number of simultaneous connections used by the application server to connect to the Process Server database. The default value is 15.
ps.logpath	The directory where log files for the Process Server are stored. Defaults to `${appserver.root}/logs` if no value is set.
ps.minmem	The initial Java heap size (in MB) allocated for the Process Server.
ps.maxmem	The maximum Java heap size (in MB) allocated for the Process Server.
ps.enablePerformanceServerComm	If Performance Data Warehouse is used, this property must be set to `true`. If it is set to `true`, then the `ts.jndiUrl` property must be specified, in order for communication with the Performance Data Warehouse to work properly.

Variable Name	Description
script.websphere.ps.config	Script to run for configuration when installing Process Server. This is the platform-independent form of the script name, without any platform-specific file extension.
script.websphere.ps.deploy	Script to run for deployment when installing Process Server. This is the platform-independent form of the script name, without any platform-specific file extension.
script.websphere.ps.copypolicy	Script to run to copy the WebSphere Application Server policy file into WAR files. This is the platform-independent form of the script name, without any platform-specific file extension.

Properties for Lombardi Performance Data Warehouse

Variable Name	Description
ts.database	The type of database to which the Performance Data Warehouse writes data. The supported database types are db2, mssql, and oracle11g.
ts.database.use_windows_auth	This property is only needed when ts.database is set to mssql. If set to true, Microsoft SQL Server is configured to use Windows authentication.
ts.database.url	The JDBC URL to the Performance Data Warehouse database. The string might differ depending on the database type, host, and database name. Make sure the name of the Lombardi Performance Data Warehouse database in this URL is correct.
ts.database.username	The name of the user who can access the Performance Data Warehouse database.
ts.database.password	The password required for the specified user to access the Performance Data Warehouse database.
ts.database.additionalXaProperties	Distributed transaction properties for the database.
ts.database.driver	Fully qualified JDBC driver class name.
ts.database.jar.file	JAR archive name that contains the JDBC driver.
ts.host	Optional: The host used by Performance Data Warehouse. The default value is localhost. If the server is to be used in a multi-user environment, localhost cannot be used; you must set the host to the name of the machine.
ts.port	Optional: The port used by Performance Data Warehouse for Web access. Defaults to 9085 if no value is set.
ts.jndiPort	Optional: The port used for data access. Defaults to 2811 if no value is set.
ts.url	The URL used to access the HTTP service of the Performance Data Warehouse. Defaults to http://[ts.host]:[ts.port] if no value is set.
ts.jndiUrl	The URL used to access the directory service of the Performance Data Warehouse. Defaults to corbaname:iiop:[ts.host]:[ts.jndiPort]. When adding a node to a cluster, edit this property to include the new node as described in Adding a node to an existing cluster.
ts.loglevel	The log level for the logs stored in ts.logpath.
ts.initpool	Number of connections to the database that the application server opens at start-up time. The default value is 2.
ts.maxpool	Maximum number of simultaneous connections used by the application server to connect to the Performance Data Warehouse database. The default value is 15.
ts.logpath	The directory where log files for the Performance Data Warehouse are stored. Defaults to ${appserver.root}/logs if no value is set.
ts.minmem	The initial Java heap size (in MB) allocated for the Performance Data Warehouse.
ts.maxmem	The maximum Java heap size (in MB) allocated for the Performance Data Warehouse.

Variable Name	Description
twperfserver.servername	The WebSphere Application Server name to use for Performance Data Warehouse. Change to a unique value other than the default when adding a server to an existing cluster. See Adding a node to an existing cluster for more information.
twperfserver.virtualhost	The WebSphere Application Server virtual host name to use for Performance Data Warehouse.
script.websphere.ts.config	Script to run for configuration when installing Performance Data Warehouse. This is the platform-independent form of the script name, without any platform-specific file extension.
script.websphere.ts.deploy	Script to run for deployment when installing Performance Data Warehouse. This is the platform-independent form of the script name, without any platform-specific file extension.

Properties for communication with Lombardi Process Center Server

Variable Name	Description
pcs.host	Specify the name of the host machine where Lombardi Process Center Server is installed so that the runtime environment can connect to the Process Center.
pcs.port	Specify the port number of the host machine where Lombardi Process Center Server is installed so that the runtime environment can connect to the Process Center. This port is used by the Process Server in the runtime environment to access the HTTP service of the Process Center Server.
pcs.jndiPort	The port used for data access service. It must be specified unless the Process Center Server is being installed with the current installation.
pcs.url	The URL used to access the HTTP service of the Process Center Server. Defaults to `http://[pcs.host]:[pcs.port]` if no value is set.
pcs.jndiUrl	The URL used to access the directory service of the Process Center Server. Defaults to `jnp://[pcs.host]:[pcs.jndiPort]` if no value is set.
ps.heartbeat.interval	Specify the polling interval (in seconds) at which the Process Server in the runtime environment polls the Process Center Server for updates. Make sure the value is set to a whole number greater than 0. Defaults to 10 if no value is set. Defaults to -1 when you install an offline Process Server in your runtime environment.

If you are installing an offline Process Server that is not connected to the Process Center, the preceding properties should not be set. You can leave all settings blank and ensure that the heartbeat interval is set to 0 or a negative value.

The following properties exist in `install.properties` for historical reasons. Do not modify these properties.

Unmodifiable properties

Variable Name	Description
install.was	Must be set to `true`.
appserver.home	The home directory of the application server.
websphere.home	The home directory of the application server.
appserver.clienthome	The home directory of the application server.
target.queuemanager	Not supported.
mq.host	Not supported.
mq.port	Not supported.
mq.channel	Not supported.
ps.ear.expanded	Not supported.
ts.ear.expanded	Not supported.
ps.portal.expanded	Not supported.

Installing Lombardi runtime environment using the Automated Installer

After you configure the `install.properties` file (see Configuring the Automated Installer), you are ready to install Lombardi runtime environment using the Automated Installer.

The `build-install.xml` file located in the root of the directory where you created the Automated Installer package is a standard XML file containing ANT targets that perform particular tasks during a Lombardi installation. The `build-install.xml` file contains three top-level ANT targets that perform the entire installation and refer to the `install.properties` file for all installation parameters.

1. From a shell prompt, navigate to the root of the directory where you created the Automated Installer package (see Creating the Automated Installer package on your system).

2. Run the install target command, calling the three top-level targets as described in the following table. You must call the targets in the order shown.

    ```
    ant -buildfile build-install.xml
    ```

    Directly calling build files other than the root `build-install.xml` is not recommended.

Target	Requirements
unpack-was-iip-jdk	Must be run independently of the following two targets.
auto-install-was	Set JAVA_HOME to [auto_install_root]/was-iip-jdk before calling.
install	Set JAVA_HOME to [auto_install_root]/was-iip-jdk before calling.

You might want to capture the progress of the Lombardi installation to display in a user interface (for example, in a progress bar). The easiest way to accomplish this is to capture the output from ANT and then use keywords such as `install-ps:`.

Using ANT targets in an integrated installation

Lombardi Automated Installer is written in ANT script and can be integrated with another installer. You can write custom ANT scripts that utilize logic provided by Lombardi Automated Installer ANT scripts. ANT is a freely available Java-based open source product that can automate complex system tasks like installation.

The top-level ANT targets described in the table in the preceding section comprise a collection of lower-level targets. You can use the lower-level targets to gain more control over which Lombardi components are installed, and in what order. This is useful particularly when installing Lombardi components as part of a larger system.

The Apache ANT Web site provides technical documentation and other useful resources:

http://ant.apache.org/

Post-installation configuration

After installing Lombardi runtime environment on your system, you need to configure your Lombardi installation to enable the installed components to run, as described in the following sections.

The directory where you installed Lombardi product components is referred to as `[Lombardi_home]`.

Configuring a new cluster

If you enable clustering during Lombardi installation and do *not* add a node to an existing cluster, the installer creates a deployment manager plus the first node in the new cluster. After installation is complete, you need to start the deployment manager and synchronize the new node as described in the following steps.

For an automated installation, the `websphere.cluster.enabled.checked` property must be set to true to create a new cluster for a runtime environment.

1. After Lombardi installation completes, start the deployment manager using the following command:

   `[Lombardi_home]/AppServer/profiles/Dmgr01/bin/startManager.sh`

2. When the deployment managers starts, synchronize the deployment manager and the first node using the following command:

   `[Lombardi_home]/AppServer/profiles/Lombardi/bin/syncNode.sh [DM_host]`
   `[DM_SOAP_Connector_port] -user tw_user -password tw_user`

   Where:

[DM_host]	Is the name of the machine where the deployment manager is installed and running.
[DM_SOAP_connector_port]	(Not required if the deployment manager is running on its standard port.) Is the port to use for the SOAP connection to the deployment manager host. The deployment manager SystemOut.log includes this port. Search the logs for the text: `SOAP connector is available at port`
-user	Specifies the user for this operation. You must specify tw_user.
-password	Specifies the password of the user. You must specify the password for tw_user, which is tw_user by default.

3. If you did not select the option to initialize your databases during installation, follow the instructions in Loading initial data.

4. Start Lombardi product components as instructed in Starting and stopping Lombardi servers.

Adding a node to an existing cluster

If you enable clustering and elect to **Add to existing cluster** during installation, you need to complete the tasks outlined in the following sections.

Adding a node to an existing cluster requires an automated installation package and use of the automated installer.

Running Lombardi installer	Lists requirements for creating an installation package using the interactive installer.
Editing the install.properties file	Lists the settings to verify and change in the `install.properties` files before running an automated installation.
Adding cluster members	Describes how to add the newly installed node to the existing cluster using Lombardi commands and WebSphere administrative console.
Running configuration scripts for new cluster members	Describes how to run required configuration scripts for new cluster members.
Setting ports and virtual hosts for vertical clusters	Describes configuration required only for vertical clusters that use the Generate unique HTTP ports option.
Configuring the messaging engine	Describes required configuration for each new node that you add to the cluster.
Editing Lombardi configuration files	Describes configuration file edits required only if you did not edit the `ps.jndiUrl` and `ts.jndiUrl` properties in the `install.properties` file prior to automated installation.
Synchronizing nodes and starting Lombardi product components	Describes how to synchronize the nodes in your cluster and start Lombardi servers.

Running Lombardi installer

When you run the interactive Lombardi installer to create an installation package for the automated installer, be sure to:

- Install Lombardi product components in the same directory location for each node in your cluster.

- Specify the same databases when you install a new Lombardi cluster and when you install each node in that cluster.

Editing the install.properties file

When you change settings in the install.properties file before executing runtime server installation, you need to:

- Make sure that the `websphere.cluster.enabled.checked` and `websphere.cluster.add` properties are both set to true. (If you enabled these options using the interactive installer, they should be set to true.)

- Use the `appserver.cell` property to specify the WebSphere Application Server cell to use to create the node. This must be a valid WebSphere cell name that is different from the cell that you are federating this server into.

- Use the `appserver.node` property to specify the WebSphere Application Server node to use for this Process Server. This must be a valid WebSphere node name that is unique among all nodes of the cell that are you federating this server into.

- Change the `appserver.servername` and `twperfserver.servername` properties to unique values other than the defaults.

- Edit the `ps.jndiUrl` and `ts.jndiUrl` properties to include the new node using the following command syntax:

  ```
  corbaname:iiop:[your_host1]:[bootstrap_port1],: [your_host2]:[bootstrap_port2]
  ```

  Ensure you are using the same bootstrap addresses as defined for all servers in your cluster. You can check all ports in the WebSphere administration console by navigating to: **Servers** > **Server Types** > **WebSphere application servers** > **[server_name]** > **Communication** > **Ports**

  If you do not edit the `ps.jndiUrl` and `ts.jndiUrl` properties before installation, you can make the necessary changes after installation as described in Editing Lombardi configuration files.

Adding cluster members

After installation, you need to add the newly installed node to the Lombardi cluster environment using the following steps.

The deployment manager for the Lombardi cluster must be up and running before performing the following tasks. See Configuring a new cluster for more information.

1. Run the following command to add the node to the Lombardi cluster environment:

   ```
   [Lombardi_home]/AppServer/profiles/Lombardi/bin/addNode.sh [DM_host]
   [DM_SOAP_Connector_port] -noagent -user tw_user -password tw_user
   ```

   Where:

[DM_host]	Is the name of the machine where the deployment manager is installed and running.
[DM_SOAP_connector_port]	(Not required if the deployment manager is running on its standard port.) Is the port to use for the SOAP connection to the deployment manager host. The deployment manager SystemOut.log includes this port. Search the logs for the text: `SOAP connector is available at port`
-noagent	Directs the addNode command not to launch the node agent process for the new node. You must include this parameter. If not, you may have startup issues with the node agent.
-user	Specifies the user for this operation. You must specify tw_user.
-password	Specifies the password of the user. You must specify the password for tw_user, which is tw_user by default.

2. From the WebSphere administrative console, add a new cluster member to the Lombardi cluster using an existing node as a template. Perform this step for both Lombardi Process Server and Performance Data Warehouse clusters.

When adding a new cluster member using the WebSphere administrative console, be sure to:

- Use the same names that you specified for your new cluster members in the `install.properties` file. (See Editing the install.properties file.)

- Select the correct new node.

- Disable the option to generate unique HTTP ports unless you are adding a server on the same node or you are setting up a vertical cluster.

3. Ensure that you save to the master configuration before running the configuration scripts for new cluster members as described in the following section.

Running configuration scripts for new cluster members

You must run a configuration script for each new cluster member as outlined in the following steps.

1. Go to: `[Lombardi_home]/process-server/bin`

2. Execute the following command:

   ```
   configure_cluster_member.sh -cell [cell_name] -name [cluster_member_name]
   -schema [schema_name] {-conntype [SOAP|RMI] -host [host name] -port [port]
   -user [user_name] -password [password]}
   ```

   Where:

[cell_name]	Is the cell name for the cluster.
[cluster_member_name]	Is the name of the cluster member you have added as instructed in the preceding section, Adding cluster members.
[schema_name]	Is the schema name of the database that will store the internal SIB tables used by the new cluster member.
-conntype	Specifies the connection type to use to connect to the cluster host.
-host	Specifies the name of the machine that hosts the cluster.
-port	Specifies the port to use to connect to the cluster host.
-user	Specifies the user for this operation. You must specify tw_user.
-password	Specifies the password of the user. You must specify the password for tw_user, which is tw_user by default.

   For example:

   ```
   configure_cluster_member.sh -cell milesc-w500Cell01 -name twprocsvr2 -schema
   schemanode02 -conntype SOAP -host localhost -port 8879 -user tw_user -password
   tw_user
   ```

3. Go to: `[Lombardi_home]/performance-data-warehouse/bin`

4. Execute the following command:

   ```
   configure_cluster_member.sh -cell [cell_name] -name [cluster_member_name]
   -schema [schema_name] {-conntype [SOAP|RMI] -host [host name] -port [port]
   -user [user_name] -password [password]}
   ```

   Where:

[cell_name]	Is the cell name for the cluster.
[cluster_member_name]	Is the name of the cluster member you have added as instructed in the preceding section, Adding cluster members.
[schema_name]	Is the schema name of the database that will store the internal SIB tables used by the new cluster member.
-conntype	Specifies the connection type to use to connect to the cluster host.
-host	Specifies the name of the machine that hosts the cluster.
-port	Specifies the port to use to connect to the cluster host.
-user	Specifies the user for this operation. You must specify tw_user.
-password	Specifies the password of the user. You must specify the password for tw_user, which is tw_user by default.

For example:

```
configure_cluster_member.sh -cell milesc-w500Cell01 -name twperfsvr2 -schema
schemanode02 -conntype SOAP -host localhost -port 8879 -user tw_user -password
tw_user
```

Setting ports and virtual hosts for vertical clusters

These steps are required only for vertical clusters that use the **Generate unique HTTP ports** option.

1. In the WebSphere administrative console, go to: **Servers** > **Server Types** > **WebSphere application servers** > **[your_new_server]** > **Web Container Settings** > **Web Container Transport Chains** > **[default_inbound_transport_chain]** > **[inbound_channel]** > **Ports**

2. Ensure that the ports of the default inbound transport chain (by default, WCInboundDefault) are unique and that they are assigned to the virtual host of each cluster member.

> To view virtual host properties in the WebSphere administrative console, go to **Environment** > **Virtual Hosts** > **[your virtual host]** > **Host Aliases**.

For example, to change the ports for Lombardi Process Server, navigate to: **Servers** > **Server Types** > **WebSphere application servers** > **twprocsvr** > **Web Container Settings** > **Web Container Transport Chains** > **WCInboundDefault** > **TCP inbound channel (TCP_2)** > **Ports**

> Your cluster will not run unless the ports for the Web Container default host (by default, WC_defaulthost) are correctly assigned to the virtual host of each cluster member.

Configuring the messaging engine

Perform the following steps first for the nodes in Lombardi Process Server cluster and then for the nodes in Lombardi Performance Data Warehouse cluster.

1. In the WebSphere administrative console, go to: **Resources** > **JMS** > **Topic connection factories**

2. Select one of the listed resources, for example, **TWClientConnectionFactory**.

   By default, the **Provider endpoints** for JMS connections are configured for a single node (using node1_host:port syntax).

3. Change the **Provider endpoints** value to a comma-delimited list of all nodes using the syntax:
   `node1_host:port,node2_host:port`

> The port value is the SIB_ENDPOINT_ADDRESS that you can find by going to: **Servers** > **Server Types** > **WebSphere application servers** > **[server_name]** > **Communication** > **Ports**.

4. Make the same change for each Topic connection factory resource.

5. In the WebSphere administrative console, go to: **Resources** > **JMS** > **Queue connection factories**

6. Select one of the listed resources, for example, **QueueConnectionFactory**.

7. Change the **Provider endpoints** value to a comma-delimited list of all nodes as you did in the previous steps for the Topic connection factory resources.

8. Make the same change for each Queue connection factory resource.

Editing Lombardi configuration files

This section includes the following tasks:

Setting the JNDI URLs in your Lombardi configuration files	Necessary only if you did not edit the `ps.jndiUrl` and `ts.jndiUrl` properties in the `install.properties` file prior to automated installation.
Altering `cache-jms-jndi-lookups` setting	Necessary on each node of a clustered configuration.

Setting the JNDI URLs in your Lombardi configuration files is necessary only if you did not edit the `ps.jndiUrl` and `ts.jndiUrl` properties in the `install.properties` file prior to installation.

When adding a node to a cluster, you should configure the JNDI URLs in your Lombardi configuration files to include the new node using the following command syntax:

`corbaname:iiop:[your_host1]:[bootstrap_port1],: [your_host2]:[bootstrap_port2]`

> Ensure you are using the same bootstrap addresses as defined for all servers in your cluster. You can check all ports in the WebSphere administration console by navigating to: **Servers** > **Server Types** > **WebSphere application servers** > **[server_name]** > **Communication** > **Ports**

The Lombardi configuration files that require this information include the following:

- `00Static.xml`

- `50AppServer.xml`

- `80EventManager.xml`

You should always use the `100Custom.xml` file to change Lombardi configuration settings. For example, you should not change properties in
`[Lombardi_home]/process-server/config/system/00Static.xml` directly. You should instead add the configuration that you want to change to the
`[Lombardi_home]/process-server/config/100Custom.xml` file. Lombardi merges the changes that you make in the `100Custom.xml` file with the original configuration, as described in Changing Lombardi server properties.

The following example shows the entries needed in
`[Lombardi_home]/process-server/config/100Custom.xml` for Lombardi Process Server configuration:

```
<properties>

    <common merge="mergeChildren">

            <performance-server-communication merge="mergeChildren">
            <provider merge="mergeChildren">
                <url merge="replace">corbaname:iiop:Paris.mycompany.com:3811,:Rome.mycompany.com:3811</url>
            </provider>
        </performance-server-communication>

    </common>

    <server merge="mergeChildren">

            <cache merge="mergeChildren">
            <provider-url merge="replace">corbaname:iiop:Paris.mycompany.com:3812,:Rome.mycompany.com:3812</provider-url>
        </cache>

            <jndi-url merge="replace">corbaname:iiop:Paris.mycompany:3812,:Rome.mycompany.com:3812</jndi-url>

    </server>

    <event-manager merge="mergeChildren">
            <jndi merge="replace">
                    <env key="java.naming.provider.url" value="corbaname:iiop:Paris.mycompany:3812,:Rome.mycompany.com:3812" />
            </jndi>
    </event-manager>

</properties>
```

The following example shows the entries needed in
`[Lombardi_home]/performance-data-warehouse/config/100Custom.xml` for Lombardi Performance Data Warehouse configuration:

```
<properties>

    <server merge="mergeChildren">

            <jndi-url merge="replace">corbaname:iiop:Paris.mycompany.com:3811,:Rome.mycompany.com:3811</jndi-url>

    </server>
</properties>
```

For each node (server) in a cluster, you need to include the following entry in the
`[Lombardi_home]/process-server/config/100Custom.xml` file:

```
<properties>

    <common merge="mergeChildren">

            <cache-jms-jndi-lookups merge="replace">false</cache-jms-jndi-lookups>

    </common>

</properties>
```

Synchronizing nodes and starting Lombardi product components

After you have performed the configuration tasks necessary for your clustered environment, you can synchronize all nodes in your clusters and start Lombardi product components as described in the following steps.

1. If running, stop each node/server and then the deployment manger using the following commands:

    `[Lombardi_home]/AppServer/profiles/Lombardi/bin/stopNode.sh`

    `[Lombardi_home]/AppServer/profiles/Dmgr01/bin/stopManager.sh -user tw_user -password tw_user`

2. Start the deployment manager using the following command:

    `[Lombardi_home]/AppServer/profiles/Dmgr01/bin/startManager.sh`

3. When the deployment managers starts, synchronize the deployment manager and each node using the following command:

   ```
   [Lombardi_home]/AppServer/profiles/Lombardi/bin/syncNode.sh [DM_host]
   [DM_SOAP_Connector_port] -user tw_user -password tw_user
   ```

   Where:

[DM_host]	Is the name of the machine where the deployment manager is installed and running.
[DM_SOAP_connector_port]	(Not required if the deployment manager is running on its standard port.) Is the port to use for the SOAP connection to the deployment manager host. The deployment manager SystemOut.log includes this port. Search the logs for the text: `SOAP connector is available at port`
-user	Specifies the user for this operation. You must specify tw_user.
-password	Specifies the password of the user. You must specify the password for tw_user, which is tw_user by default.

4. Initialize your Lombardi databases by following the instructions in Loading initial data.

5. Start Lombardi product components on all nodes as instructed in Starting and stopping Lombardi servers.

Loading initial data

If you do not select the option to initialize your database during installation of Lombardi product components, you need to run an initialization script after installation is complete. When you run the initialization script, it creates the required schema and permissions for the Lombardi databases, imports system data, and (on runtime environments) deploys snapshots of imported toolkits and process applications.

1. From a command line console, navigate to your `[Lombardi_home] \twinit` directory.

2. Run the following command:

   ```
   twinit.sh
   ```

   The console should report a successful build.

   If the build fails, for example, due to database server connectivity or user permission issues, inspect the log file in the `twinit\logs` folder for errors. Troubleshoot the indicated issues and then run the script again. The script will resume from the last successful task and will not overwrite any existing data.

3. Start the management tool for your database and then verify that Lombardi database tables exist with data. The tables are named with a `lsw_` prefix (for example, `[schema_name].lsw_usr`).

Starting and stopping Lombardi servers

You can start Lombardi servers simultaneously or one at a time as described in the following procedures.

> If you have configured a server cluster for your runtime environment, be sure to start and stop all servers in the cluster.

To start both the Process Server and Performance Data Warehouse:

1. From a shell prompt, navigate to the `[Lombardi_home]/bin` directory.

2. Run `./startLombardi.sh`

To stop both the Process Server and Performance Data Warehouse:

1. From a shell prompt, navigate to the `[Lombardi_home]/bin` directory.

2. Run `./stopLombardi.sh`

> If the stop script fails, specify the user and password when running the script. For example:
>
> `./stopLombardi.sh -user tw_user -password tw_user`

To start the Process Server:

1. From a shell prompt, navigate to the `[Lombardi_home]/process-server/bin` directory.

2. Run `./startProcessServer.sh`

To stop the Process Server:

1. From a shell prompt, navigate to the `[Lombardi_home]/process-server/bin` directory.

2. Run `./stopProcessServer.sh`

To start the Performance Data Warehouse:

1. From a shell prompt, navigate to the `[Lombardi_home]/performance-data-warehouse/bin` directory.

2. Run `./startPerformanceDataWarehouse.sh`

To stop the Performance Data Warehouse:

1. From a shell prompt, navigate to the `[Lombardi_home]/performance-data-warehouse/bin` directory.

2. Run `./stopPerformanceDataWarehouse.sh`

Configuring an LDAP security provider

By default, WebSphere Lombardi Edition is configured with a single federated repository that contains only the internal Lombardi security provider. To use an additional LDAP security provider, you need to add the provider to the federated repository. The following steps describe how to configure an LDAP security provider (such as Microsoft Active Directory) with WebSphere federated repository.

1. From the WebSphere administrative console, navigate to: **Security** > **Global security**

2. Under **User account repository**, select **Federated repositories** from the **Available realm definitions** drop-down list.

3. Click the **Configure** button.

4. Under **Related items**, click **Manage repositories**.

5. Click the **Add** button and input parameters for the provider that you want to add. For example, to add Microsoft Active Directory, you would input parameters like the following:

Repository identifier	`SALOMLDAP // change to suit`
Directory type	`Microsoft Windows Active Directory`
Primary host name	`10.1.5.18`
Bind distinguished name	`cn=LDAP_USER,CN=Users,DC=COMPANYQA,DC=com`
Bind password	`pwsaaswp`

6. Click the **OK** button and then click Save.

7. Click the **Add Base entry to Realm** button and provide values like the following:

Distinguished name of a base entry that uniquely identifies this set of entries in the realm	`DC=COMPANYQA,DC=com`
Distinguished name of a base entry in this repository	`cn=Users,DC=COMPANYQA,DC=com`

8. Click the **OK** button and then click Save.

9. On the Global Security page, click the **Set as current** button and then click the **Apply** button.

10. Shut down all Lombardi servers as instructed in Starting and stopping Lombardi servers.

11. Make sure no duplicate users exist in the Lombardi internal security provider and the security provider that you just added. If duplicate users exist, you will get exceptions when you run WebSphere Lombardi Edition product components.

12. Start all Lombardi servers as instructed in Starting and stopping Lombardi servers.

    If you have configured a server cluster for your runtime environment, be sure to stop and restart all servers in the cluster.

Setting up Lombardi user accounts

IBM WebSphere Lombardi Edition includes an internal security provider, which you can use to create and maintain Lombardi users and groups. You can use the internal Lombardi security provider in conjunction with an external security provider (such as LDAP) that you have configured with WebSphere application server (see Configuring an LDAP security provider). Lombardi internal security provider includes several default users and groups. When you use the internal Lombardi security provider in conjunction with an external provider, the users and groups from both providers are available for selection throughout Lombardi.

> Do not remove the default Lombardi administrator account, tw_admin, or the default administrator group, tw_admins. Administration of Lombardi servers and users is not possible without these default accounts. You can change the password for the tw_admin account as described in Changing internal Lombardi passwords.

The best way to manage security in Lombardi is by using groups. For example, the best way to grant administrative access to Lombardi is to add preexisting groups of users from your external security provider to tw_admins, which is a Lombardi group whose members have administrative access to Lombardi product components by default. Then when changes are required, you can simply add or remove individual users

from the groups that exist in your external security provider. This practice ensures that the security maintenance you perform in your external provider does not require additional work in Lombardi.

> For more information about managing Lombardi users and groups in the Process Admin Console, see *Lombardi Administration Guide.*

When initially configuring Lombardi servers, you should complete tasks similar to the following to set up your users:

Task	Interface	To learn more..
1. Access the Process Admin Console by opening your Web browser to the following location: `http://[host_name]:[port]/ProcessAdmin`. Log on using the default administrative account (tw_admin) and the default password (tw_admin).	Process Admin Console	See *Accessing the Process Admin Console* in *Lombardi Administration Guide*
2. (Optional) Create the different types of users and groups that your users will need in Lombardi.	Process Admin Console	See *Creating and maintaining users* and *Creating and maintaining groups* in *Lombardi Administration Guide*
3. Add members to the default Lombardi groups or groups that you have created. You can add users and groups from any configured external provider (such as LDAP) and internal Lombardi users and groups.	Process Admin Console	See *Managing group membership* in *Lombardi Administration Guide*
4. Access the Process Center Console by opening your Web browser to the following location: `http://[host_name]:[port]/ProcessCenter`. Log on using the default administrative account (tw_admin) and the default password (tw_admin). Add the users and groups who need access to the repository, grant administrative access to the appropriate users, and then establish who can access each process application and toolkit. The best way to grant access to the repository is to add members to the default Lombardi group, tw_authors.	Process Center Console	See *Managing access to the Process Center repository* in the *Authoring Environment User Guide* or online help

Changing internal Lombardi passwords

Once you install Lombardi product components, you can log on to the Process Admin Console and change the default passwords to make your environment more secure. You can change all user passwords if you log on as `tw_admin`.

To change passwords:

1. Access the Process Admin Console by opening your Web browser to the following location: `http://[host_name]:[port]/ProcessAdmin`. Log on using the default administrative account (tw_admin) and the default password (tw_admin).

2. On the Admin tab in the Process Admin Console, click **User Management**.

3. Click the **User Management** option.

4. In the **User Management** > **Maintain User Settings** dialog, enter a user name in the **Retrieve Profile** field.

5. Click the **Retrieve** button.

6. Change settings as required and click the **Update** button.

If you want to change the default Lombardi administrative account user name and password, you must also update the relevant Lombardi configuration files to reflect the new values, as described in Changing the default administrative account password. The new values can be encrypted using the EncryptPassword utility provided with Lombardi, as described in Encrypting passwords.

Changing the default administrative account password

Be sure to change the default Lombardi internal security provider's administrative account password if you intend to restrict access to administrative functions. Doing so is prudent because administrative users can halt and delete processes. To change the `tw_admin` password you need to make changes for Lombardi servers and other tools as described in the following sections:

• Lombardi server changes

• Lombardi tool changes

Lombardi server changes

1. Log on to the Process Admin Console using the default administrative account (tw_admin) and the default password (tw_admin), and change the password for the tw_admin account in the Process Admin Console as described in Changing internal Lombardi passwords.

2. Log out of the Process Admin Console.

3. Stop Lombardi Process Server (see Starting and stopping Lombardi servers).

4. Generate an encrypted value for the new password as described in Encrypting passwords.

5. Copy the following section of the `[Lombardi_home]/process-server/config/system/99Local.xml` file to the `[Lombardi_home]/process-server/config/100Custom.xml` file and make the following changes.

   Add `merge="replace"` to the `<system-lane users>` tag. Also change the value of the password attribute to the encrypted value of the new password (make sure the value of the `password-encrypted` attribute is set to `true`):

```
<server merge="mergeChildren">
    <bpd-engine merge="mergeChildren">
        <system-lane-users merge="replace">
    <!-- Use java -cp utility.jar com.lombardisoftware.utility.EncryptPassword <password> -->
    <!-- to generate encrypted password -->
        <user login="tw_admin" password-encrypted="true" password="new-password]"/>
        </system-lane-users>
    </bpd-engine>
</server>
```

> To learn more about the `merge` options that you can use in the 100Custom.xml file, see Changing Lombardi server properties.

Save your changes.

6. Copy the following lines from the
    `[Lombardi_home]/process-server/config/system/80EventManager.xml` file to the
    `[Lombardi_home]/process-server/config/100Custom.xml` file.

    Add `merge="replace"` to the `<password>` tag and change the value of the `<password>` element
    to the encrypted value of the new password. Also change the value of the `<password-encrypted>`
    element to `true`):

    ```
    <event-manager merge="mergeChildren">
       <password merge="replace">[new-password]</password>
       <password-encrypted merge="replace">true</password-encrypted>
    </event-manager>
    ```

    Save your changes.

7. The contents of your `100Custom.xml` file should look like the following example when you are finished
    with your edits:

    ```
    <properties>
       <server merge="mergeChildren">
        <bpd-engine merge="mergeChildren">
           <system-lane-users merge="replace">
       <!-- Use java -cp utility.jar com.lombardisoftware.utility.EncryptPassword <password> -->
       <!-- to generate encrypted password -->
           <user login="tw_admin" password-encrypted="true" password="new-password]"/>
           </system-lane-users>
        </bpd-engine>
       </server>
       <event-manager merge="mergeChildren">
       <password merge="replace">[new-password]</password>
       <password-encrypted merge="replace">true</password-encrypted>
       </event-manager>
    </properties>
    ```

8. Start Lombardi Process Server (see Starting and stopping Lombardi servers). (If you are running Lombardi
    in a clustered environment, be sure to restart all servers in the cluster.)

Lombardi tool changes

In addition to making the changes to the `100Custom.xml` file as described in the preceding section, you
must also edit the following files to universally change the tw_admin password. The following files are
located in `[Lombardi_home]`.

The files in the twinit directory are created when you initialize Lombardi databases, which is an option that
can be enabled during installation or a task that can be completed after installation as described in Loading
initial data.

File	Required changes
tools/process-installer/process-installer.properties	Change pcs.encrypted and ps.encrypted to the encrypted value from the preceding procedure.
twinit/tw-init.xml	Change the value of the bootstrap.password property to the encrypted value from the preceding procedure.
twinit/resources/config/system/99Local.xml	Change the system-lane-users password to the encrypted value from the preceding procedure. Make sure the value of the `<password-encrypted>` element is `true`.

File	Required changes
twinit/resources/config/system/80EventManager.xml	Change the event-manager password to the encrypted value from the preceding procedure. Make sure the value of the `<password-encrypted>` element is `true`.

Encrypting passwords

When you install Lombardi product components, passwords in the installed files are generally encrypted by default. If you ever need to change these passwords, you can easily encrypt the new passwords using the Lombardi EncryptPassword utility:

1. From a shell prompt, navigate to the `[Lombardi_home]/twinit/lib` directory and then run the following command:

    ```
    java -cp utility.jar com.lombardisoftware.utility.EncryptPassword <password>
    ```

    where `<password>` is the password you want to encrypt.

    Example:

    ```
    java -cp utility.jar com.lombardisoftware.utility.EncryptPassword JohnDoe
    ```

    prints

    ```
    taVCmTLTWWgkTvfvavu/7g==:sROM4ZbvqRDFYMobWXqvwg==
    ```

    If you are running the utility from a directory other than the above, you must replace `utility.jar` with the exact path to the `utility.jar` file.

    If the utility fails with a `ClassNotFound` exception, set your java home to the following directory before running the utility: `[Lombardi_home]/AppServer/java/bin/java`

    Example:

    ```
    java -cp /opt/[Lombardi_home]/twinit/lib/utility.jar
    com.lombardisoftware.utility.EncryptPassword JohnDoe
    ```

    The EncryptPassword utility only works in one direction; you cannot use it to decrypt passwords that are already encrypted.

2. Replace the existing encrypted password in your Lombardi configuration files (such as `[Lombardi_home]/process-server/config/100Custom.xml`.

    When you need to change encrypted passwords in Lombardi configuration files such as `98Database.xml`, `80EventManager.xml`, and `99Local.xml`, you should copy the necessary sections from those files to the `100Custom.xml` file and make changes there as described in Changing the default administrative account password.

Configuring single sign-on

WebSphere Lombardi Edition supports Kerberos-based automatic login (often referred to as Single Sign-On, or SSO). This functionality reduces configuration overhead and simplifies the authentication process by

using the credentials of the current Windows user to authenticate with Lombardi. The Kerberos protocol enables Lombardi users to automatically log on to Lombardi interfaces, including Process Center Console, Process Portal, Process Admin Console, and Performance Admin Console.

When configuring single sign-on for Lombardi product components, ensure that you meet the following requirements:

- For detailed and comprehensive information, refer to the redbook entitled *Implementing Kerberos in a WebSphere Application Server Environment*, which you can find here:
  http://www.redbooks.ibm.com/abstracts/sg247771.html?Open

- Synchronize the date and time between the Key Distribution Center (KDC) and the machines running Lombardi servers.

- Lombardi servers must be run under the domain account for which the Kerberos keytab is generated.

- You must have only one service account and only the service account should be mapped to the Kerberos endpoint.

To configure single sign-on for Lombardi product components, you need to complete the tasks outlined in the following sections:

Configuring the Key Distribution Center (KDC)	Describes the steps required to configure a KDC.
Configuring Lombardi server hosts	Describes the steps required to configure a Windows host for a Lombardi server.
Using WebSphere Admin Console to enable SPNEGO	Describes the steps required to enable SPNEGO.
Configuring your browser	Describes the browser configuration required for each user who wants to take advantage of SSO.

Configuring the Key Distribution Center (KDC)

The following example describes the steps required to configure a Key Distribution Center (KDC).

1. Create a Microsoft Active Directory user account that maps to the Kerberos service principal name (SPN) for WebSphere Application Server as follows:

   a) On the Windows Server system that serves as the Microsoft domain controller, log on as administrator.

   b) Go to: **Control Panel** > **Administrative Tools** > **Active Directory Users and Computers**

   c) Create a user account. For the following sample procedure, the account is called `waskerb`. Make sure that **Use DES encryption types for this account** is not enabled/checked.

2. Use ktpass.exe to create the Kerberos keytab file as shown in the following example:

   ```
    ktpass -out c:\waskerb.keytab -princ HTTP/tw7.sso.com@SSO.COM -mapUser
   waskerb -mapOp set -pass * -crypto RC4-HMAC-NT

   Targeting domain controller: tw7.sso.com
   Using legacy password setting method
   Successfully mapped HTTP/tw7.sso.com to waskerb.
   Type the password for HTTP/tw7.sso.com:
   Type the password again to confirm:
   ```

```
WARNING: pType and account type do not match. This might cause
problems.
Key created.
Output keytab to c:\waskerb.keytab:
Keytab version: 0x502
keysize 72 HTTP/tw7.sso.com@SSO.COM ptype 0
(KRB5_NT_UNKNOWN) vno 5
etype 0x17 (RC4-HMAC) keylength 16
(0x25cf2dc91be14a3509cbada1743bef84)
```

3. Verify the SPN list as shown in the following example:

```
C:\>setspn.exe -l waskerb

Registered ServicePrincipalNames for
CN=waskerb,CN=Users,DC=sso,DC=com:
HTTP/tw7.sso.com
```

4. Run setspn:

```
Setspn -A HTTP/tw7.sso.com@SSO.COM tw7
```

Configuring Lombardi server hosts

The following example describes the steps required to configure a Windows host for a Lombardi server.

1. Copy the keytab file to `c:\windows`.

2. Run `wsadmin` as shown in the following example:

```
wsadmin>$AdminTask createKrbConfigFile {-krbPath c:/WINDOWS/krb5.ini -realm
 SSO.COM -kdcHost sso.com -dns sso.com -keytabPath c:/WINDOWS/waskerb.keytab}

c:/WINDOWS/krb5.ini has been created.
```

3. Make sure your newly created `krb5.ini` file resembles the following example. Pay attention to the encryption method.

```
[libdefaults]
    default_realm = SSO.COM
    default_keytab_name = FILE:///c:/WINDOWS/waskerb.keytab
    default_tkt_enctypes = rc4-hmac
    default_tgs_enctypes = rc4-hmac
    forwardable = false
    renewable = false
    clockskew = 300
[realms]
  SSO.COM = {
    kdc = sso.com:88
    default_domain = sso.com
  }
```

```
[domain_realm]
     .sso.com = SSO.COM
```

Using WebSphere Admin Console to enable SPNEGO

The following sample procedure assumes that you have already configured an LDAP security provider.

1. From the WebSphere administrative console, navigate to: **Security** > **Global security**

2. Under authentication, click **Kerberos configuration** and then select **SPNEGO Web authentication** under Related Configuration.

3. You need to define a filter for each application that will participate in SSO. To do so, under SPNEGO Filters click the **New** button and provide the following information:

Host name	For this sample the host name is: `tw7.sso.com`
Kerberos realm name	For this sample the realm name is: `SSO.COM`

4. Enable the **Trim Kerberos realm from principal name** option.

5. Click the **Apply** button.

6. Click the **Enable SPNEGO** option, select the Kerberos configuration and keytab file, and then click **Apply**.

7. On the Global security panel, in the Authentication section, click **Kerberos configuration**.

8. Change the Kerberos service name to `HTTP`.

9. Enter `SSO.COM` as the Kerberos realm name.

10. Ensure the **Trim Kerberos realm from principal name** option is enabled.

11. Clear the checkbox for the **Enable delegation of Kerberos credentials** option.

12. Save the configuration changes.

13. Stop and start all Lombardi servers.

Configuring your browser

When the preceding tasks are completed, each user who wants to take advantage of SSO must appropriately configure their browser settings. The following example describes how to configure Internet Explorer for the configuration established using the preceding tasks.

1. In Internet Explorer, go to **Tools** > **Internet Options** > **Security** > **Local Intranet** > **Sites** and click the **Advanced** option.

2. Add the WebSphere fully qualified name to the sites list. (For the preceding sample: `http://tw7.sso.com`)

3. Go to the **Advanced** tab and ensure that the **Enable Integrated Windows Authentication** option is selected.

Troubleshooting

If you have issues configuring SSO, refer to the following resources:

- http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/rsec_SPNEGO_trouble_shoot.html

- http://www.redbooks.ibm.com/abstracts/sg247771.html?Open

Optional configuration tasks for Lombardi servers

The following topics provide instructions for optional post-installation configuration tasks for your Lombardi installation and environment.

Changing Lombardi server properties

When changing a Lombardi configuration, you should not change the setting in the original configuration file. For example, you should not change properties in `[Lombardi_home]/process-server/config/system/00Static.xml` directly. You should instead add the configuration you want to change to the `[Lombardi_home]/process-server/config/100Custom.xml` file. Lombardi merges the changes you make in the `100Custom.xml` file with the original configuration, as described in the following table.

> You should always use the 100Custom.xml file to change configuration settings so that all customizations are captured in a single file and are readily available when you upgrade to a newer version of Lombardi. When you do upgrade, remember to review the settings in the 100Custom.xml file to ensure they are appropriate for the new version.

When using 100Custom.xml to change Lombardi configuration settings, you can use the merge attribute for each element. The merge attribute supports the following options:

Option	Description
`"append"`	Appends the new tag regardless.
`"mergeChildren"`	Merges the new tag with the first of the existing tags. (This is the default behavior.)
`"replace"`	Replaces all old tags with the new one.

The following example describes how to use 100Custom.xml to change the environment name in the 99Local.xml file.

1. Stop Lombardi Process Server (see Starting and stopping Lombardi servers).

2. Open the `[Lombardi_home]/process-server/config/system/99Local.xml` file in a text editor and you see content like the following:

```
<properties>
...
  <common merge="mergeChildren">
  ....
     <environment-name>[existing_name]</environment-name>
     ...
  </common>
...
</properties>
```

3. To change the environment name, you can specify the following in the 100Custom.xml file:

```
<properties>
    <common merge="mergeChildren">
        <environment-name merge="replace">[new_name]</environment-name>
    </common>
</properties>
```

4. Start Lombardi Process Server (see Starting and stopping Lombardi servers). (If you are running Lombardi in a clustered environment, be sure to restart all servers in the cluster.)

> Make a backup copy of the configuration files you are changing and when backing them up, be sure to change the file extension to something like .bak. Changing the extension is necessary to ensure that Lombardi does not attempt to load the backup files instead of the originals during start up.

Connecting a runtime environment to a Process Center

The Process Server in each Lombardi runtime environment can be connected to the Process Center. This enables Lombardi users to install snapshots of process applications from the Process Center Console to the connected Process Servers in test, production, and other environments. An example of this configuration (one Process Center connected to multiple runtime environments) is shown in Lombardi architecture.

If you want to connect the Process Server in your runtime environment to a different Process Center or if you want to connect an offline Process Server to a Process Center, you do not need to reinstall the runtime environment. However, you must update the Process Center host information in the Lombardi configuration properties as described in the following procedure.

1. Stop Lombardi Process Server (see Starting and stopping Lombardi servers).

2. Open the `[Lombardi_home]/process-server/config/system/99Local.xml` file in a text editor.

3. Find the following section:

   ```
   <repository-server-url>http://[process-center-host-name]:[port]/ProcessCenter</repository-server-url>
   <repository-server-user>tw_runtime_server</repository-server-user>
   <repository-server-password>XDfGSzXrZ7nYglKiB:yMR2XQwJXzQACE</repository-server-password>
   <repository-server-password-encrypted>true</repository-server-password-encrypted>
   <repository-server-designated-user>tw_author</repository-server-designated-user>
   <repository-server-designated-password>fZN75qYgnZll0MCV80AAhA--</repository-server-designated-password>
   <repository-server-designated-password-encrypted>true</repository-server-designated-password-encrypted>
   <repository-server-interval>5</repository-server-interval>
   ```

   The name and port of your Process Center host are defined in the `<repository-server-url>` element.

4. Copy the `<repository-server-url>` element and any other lines that contain elements that you want to change to the `[Lombardi_home]/process-server/config/100Custom.xml` file.

5. Change the value of `<repository-server-url>` to the name and port of the Process Center host to which you want to connect, using the following syntax:
   `http://[process-center-host-name]:[port]`. Use `merge="replace"` in the 100Custom.xml file as shown in the following example:

   ```
   <repository-server-url
   merge="replace">[http://process-center-host-name:port]/ProcessCenter<repository-server-url>
   ```

To learn more about the `merge` options that you can use in the `100Custom.xml` file, see Changing Lombardi server properties.

6. You can also change the following elements:

Element	Description
repository-server-user	User name used to connect to the designated Process Center. By default, Lombardi uses the tw_runtime_server user account. If you specify a different user for this connection, the user that you specify must exist on the Process Center.
repository-server-password	Password used to connect to the designated Process Center. By default, Lombardi uses the password for the tw_runtime_server user account.
repository-server-password-encrypted	By default, the password for the account used to connect to the Process Center is encrypted.
repository-server-designated-user	User name used to connect to this runtime Process Server when: (1) running BPDs or services in the Inspector in Lombardi Authoring Environment and (2) installing snapshots of process applications. (When running BPDs or services in the Inspector, you can choose to run on any connected Process Server. When installing process application snapshots, you can choose any connected Process Server.) By default, Lombardi uses the tw_author user account. If you specify a different user for this connection, the user that you specify must exist on the runtime Process Server and must be a member of the tw_authors security group.
repository-server-designated-password	Password used to connect to this runtime Process Server when running BPDs or services in the Inspector in Lombardi Authoring Environment. By default, Lombardi uses the password for the tw_author user account.
repository-server-designated-password-encrypted	By default, this password is encrypted.
repository-server-interval	Polling interval (in seconds) at which the Process Server in your runtime environment polls the Process Center for updates. Polling is disabled by using a negative value, so be sure that this value is positive when changing settings to connect to a Process Center.

See Encrypting passwords for more information about encrypting passwords.

7. The contents of your `100Custom.xml` file should look like the following example when you are finished with your edits. In the following example, we're specifying a Process Center Server and changing the password for the the default user account (tw_runtime_server) used to connect to that server:

```
<properties>
 <common merge="mergeChildren">
  <server merge="mergeChildren">
    <repository-server-url merge="replace">[http://process-center-host-name:port/ProcessCenter]</repository-server-url>
    <repository-server-password merge="replace">[new_password]</repository-server-password>
  </server>
 </common>
</properties>
```

8. Save your changes.

9. Start Lombardi Process Server (see Starting and stopping Lombardi servers).

Changing the Process Server environment type

During development in Lombardi Authoring Environment, process authors can set environment-specific variables for each process application. These variables enable process authors to define values for each type of environment in which a process will run. The environment type that you select during installation establishes the type of environment in which this server resides and thus the environment variable values that apply to this server. Although Lombardi Installer only supports four environment types (development, test, staging, and production), you can specify a custom environment type for your Process Server in the configuration properties using the following procedure.

To change the environment type for the Process Server:

1. Stop Lombardi Process Server (see Starting and stopping Lombardi servers).

2. Open the `[Lombardi_home]/process-server/config/system/99Local.xml` file in a text editor.

3. Copy the following element to the `[Lombardi_home]/process-server/config/100Custom.xml` file:

    ```
    <environment-type>TEST</environment-type>
    ```

4. Change the value of the `<environment-type>` element to the desired type. Use `merge="replace"` in the `100Custom.xml` file as shown in the following example:

    ```
    <environment-type merge="replace">STAGING</environment-type>
    ```

    > To learn more about the `merge` options that you can use in the `100Custom.xml` file, see Changing Lombardi server properties.

5. The contents of your `100Custom.xml` file should look like the following example when you are finished with your edits:

    ```
    <properties>
        <common merge="mergeChildren">
            <environment-type merge="replace">STAGING</environment-type>
        </common>
    </properties>
    ```

6. Save your changes.

7. Start Lombardi Process Server (see Starting and stopping Lombardi servers).

Running a Process Server without a Performance Data Warehouse

You might find a situation when you want to run Lombardi Process Server without connecting to a Performance Data Warehouse. You can disconnect the Performance Data Warehouse communication by changing a setting in the Lombardi properties files. Disconnecting the Performance Data Warehouse communication enables you to work in the Process Server without seeing connection errors.

> If you disconnect, you will not be able to track data required for reports and optimization.

1. Open the `[Lombardi_home]/process-server/config/100Custom.xml` file in a text editor.

2. Add the following elements to merge with the other properties files:

   ```
   <common> <performance-data-warehouse-communication> <enabled
   merge="replace">false</enabled> </performance-data-warehouse-communication>
    </common>
   ```

3. Restart Lombardi Process Server. Communication between the Performance Data Warehouse and the Process Server is now disabled. Any tracking you have set up will not occur during the time that the communication is down.

Preventing XSS vulnerability

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in Web applications that allow a malicious Web user to inject code, such as HTML and client-side scripts, into Web pages viewed by other users. Attackers can exploit cross-site scripting vulnerability to avoid security access controls.

The `<escape-user-input>` property in the `[Lombardi_home]/process-server/config/system/00Static.xml` configuration file prevents a user who is working in Lombardi Process Portal or Lombardi Process Admin Console from executing Java™ script that is embedded in the URLs and might cause a security risk.

The `<escape-user-input>` property can have one of two values:

Value	Function
false	Disables escaping of user input in Lombardi Process Portal and Process Admin Console URLs. When escaping is disabled, Java script that is inserted into the URLs can be executed.
true	Enables escaping of user input in Lombardi Process Portal and Process Admin Console URLs. When escaping is enabled, Java script that is inserted into the URLs cannot be executed.

Allocating Async and Sync threads

Separate pools of threads are available for running Undercover Agents (UCAs) and processes (BPDs) in designated Asynchronous and Synchronous queues. To control the number of threads assigned to executing BPDs, and asynchronous and synchronous UCAs, override the default values set in the `[Lombardi_home]/process-server/config/system/80EventManager.xml` system configuration file using the `[Lombardi_home]/process-server/config/100Custom.xml` file.

> Increasing the number of threads increases the required system resources needed for processing.

See Changing Lombardi server properties for more information about overriding default values in Lombardi configuration files.

Specifying the run-time sender of tasks

When a task is generated at run time and sent to the assigned participant, the task sender is the `tw_admin` user. For example, the **Sent From** field in Lombardi Process Portal displays this user name.

The default task sender is set in the `[Lombardi_home]/process-server/config/system/99Local.xml` file.

To specify another sender for run-time tasks:

1. Open the `[Lombardi_home]/process-server/config/100Custom.xml` file in a text editor, and then add the following code block inside the <common> element of the file:

```
<common merge="mergeChildren">
     <user-to-create-task
merge="replace">[custom_task_sender_name]</user-to-create-task>
   </common>
```

2. For the custom task sender name, specify the appropriate user name and then save your changes.

3. Restart the Lombardi server (if you are running a clustered environment, restart all servers in the cluster).

Removing Lombardi components

If you need to remove Lombardi from your system, the Uninstaller program removes installation files and folders for Lombardi and the embedded application server.

Before you begin:

- Back up any files that you want to preserve.

1. Run the Uninstaller program, `[Lombardi_home]/uninstall/uninstall.sh` .

2. When the Uninstaller program completes, you might need to remove some installation files manually. For example, if you have a shell open to a directory in the Lombardi installation directory, the Uninstaller program cannot remove that directory.

Notices and Trademarks

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this documentation does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (I) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Intellectual Property Dept. for WebSphere Software
IBM Corporation
3600 Steeles Ave. East
Markham, Ontario
Canada L3R 9Z7

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks and service marks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.